

Lígia Maria Carvalho Sousa

**Esquema de Controle Adaptativo de Fluxos  
de Tráfego Baseado em Modelagem *Fuzzy* Preditiva**

**Dissertação de Mestrado** apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.  
Área de concentração: Telecomunicações e Telemática.

Banca Examinadora:

Prof. Dr. Lee Luan Ling - UNICAMP

Prof. Dr. Paulo Cardieri - UNICAMP

Prof. Dr. Renato Baldini Filho - UNICAMP

Prof. Dr. Roseli Aparecida Romero - USP (São Carlos)

Campinas, SP

2007

Lígia Maria Carvalho Sousa

**Esquema de Controle Adaptativo de Fluxos  
de Tráfego Baseado em Modelagem *Fuzzy* Preditiva**

**Dissertação de Mestrado** apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Telecomunicações e Telemática.

Orientador: Lee Luan Ling

Co-orientador: Flávio Henrique Teles Vieira

Campinas, SP  
2007

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

So85e Sousa, Lígia Maria Carvalho  
Esquema de controle adaptativo de fluxos de tráfego baseado em modelagem *fuzzy* preditiva \ Ligia Maria Carvalho Sousa. – Campinas, SP: [s.n.], 2007.

Orientadores: Lee Luan Ling, Flavio Henrique Teles Vieira

Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Redes de Computação. 2. Telecomunicações - Tráfego. 3. Previsão. 4. Lógica Difusa. 5. Modelagem de Dados. I. Lee, Luan Ling. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título

Título em Inglês: Predictive Fuzzy Modeling for Adaptive Control of Network Traffic Flows  
Palavras-chave em Inglês: Computer networks, Traffic prediction, Flows control, Congestion, Fuzzy logic, TSK model, Quality of service.  
Área de concentração: Telecomunicações e Telemática  
Titulação: Mestre em Engenharia Elétrica  
Banca Examinadora: Roseli Aparecida Francellin Romero, Paulo Cardieri e Renato Baldini Filho  
Data da defesa: 24/05/2007  
Programa de Pós-Graduação: Engenharia Elétrica

# Resumo

O presente trabalho propõe um esquema de controle adaptativo de fluxos baseado no modelo *fuzzy* TSK. Neste esquema de controle, o modelo *fuzzy* TSK é utilizado para prever adaptativamente o tamanho da fila no *buffer* em um enlace. Com o objetivo de ajustar dinamicamente os parâmetros do modelo *fuzzy* TSK, propomos um algoritmo de treinamento adaptativo. Na primeira etapa do algoritmo de treinamento proposto, os parâmetros das partes premissas e das partes conseqüentes do modelo são obtidos. A segunda etapa consiste de um algoritmo de refinamento dos parâmetros do modelo baseado em gradiente descendente. A eficiência do preditor proposto é avaliada através da comparação com outros preditores adaptativos fazendo uso de traços de tráfego reais. A partir dos parâmetros do modelo *fuzzy* TSK, derivamos uma expressão para a taxa da fonte controlável a qual minimiza a variância do tamanho de fila no *buffer*. O controle de congestionamento proposto é então aplicado em diferentes cenários de rede com vários nós. Comparações realizadas com outros métodos de controle de congestionamento demonstram que o controle de congestionamento proposto obtém menores taxas de perdas e consegue de fato manter o tamanho da fila no *buffer* abaixo do valor desejado.

**Palavras-chave:** Redes de Computadores, Predição de Tráfego, Controle de Fluxos, Congestionamento, Lógica Nebulosa, Modelagem TSK, Qualidade de Serviço.

# Abstract

The present work proposes a adaptive control of traffic flows based in the TSK fuzzy model. In this control, the TSK fuzzy model is used to predict in a manner adaptive the buffer length in one output link. With the objective of dynamically adjust the parameters of the TSK fuzzy model, we proposed a adaptive training algorithm. In the first stage of the proposed training algorithm, the parameters of the premise and consequent parts of the model are obtained. The second stage consists of a refining algorithm of the parameters based in descent gradient. The effectiveness of the proposed predictor is evaluated through comparison with other adaptive predictors by using real network traffic traces. With the parameters of the TSK model, we derive an expression for the controllable source rate which minimizes the variance of the buffer length. The proposed congestion control is applied in different network sceneries with several nodes. Comparison made with others congestion control methods demonstrates that the proposed congestion control obtain lesser loss rate and gets in fact to keep the buffer length below of the reference level.

**Keywords:** Computer Networks, Traffic Prediction, Flows Control, Congestion, Fuzzy Logic, TSK Model, Quality of Service.

*"Tudo é do Pai,  
toda honra e toda glória,  
é Dele a vitória  
alcançada em minha vida."*



# Agradecimentos

À Deus, minha Fortaleza, por iluminar meu caminho e me dar forças para seguir sempre em frente.

Aos meus pais João e Margarida, os grandes responsáveis por eu estar hoje aqui. Agradeço pelo amor incondicional, companheirismo e incentivo. Foi na distância que pude perceber a grande importância deles na minha vida, e sei que sem eles nada disso teria sido possível ou teria valido a pena. Sei o quanto eles sofreram com a minha ausência.

Ao meu irmão Lívio, pelo incentivo e apoio. Por todas as dúvidas tiradas, pela força dada nos meus primeiros difíceis meses em Campinas e por toda a paciência.

Ao meu namorado Eduardo, pelo amor, companheirismo, paciência e total incentivo mesmo sabendo que ficaríamos distantes. Com o seu amor, pude superar os momentos difíceis que passei e, de certa forma, eles serviram para fortalecer o nosso sentimento.

Ao meu orientador Lee Luan Ling pelo apoio e orientação dedicados a mim durante este mestrado. Agradeço também pela sempre prontidão, pela confiança em mim depositada e pelo incentivo e esforço para com a minha carreira acadêmica refletidos nas publicações que tive durante o mestrado.

Ao meu co-orientador Flávio Henrique pelo companheirismo e expressiva contribuição a este trabalho. Pelo interesse sempre demonstrado com o meu mestrado, pelas correções e comentários que enriqueceram bastante este trabalho, pelos ensinamentos e por todo o apoio.

Aos companheiros de laboratório Christian e Ricardo Nagel, por toda a ajuda e apoio.

Aos amigos e companheiros que convivi em Campinas, os quais tornaram esse período do mestrado agradável e menos saudosos: Ariadne, André, Angela, Fabiano, Fabio Lumertz, Helder, Marzio, Rafael Mota, Rodrigo, Tarciana, tia Graçinha. Agradeço por todos os churrascos na República dos Cearenses, pelos lanches na cantina da mecânica, pelas sextas e sábados no Kaula, pelas confraternizações de Natal, pelos almoços organizados sempre de última hora no domingo, pelos jogos durante a copa, pelas festas de aniversário "não-tão-surpresas", enfim, por todo o bom convívio que tive aqui, pois mais que grandes amigos foram minha família.

Ao meu amigo-irmão Rafael Barbosa (Babinha) por toda a amizade e apoio, por todas as dúvidas tiradas, por toda a preocupação e dedicação, enfim, por ser um amigo tão especial. Desculpa também pelo "Babinha", mas não poderia deixar passar!

Aos profs. Charles Cavalcante e João César Mota pela grande ajuda na fase final de correção da

dissertação e por todo o apoio e acolhida no meu regresso à Fortaleza.

Aos familiares e amigos, tão importante nesse processo, agradeço pelo apoio e torcida.

*Aos meus pais João e Margarida pelo amor incondicional,  
ao meu irmão Lívio pelos ensinamentos e apoio  
e ao meu namorado Eduardo pelo companheirismo,  
amor e compreensão.*



# Sumário

<b>Lista de Figuras</b>	<b>xv</b>
<b>Lista de Tabelas</b>	<b>xvii</b>
<b>Glossário</b>	<b>xix</b>
<b>Lista de Símbolos</b>	<b>xxi</b>
<b>Trabalhos Publicados Pelo Autor</b>	<b>xxiii</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 Controle de Congestionamento e Redes de Serviços</b>	<b>5</b>
2.1 Introdução . . . . .	5
2.2 Princípios Básicos dos Mecanismos de Controle de Congestionamento . . . . .	6
2.2.1 Controle de Congestionamento Baseado em Janela de Transmissão e Baseado em Taxa de Transmissão . . . . .	6
2.2.2 <i>Unicast e Multicast</i> . . . . .	7
2.2.3 Taxa Única e Multi-taxa . . . . .	8
2.2.4 Controle de Congestionamento Apoiado pelo Roteador e Fim-a-fim . . . . .	8
2.2.5 Controle de Congestionamento Orientados ao Transmissor e Orientados ao Receptor . . . . .	9
2.3 Mecanismos de Realimentação do Nível de Congestionamento . . . . .	9
2.3.1 Descarte de Pacotes . . . . .	10
2.3.2 Notificação de Congestionamento Explícito - ECN . . . . .	10
2.3.3 <i>ICMP Source Quench</i> . . . . .	10
2.4 Políticas de Gerência de Filas nos Roteadores . . . . .	11
2.4.1 Fila FIFO . . . . .	11
2.4.2 Fila de Prioridade Estrita - SP . . . . .	11
2.4.3 Fila Baseada em Classes - CBQ . . . . .	12
2.4.4 Fila Justa - FQ e Fila Justa Ponderada - WFQ . . . . .	12
2.5 Políticas de Descarte de Pacotes nas Filas dos Roteadores . . . . .	13
2.5.1 Descarte Caudatório ( <i>Tail Drop</i> ) . . . . .	13
2.5.2 Detecção Precoce Aleatória ( <i>Random Early Detection</i> ) - RED . . . . .	14
2.6 Análise dos Tipos de Serviços em Redes de Dados . . . . .	14

2.6.1	Redes de Pacotes e de Circuitos . . . . .	15
2.6.2	Qualidade de Serviço - QoS . . . . .	15
2.7	Considerações Finais . . . . .	19
<b>3</b>	<b>Sistemas Nebulosos</b>	<b>21</b>
3.1	Introdução . . . . .	21
3.2	Conjuntos Nebulosos . . . . .	22
3.2.1	Definição de Conjuntos Nebulosos . . . . .	22
3.2.2	Funções de Pertinência . . . . .	23
3.2.3	Definições e Operações com Conjuntos Nebulosos . . . . .	24
3.2.4	Operações Algébricas . . . . .	25
3.3	Relações Nebulosas . . . . .	25
3.3.1	Definição . . . . .	25
3.3.2	Norma-T e Conorma-T (ou S-Norma) . . . . .	26
3.4	Lógica Nebulosa ( <i>Fuzzy Logic</i> ) . . . . .	26
3.4.1	Variável Nebulosa e Variável Linguística . . . . .	27
3.4.2	Regra de Inferência . . . . .	27
3.4.3	Sistemas Nebulosos . . . . .	27
3.5	Considerações Finais . . . . .	30
<b>4</b>	<b>Esquema de Controle de Congestionamento Adaptativo Baseado em Modelagem <i>Fuzzy</i></b>	<b>31</b>
4.1	Introdução . . . . .	31
4.2	Descrição do Sistema de Controle de Congestionamento Proposto . . . . .	31
4.3	Preditor <i>Fuzzy</i> TSK Adaptativo . . . . .	33
4.3.1	Modelo <i>Fuzzy</i> TSK Aplicado à Predição do Tamanho da Fila no <i>Buffer</i> . . . . .	33
4.3.2	Algoritmo ARFARC - Parte 1: Modelo TSK Inicial . . . . .	35
4.3.3	Algoritmo ARFARC - Parte 2: Algoritmo de Refinamento . . . . .	39
4.4	Cálculo da Taxa Ótima de Controle de Fluxo . . . . .	41
4.5	Considerações Finais . . . . .	44
<b>5</b>	<b>Avaliação do Preditor Proposto</b>	<b>47</b>
5.1	Introdução . . . . .	47
5.2	Predição Linear . . . . .	47
5.2.1	Algoritmo LMS ( <i>Least Mean Square</i> ) . . . . .	52
5.2.2	Algoritmo RLS ( <i>Recursive Least Square</i> ) . . . . .	54
5.3	Medidas de Desempenho de Predição . . . . .	58
5.3.1	Erro Quadrático Médio - EQM . . . . .	58
5.3.2	Erro Quadrático Médio Normalizado 1 - EQMN1 . . . . .	59
5.3.3	Erro Quadrático Médio Normalizado 2 - EQMN2 . . . . .	59
5.4	Caracterização das Séries de Tráfego Utilizadas . . . . .	60
5.4.1	A Característica Multifractal . . . . .	60
5.4.2	Verificação do Comportamento Multifractal do Tráfego - Diagrama Multiescala Linear . . . . .	61
5.4.3	Traços Utilizados nas Simulações . . . . .	62

5.4.4	Verificação do Comportamento Multifractal dos Traços de Tráfego Utilizados	64
5.5	Análise de Desempenho de Predição	66
5.5.1	Predição dos Traços de Tráfego	68
5.5.2	Análise do Desempenho do Algoritmo de Refinamento	72
5.5.3	Análise do Erro Quadrático Médio	74
5.6	Considerações Finais	75
<b>6</b>	<b>Análise do Esquema de Controle de Congestionamento Proposto</b>	<b>79</b>
6.1	Introdução	79
6.2	Esquemas de Controle de Congestionamento Considerados como Comparação	80
6.2.1	Realimentação Binária ( <i>Binary Feedback</i> )	80
6.2.2	Controle de Congestionamento Proporcional ( <i>Proportional Congestion Control</i> )	80
6.3	Medidas de Desempenho	82
6.4	Etapa 1 da Análise: Avaliação em 1 Nó	82
6.4.1	Caso 1 - Taxa Controlável > Taxa Não-Controlável	82
6.4.2	Caso 2 - Taxa Controlável < Taxa Não-Controlável	85
6.5	Etapa 2 da Análise: Avaliação em 3 Nós - Traços de Tráfego <i>dec-pkt-2</i> e <i>dec-pkt-3</i>	87
6.5.1	Cenário de Simulação	87
6.5.2	Desempenho do Esquema de Controle de Congestionamento Proposto	88
6.6	Etapa 3 da Análise: Avaliação em 1 nó - Traços monofractais fGn	94
6.7	Etapa 4 da Análise: Avaliação com 3 Nós - Traços de Tráfego <i>3-7-I-1</i> , <i>4-7-I-1</i> , <i>4-7-I-9</i> , <i>4-7-I-5</i> e <i>4-7-I-2</i>	97
6.7.1	Cenário de Simulação (Séries Multifractais)	97
6.7.2	Desempenho do Esquema de Controle de Congestionamento	97
6.8	Análise da Predição do Tamanho da Fila no <i>Buffer</i>	103
6.9	Análise do Efeito do Fator de Ponderação ( $\lambda$ ) no Desempenho do Sistema	107
6.10	Aplicações	107
6.10.1	ATM - <i>Asynchronous Transfer Mode</i>	108
6.10.2	MPLS - <i>Multiprotocol Label Switching</i>	108
6.11	Considerações Finais	110
<b>7</b>	<b>Conclusões</b>	<b>111</b>
	<b>Referências bibliográficas</b>	<b>113</b>
<b>A</b>	<b>Cálculo Recursivo do Vetor de Parâmetros Consequentes <math>\vec{a}^i</math></b>	<b>119</b>
<b>B</b>	<b>Teorema da Inversão Matricial</b>	<b>123</b>
B.1	Desenvolvimento Matemático da Equação da Matriz de Covariância $P_i(k+1)$ Equação (A.14)	123
B.2	Desenvolvimento Matemático da Equação da Matriz de Correlação Cruzada $\Phi(k)$ Equação (5.37)	124
<b>C</b>	<b>Cálculo das Equações de Ajuste dos Parâmetros das Partes Antecedentes e Consequentes</b>	<b>125</b>



# Lista de Figuras

2.1	Exemplo da Largura de Banda em Excesso para Serviços de Taxa de Bits Disponível.	17
2.2	Modelo de Realimentação da Fonte TBD com Atraso. . . . .	18
3.1	Tipos Básicos de Funções de Pertinência. . . . .	23
3.2	Sistema Nebuloso Puro com Duas Entradas e Uma Saída . . . . .	28
3.3	Sistema Nebuloso Takagi-Sugeno. . . . .	29
3.4	Sistema Nebuloso com Fuzificador e Defuzificador . . . . .	30
4.1	Modelo do Sistema de Controle de Congestionamento Proposto. . . . .	32
4.2	Situação Exemplar de um Modelo Nebuloso com 3 Regras. . . . .	41
5.1	Filtro Linear Transversal. . . . .	48
5.2	Análise da Característica Fractal dos Traços TCP/IP. . . . .	64
5.3	Análise da Característica Fractal dos Traços da Petrobrás. . . . .	65
5.4	Análise da Característica Fractal dos Traços LAN Ethernet. . . . .	66
5.5	Análise da Característica Fractal dos Traços fGn. . . . .	67
5.6	Predição a um Passo das Séries Temporais de Tráfego. Série Real (Linha Tracejada), Série Predita (Linha Cheia). . . . .	69
5.7	Predição a um Passo das Séries Temporais de Tráfego. Série Real (Linha Tracejada), Série Predita (Linha Cheia). . . . .	70
5.8	Comparação das Medidas de Erro de Predição Antes e Depois da Aplicação do Algoritmo de Refinamento para o Traço <i>dec-pkt-1</i> . . . . .	73
5.9	Comparação das Medidas de Erro de Predição Antes e Depois da Aplicação do Algoritmo de Refinamento para o Traço <i>OctInt</i> . . . . .	74
5.10	Comparação das Medidas de Erro entre o Preditor Proposto e os Utilizados na Análise.	75
5.11	Comparação das Medidas de Erro entre o Preditor Proposto e os Utilizados na Análise.	76
5.12	Comparação das Medidas de Erro entre o Preditor Proposto e os Utilizados na Análise.	77
6.1	Sinal de Controle $c(k)$ do Controle de Congestionamento Proporcional. . . . .	81
6.2	Taxas dos Fluxos Controlável e Não-controlável . . . . .	83
6.3	Análise do Tamanho da Fila no <i>Buffer</i> . . . . .	84
6.4	Taxas dos Fluxos Controlável e Não-controlável. . . . .	85
6.5	Análise do Tamanho da Fila no <i>Buffer</i> . . . . .	86
6.6	Topologia de Rede Proposta para Análise. . . . .	87
6.7	Taxas dos Fluxos Controlável e Não-Controlável para Cada Nó do Cenário. . . . .	88

6.8	Tamanho da Fila no <i>Buffer</i> Antes e Depois da Aplicação do Controle de Congestionamento Proposto para Cada Nó do Cenário. Antes (Linha Tracejada), Depois (Linha Cheia). . . . .	89
6.9	Taxa da Fonte Controlável Antes e Depois da Aplicação do Controle de Congestionamento Proposto para Cada Nó do Cenário. Antes (Linha Tracejada), Depois (Linha Cheia). . . . .	90
6.10	Comparação do Tamanho da Fila no <i>Buffer</i> Resultante do Controle de Congestionamento Proposto e dos Outros Métodos Comparativos para Cada Nó do Cenário. . . .	91
6.11	Taxa da Fonte Controlável Antes e Depois da Aplicação do Controle de Realimentação Binária e do Controle Proporcional para o Nó 1 do Cenário. Antes (Linha Tracejada), Depois (Linha Cheia). . . . .	92
6.12	Análise das Taxas Controlável e Não-controlável . . . . .	95
6.13	Análise do Tamanho da Fila no <i>Buffer</i> . . . . .	95
6.14	Taxa da Fonte Controlável Antes e Depois da Aplicação do Controle de Realimentação Binária e do Controle Proporcional. Antes (Linha Tracejada), Depois (Linha Cheia). . . . .	96
6.15	Taxas dos Fluxos Controlável e Não-Controlável para Cada Nó do Cenário. . . . .	98
6.16	Tamanho da Fila no <i>Buffer</i> Antes e Depois da Aplicação do Controle de Congestionamento Proposto para Cada Nó do Cenário. Antes (Linha Tracejada), Depois (Linha Cheia). . . . .	99
6.17	Taxa da Fonte Controlável Antes e Depois da Aplicação do Controle de Congestionamento Proposto para Cada Nó do Cenário. Antes (Linha Tracejada), Depois (Linha Cheia). . . . .	100
6.18	Comparação do Tamanho da Fila no <i>Buffer</i> Resultante do Controle de Congestionamento Proposto e dos Outros Métodos Comparativos para Cada Nó do Cenário. . . .	101
6.19	Predição a um Passo do Tamanho da Fila no <i>Buffer</i> . Real (Linha Tracejada), Predito (Linha Cheia). . . . .	105
6.20	Predição a um Passo do Tamanho da Fila no <i>Buffer</i> . Real (Linha Tracejada), Predito (Linha Cheia). . . . .	106
6.21	Estrutura Básica de uma Rede MPLS. . . . .	109

# Lista de Tabelas

5.1	Taxa Média e Taxa de Pico dos Traços de Tráfego Utilizados nas Simulações. . . . .	63
5.2	Resultados de Desempenho do Preditor Proposto para Diferentes Ordens $n$ . . . . .	68
5.3	Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Traço de Tráfego Real <i>dec-pkt-1</i> . . . . .	71
5.4	Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Traço de Tráfego Real <i>dec-pkt-2</i> . . . . .	71
5.5	Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Traço de Tráfego Real <i>OctExt</i> . . . . .	71
5.6	Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Traço de Tráfego Real <i>OctInt</i> . . . . .	71
5.7	Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Traço de Tráfego Real <i>FL10-7-S-1</i> . . . . .	72
5.8	Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Traço de Tráfego Real <i>4-7-I-9</i> . . . . .	72
5.9	Comparação dos Resultados de Desempenho do Preditor Proposto Antes e Depois da Aplicação do Algoritmo de Refinamento . . . . .	73
6.1	Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento para o Caso 1. . . . .	84
6.2	Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento para o Caso 2. . . . .	86
6.3	Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento no Nó 1 - Cenário 1. . . . .	93
6.4	Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento no Nó 2 - Cenário 1. . . . .	93
6.5	Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento no Nó 3 - Cenário 1. . . . .	93
6.6	Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento. . . . .	96
6.7	Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento no Nó 1 - Cenário 2. . . . .	102
6.8	Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento no Nó 2 - Cenário 2. . . . .	102

---

6.9	Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento no Nó 3 - Cenário 2. . . . .	102
6.10	Resultados de Desempenho do Preditor Proposto para Diferentes Ordens $n$ Considerando o Nó 1. . . . .	104
6.11	Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Nó 1. . . . .	104
6.12	Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Nó 2. . . . .	104
6.13	Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Nó 3. . . . .	104
6.14	Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Nó com Traços Monofractais fGn. . . . .	106
6.15	Resultados de Desempenho do Preditor Proposto Para Diferentes Ordens $n$ Considerando o Nó com Traços Monofractais fGn. . . . .	106
6.16	Resultados de Desempenho do Esquema de Controle de Congestionamento Proposto para Diferentes $\lambda$ 's. . . . .	107

# Glossário

AIMD - Additive Increase Multiplicative Decrease

ARFARC - Agrupamento Recursivo *Fuzzy* Adaptativo com Reset da Covariância

ATM - Asynchronous Transfer Mode

BLR - Byte Loss Rate

CBQ - Class-Based Queuing

ECN - Explicit Congestion Notification

FIFO - First-In First-Out

FQ - Fair Queuing

ICMP - Internet Control Message Protocol

IP - Internet Protocol

ISDN - Integrated Services Digital Network

MCR - Minimum Cell Rate

MPLS - Multiprotocol Label Switching

QoS - Quality of Service

RED - Random Early Detection

TBC - Taxa de Bits Constante

TBD - Taxa de Bits Disponível

TBV - Taxa de Bits Variável

TCP - Transmission Control Protocol

TPB - Taxa de Perda de Byte

UPC - Usage Parameter Control

WFQ - Weighted Fair Queuing



# Lista de Símbolos

- $\eta$  - Capacidade do enlace de saída
- $d$  - Atraso de *round-trip*
- $\mu(k)$  - Tráfego controlável no instante  $k$
- $\nu(k)$  - Tráfego não-controlável no instante  $k$
- $b(k)$  - Tamanho da fila no *buffer* no instante  $k$
- $b^\tau$  - Nível desejado para o tamanho da fila no *buffer*
- $\hat{b}(k)$  - Predição do tamanho da fila no *buffer* no instante  $k$
- $\mu^o(k)$  - Taxa de controle ótima
- $B_{max}$  - Tamanho do *buffer*
- $N$  - Número de dados de treinamento
- $\vec{x}(k)$  -  $k$ -ésimo vetor de entrada do sistema (padrão de treinamento)
- $y(k)$  - Saída desejada do sistema para a entrada  $\vec{x}(k)$
- $C$  - Número de regras nebulosas do sistema
- $n$  - Número de elementos do vetor de entrada  $\vec{x}(k)$
- $\vec{a}^i(k)$  - Vetor de parâmetros das partes consequentes da  $i$ -ésima regra para o  $k$ -ésimo padrão de treinamento
- $\vec{\theta}_j^i$  - Vetor de parâmetros das partes premissas da  $i$ -ésima regra para o  $j$ -ésimo elemento do vetor de entrada  $\vec{x}(k)$
- $\hat{y}(k)$  - Saída do sistema no instante  $k$
- $w^i(k)$  - Grau de ativação da  $i$ -ésima regra no instante  $k$
- $u_{ik}$  - Grau de pertinência do  $k$ -ésimo padrão de treinamento para a  $i$ -ésima regra
- $r_{ik}$  - Erro entre a  $k$ -ésima saída desejada do sistema modelado e a saída da  $i$ -ésima regra com a  $k$ -ésima entrada
- $d_{ik}$  - Distância entre o  $k$ -ésimo vetor de entrada e o centro do  $i$ -ésimo *cluster*
- $\beta_i$  - Centro do  $i$ -ésimo *cluster*
- $\lambda_k$  - Multiplicadores de Lagrange
- $P_i(k)$  - Matriz de covariância
- $A_j^i(\vec{\theta}_j^i)$  - Conjunto nebuloso da  $i$ -ésima regra para a entrada  $x_j(k)$
- $e(k)$  - Diferença entre a saída desejada  $y(k)$  e a saída do modelo TSK  $\hat{y}(k)$



# Trabalhos Publicados Pelo Autor

1. L.M.C. Sousa, F.H.T. Vieira, L.L. Lee. “A Fuzzy Approach for Adaptive Control of MPLS Network Traffic Flows”. *VI International Telecommunications Symposium (ITS’2006)*, Fortaleza, Ceará, Brasil, Setembro 2006.
2. L.M.C. Sousa, F.H.T. Vieira, L.L. Lee. “Controle Fuzzy Adaptativo de Fluxos de Tráfego”. *XVI Congresso Brasileiro de Automática (CBA’2006)*, Salvador, Bahia, Brasil, Outubro 2006.
3. L.M.C. Sousa, F.H.T. Vieira, L.L. Lee. “Adaptive Fuzzy Modeling for Predictive Control of High-Speed Network Traffic”. *International Workshop on Telecommunications (IWT’2007)*, Santa Rita do Sapucaí, Minas Gerais, Brasil, Fevereiro 2007.



# Capítulo 1

## Introdução

O congestionamento em redes de computadores é um problema de compartilhamento de recursos que ocorre quando usuários inserem na rede uma quantidade maior de dados do que a mesma é capaz de atender [1]. Esse problema é uma questão fundamental em engenharia de tráfego, pois sua ocorrência prejudica o desempenho das redes e pode degradar os parâmetros de qualidade de serviço (QoS, do termo em inglês *Quality of Service*). Tal evento é, portanto, indesejável e pode tornar-se ainda mais crítico nos próximos anos devido ao crescimento do número de usuários, do número de aplicações, principalmente as de multimídia, e da complexidade das mesmas [1]. Em [2], Awduche afirma que o congestionamento pode ter duas causas básicas: a insuficiência de recursos da rede para acomodar os fluxos de tráfego de entrada e a má distribuição do tráfego na rede, ocasionando subutilização em um conjunto de recursos e sobrecarga em outro.

Acreditava-se que a solução para o congestionamento consistia somente de um incremento significativo das velocidades de transmissão dos enlaces (canais), de um acréscimo da capacidade de processamento dos nós de comunicação e da utilização de grandes *buffers* para o armazenamento de pacotes. Porém, Jain mostrou em [3] que esses procedimentos, por si só, não constituem uma solução eficiente, visto que o congestionamento é um problema dinâmico e não pode ser resolvido com soluções estáticas como as acima mencionadas. Em redes IP, por exemplo, o congestionamento pode ocorrer mesmo com a utilização de um *buffer* infinito. Isso ocorre porque a fila no *buffer* se tornaria tão longa que os pacotes teriam que ser retransmitidos pelos usuários, à medida que os mesmos fossem descartados devido ao longo atraso na fila.

Vários mecanismos de gerenciamento e controle do congestionamento são encontrados na literatura e, basicamente, podem ser classificados em dois grupos: o primeiro, que aumenta a disponibilidade dos recursos através da reconfiguração dinâmica dos mesmos e o segundo, mais usado, que reduz a demanda de tráfego ao nível da disponibilidade dos recursos.

No primeiro grupo de mecanismos de controle citado, os usuários da rede não precisam ter co-

nhecimento da ocorrência do congestionamento, pois somente a rede é responsável pela solução. A abordagem deste grupo de mecanismos é difícil de ser implementada e os custos envolvidos, por serem altos, inibem sua utilização [4]. No segundo grupo, os mecanismos de controle de congestionamento são subdivididos em: aqueles que descartam pacotes, e aqueles que limitam a taxa de entrada de pacotes na rede. Contudo, é possível que nos mecanismos de controle que se baseiam no descarte de pacotes, ocorra um aumento do congestionamento devido ao excesso de retransmissões dos pacotes descartados.

As alternativas de controle mencionadas até aqui podem também tratar o problema de forma pró-ativa, prevendo o comportamento do tráfego e os possíveis congestionamentos de modo que medidas de controle possam ser tomadas. Tais medidas vão desde o descarte de pacotes ao provimento de banda adicional aos canais (capacidade do servidor).

Desse modo, duas estratégias distintas de controle de congestionamento são definidas: recuperação e prevenção [4][5][6]. A primeira atua na rede após a ocorrência de congestionamento para resolver tal problema. A segunda trata de detectar possíveis condições que levem a congestionamentos futuros e de executar procedimentos para impedi-los. Essa estratégia de prevenção, que se baseia em antecipar situações de congestionamento, deve incluir modelos que analisem o comportamento do tráfego, o qual é muitas vezes complexo e não-linear. Deste modo, é utilizada a modelagem de tráfego, a qual imita o comportamento do sistema que o gerou, sem o conhecimento de seus mecanismos de funcionamento. Muitos estudos mostram que o modelo *fuzzy* possui vantagens sobre os modelos lineares ao descrever o comportamento de processos reais desconhecidos com características não-lineares e variantes no tempo, como é o caso dos fluxos de tráfego de redes [7][8][9]. Esse modelo é capaz de representar um sistema complexo não-linear através de uma aproximação pela combinação de vários modelos locais lineares invariantes no tempo.

Algumas técnicas de controle de congestionamento utilizando lógica *fuzzy* têm sido propostas na literatura devido à ausência de um modelo matemático preciso para estimar os parâmetros de tráfego [10]. Li e Nahrstedt propõem em [11] uma ferramenta capaz de aumentar a eficiência da adaptação dos parâmetros de QoS através de um controle dinâmico *fuzzy*. Tais autores, no entanto, focam apenas o ambiente de configuração dos parâmetros de QoS e não atuam profundamente no controle de recursos na rede. Vasilakos e Anagnostakis [12] apresentam um algoritmo de roteamento *fuzzy* para determinar o melhor caminho que um pacote deve seguir, com o objetivo de oferecer garantias de QoS. Contudo, esse algoritmo não leva em consideração as variações de largura de banda e atraso das conexões na escolha do melhor caminho. Hu e Petr propõem em [13] um algoritmo de controle preditivo de fluxos de tráfego baseado em taxa. Nesse algoritmo, o tamanho da fila no *buffer* é predito usando a lógica nebulosa e, em seguida, a taxa da fonte e os parâmetros do controlador são ajustados a partir dos valores obtidos na predição. A desvantagem de tal algoritmo é que as funções

de pertinência que caracterizam as entradas do preditor nebuloso são fixas, ou seja, tais funções não se ajustam a medida que novas entradas de dados são disponibilizadas. Em [14], um esquema de controle de congestionamento *fuzzy* para redes B-ISDN (*Broadband Integrated Services Digital Network*) é proposto. Nesse esquema, o campo EFCI (*Explicit Forward Congestion Indication*) do cabeçalho das células ATM (*Asynchronous Transfer Mode*) é utilizado para indicar a ocorrência de congestionamento num determinado nó da rede. Quando uma célula chega ao seu destino, caso ela tenha passado por uma região congestionada, o campo EFCI irá indicar tal ocorrência e o destino enviará uma célula RM (*Resource Management*) para a fonte indicando que ela deve diminuir sua taxa. Entretanto, esse esquema é considerado ineficiente, pois as células RM, que avisam às fontes do congestionamento, podem ser descartadas ou atrasadas durante o seu trajeto.

As deficiências dos controles de congestionamento anteriormente citados, a ausência de mecanismos preventivos que antecipem situações de congestionamento e a complexidade envolvida nos atuais modelos de tráfego motivam novas abordagens de controle e gerenciamento de congestionamento. Um dos objetivos deste trabalho é a análise do comportamento do tráfego e das técnicas *fuzzy* de modelagem visando a predição do comportamento da fila no *buffer* e o controle da taxa de entrada.

O trabalho desenvolvido nesta dissertação leva em consideração os seguintes preceitos. Primeiro, como a maioria dos congestionamentos ocorre nos nós da rede (roteadores, *switches*, etc.), estes devem ser os responsáveis pela monitoração da carga e do congestionamento. Segundo, para evitar a ocorrência do congestionamento e a degradação dos parâmetros de QoS sem o desperdício de recursos, o controle e gerenciamento do congestionamento é feito de forma preditiva. Por fim, como a causa do congestionamento tem característica aleatória, 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. Para caracterizar os fluxos de tráfego utilizamos o modelo *fuzzy* Takagi-Sugeno (TSK), o qual faz uso da lógica nebulosa, ferramenta bastante utilizada em função das características de incerteza e imprecisão inerentes à tais fluxos.

A presente dissertação é dividida em sete capítulos, além das referências bibliográficas e apêndices. No Capítulo 2, são apresentados diferentes esquemas de controle de congestionamento divididos em várias categorias: orientado a janela e orientado a taxa; *unicast* e *multicast*; taxa única e multi-taxa; fim-a-fim e apoiado pelo roteador e orientado a transmissor ou a receptor. Em seguida, os mecanismos de realimentação que informam o nível de congestionamento na rede e as políticas de gerenciamento de filas e descartes de pacotes também são definidas. Por fim, uma análise dos tipos de serviços das redes de alta velocidade e os parâmetros de qualidade de serviço exigidos para tais serviços são detalhados.

No Capítulo 3, alguns conceitos e propriedades fundamentais da teoria nebulosa são introduzidos:

os conjuntos nebulosos, as funções de pertinência, as operações algébricas, as relações e os sistemas nebulosos.

No Capítulo 4, detalhamos o esquema de controle de congestionamento proposto. Nesse esquema, um algoritmo de treinamento adaptativo para o modelo *fuzzy* TSK é proposto com o intuito de prever adaptativamente o comportamento dos fluxos de tráfego, mais especificamente, do tamanho da fila no *buffer*. Com os parâmetros obtidos na predição, a taxa da fonte é controlada de tal forma que o tamanho da fila no *buffer* seja igual ou inferior a um nível de referência e que a sua variância se torne mínima possível.

No Capítulo 5, é realizada uma avaliação do preditor proposto através de simulações com traços de tráfego reais. Os preditores utilizados na comparação com o preditor proposto são apresentados e as comparações são feitas com base em medidas de desempenho bastante utilizadas na literatura. Para avaliar as características fractais dos traços utilizados nas simulações, uma ferramenta existente na literatura conhecida como diagrama multiescala linear é aplicada. Por fim, o desempenho do preditor proposto é analisado utilizando os traços de tráfego monofractais e multifractais.

O Capítulo 6 avalia o controlador adaptativo de fluxos de tráfego proposto. Alguns cenários de simulação são considerados, variando-se o número de nós da rede e os traços de tráfego utilizados, e o algoritmo proposto de controle de fluxos é empregado em cada nó dos cenários considerados. Comparações com métodos de controle de congestionamento já conhecidos são realizadas e medidas de desempenho, como taxa de perda, utilização do enlace e vazão dos fluxos de tráfego, são analisadas para cada esquema de controle. Por fim, no Capítulo 7 são apresentadas as conclusões e considerações finais.

## Capítulo 2

# Controle de Congestionamento e Redes de Serviços

### 2.1 Introdução

Congestionamento em redes de computadores é, basicamente, um problema de compartilhamento de recursos [1] que ocorre quando equipamentos de usuários ou nós de entrada inserem na rede uma quantidade maior de dados do que a rede é capaz de atender. Para solucionar esse problema são propostas duas abordagens: prover a rede de recursos de tal forma a satisfazer as demandas de pico e reduzir a demanda de tráfego de acordo com os recursos disponíveis.

Desde que não é considerado economicamente viável utilizar a abordagem que disponibiliza os recursos de acordo com as demandas de pico [3], várias técnicas de controle de congestionamento que se baseiam na redução da demanda têm sido propostas na literatura. Nessas técnicas, alguns mecanismos de realimentação que indicam o nível de congestionamento da rede e políticas de gerenciamento de filas dos roteadores são incorporados a fim de se obter resultados confiáveis [6].

Nesse capítulo, os princípios básicos dos mecanismos de controle e gerenciamento do congestionamento serão descritos através da classificação de suas propriedades. Em seguida, os mecanismos de realimentação do nível de congestionamento na rede e as políticas de gerenciamento de filas e de descarte de pacotes nas filas serão detalhados. Ao final do capítulo, é realizada uma análise das redes de dados atuais definindo os tipos de serviços e os requisitos de qualidade de serviço.

## 2.2 Princípios Básicos dos Mecanismos de Controle de Congestionamento

Na literatura são encontrados vários trabalhos sobre controle e gerenciamento do congestionamento. Esse campo de pesquisa têm se mostrado bastante vasto e ainda longe de ser esgotado devido à frequente necessidade de atualizações nos mecanismos de controle de congestionamento. O motivo para tais atualizações está nos complexos e conflitantes requisitos que esses mecanismos devem atender, os quais forçam compromissos de desempenho e acabam deixando espaços para otimizações/atualizações.

Em geral, esses mecanismos de controle e gerenciamento do congestionamento podem ser classificados de acordo com algumas propriedades, tais como: orientado a janela de transmissão e orientado a taxa de transmissão; *unicast* e *multicast*; taxa única e multi-taxa; fim-a-fim e apoiado pelo roteador; orientado a transmissor ou a receptor [15]. Nas seções seguintes, cada uma dessas propriedades será analisada com maiores detalhes.

### 2.2.1 Controle de Congestionamento Baseado em Janela de Transmissão e Baseado em Taxa de Transmissão

Um critério de classificação dos mecanismos de controle de congestionamento é se os mesmos adaptam a carga oferecida na rede baseado em janela ou em taxa de transmissão.

Nos mecanismos de controle de congestionamento baseados em janela, o número de bytes que podem ser transmitidos simultaneamente na rede por um usuário é limitado e denominado de janela de transmissão. Essa janela representa uma fatia da transmissão total, visto que é referente a apenas um usuário.

Cada pacote transmitido por um usuário consome um número de bytes da sua janela de transmissão. Enquanto o número de bytes transmitidos for menor que o tamanho da janela, o transmissor pode enviar mais dados. Quando os dados chegam no receptor, este envia mensagens para o transmissor confirmando o recebimento. Conforme o transmissor for recebendo essas mensagens, ele vai avançando a janela, permitindo a transmissão de novos bytes. Quando o número de bytes transmitidos for igual ao tamanho da janela, o transmissor deve parar de enviar bytes até que receba alguma confirmação por parte do receptor, permitindo a ele avançar a janela e, assim, transmitir novos bytes.

Em situações de congestionamento, o tamanho da janela é reduzido, provocando uma diminuição na taxa de transmissão. Quando o problema do congestionamento é resolvido, o tamanho da janela é incrementado e a taxa de transmissão aumenta. Esse comportamento provoca, portanto, uma adaptação da taxa de transmissão de acordo com as condições da rede.

Exemplos de algoritmos de controle de congestionamento baseados em janela, além do TCP, são o TEAR (*TCP Emulation At Receivers*) e o RAP (*Rate Adaptation Protocol*) [16][17].

Os mecanismos de controle de congestionamento baseados em taxa buscam uma adaptação dinâmica de sua taxa de transmissão de acordo com o nível de congestionamento na rede. Podem ser subdivididos em mecanismos de controle de congestionamento AIMD (*Additive Increase Multiplicative Decrease*) simples ou mecanismos de controle de congestionamento baseados em modelo TCP [15]. Os mecanismos AIMD simples imitam o comportamento do TCP, criando um comportamento de aumento ou diminuição de taxa de transmissão parecido com um fluxo TCP equivalente. Isso gera uma onda estilo "dente de serra", similar ao protocolo TCP, dificultando sua utilização em aplicações do tipo *streaming* de vídeo. Os mecanismos baseados em modelo TCP adaptam a taxa de transmissão de acordo com a vazão média do TCP em um determinado período de tempo. Por se basear na média, tal mecanismo produz resultados bem mais estáveis do que os obtidos com o AIMD simples. Exemplos de mecanismos de controle de congestionamento baseados em taxa são o TFRC (*TCP-Friendly Rate Control Protocol*) e o TFMCC (*TCP-Friendly Multicast Congestion Control*) [18][19].

### 2.2.2 Unicast e Multicast

Os mecanismos de controle de congestionamento podem ser classificados de acordo com o tipo de conexão estabelecida na rede: *unicast* ou *multicast*. Na conexão *unicast*, também conhecida como ponto-a-ponto, os dados são enviados de um emissor para um único receptor. Por outro lado, nas conexões *multicast*, a entrega de informação ocorre para múltiplos destinatários simultaneamente. Nesse tipo de conexão, os dados só passam por um link uma única vez e somente são duplicados quando o link para os destinatários se divide em duas direções.

O controle de congestionamento para conexões *multicast* tem algumas características relevantes que o torna mais complexo do que o controle de congestionamento para conexões *unicast*. Dentre elas citamos:

- Escala - o controle de congestionamento nas conexões *multicast* deve funcionar tanto para um receptor como para centenas de receptores simultâneos;
- Topologia e congestionamento - os receptores podem estar localizados em enlaces rápidos, lentos, congestionados ou livres, criando o problema da escolha da taxa de transmissão na qual os pacotes devem ser enviados. As opções são adaptar a taxa ao transmissor mais lento, ao mais rápido, utilizar o valor da taxa utilizada pela maioria dos usuários ou utilizar transmissão em camadas;
- Perda de pacotes - receptores localizados em enlaces diferentes podem obter taxa de perda de

pacotes diferentes e, desta forma, o mecanismo de controle de congestionamento do transmissor deve saber como interpretar as diferentes perdas de forma apropriada.

### 2.2.3 Taxa Única e Multi-taxa

Os mecanismos de controle de congestionamento podem ser baseados em transmissão de taxa única ou multi-taxa conforme as seguintes definições.

- Taxa única: a transmissão é efetuada para todos os receptores na mesma taxa de transmissão.
- Multi-taxa: a transmissão é enviada em mais de uma taxa simultaneamente através da divisão em camadas, nas quais o transmissor divide os dados e os transmite para diferentes grupos *multicast*. Cada receptor pode individualmente se unir a tantos grupos *multicast* quanto a largura de banda entre ele e o transmissor permitir. A qualidade de recepção será proporcional ao número de grupos aos quais o receptor estiver unido.

As transmissões *unicast* sempre utilizam taxa única, pois se baseiam na comunicação de um transmissor com um receptor, não fazendo sentido portanto efetuar transmissão com mais de uma taxa.

Em transmissões *multicast*, entretanto, é possível classificar os esquemas de controle de congestionamento em taxa única ou multi-taxa, ou seja, ou o transmissor utiliza uma única taxa de transmissão para todos os receptores, ou utiliza várias taxas simultaneamente, cada uma em um grupo *multicast* diferente. Nesse caso, é utilizada a transmissão *multicast* em camadas.

### 2.2.4 Controle de Congestionamento Apoiado pelo Roteador e Fim-a-fim

Os mecanismos de controle de congestionamento também podem ser classificados de acordo com o local onde tal controle é efetuado: nos equipamentos centrais (roteadores) ou nas estações finais (controle fim-a-fim).

O controle de congestionamento apoiado pelos equipamentos centrais (roteadores) se fundamenta no fato que é através desses equipamentos que passa todo o tráfego da rede, e não há melhor lugar para garantir a equidade entre os fluxos de tráfego de diferentes sessões.

O mecanismo de controle de congestionamento fim-a-fim não precisa de qualquer tipo de apoio adicional por parte dos roteadores. Esses mecanismos são conhecidos por possuírem controle de congestionamento nas estações finais e utilizam certo apoio da rede para inferirem o nível de congestionamento atual. O apoio da rede pode ser a reserva de recursos, permitindo às estações reagirem

a essa informação, aumentando ou diminuindo sua taxa de transmissão<sup>1</sup> ou algum mecanismo de realimentação que informa o nível atual de congestionamento na rede<sup>2</sup>.

Existem diversos métodos através dos quais a rede fornece realimentação sobre seu nível de congestionamento, entre eles podem ser citados desde simples descartes de pacotes até mecanismos tipo ECN (*Explicit Congestion Notification*). Alguns desses métodos serão analisados em maiores detalhes na seção 2.3.

### 2.2.5 Controle de Congestionamento Orientados ao Transmissor e Orientados ao Receptor

Os mecanismos de controle de congestionamento fim-a-fim definidos na seção anterior podem ser divididos em dois grupos: orientados ao transmissor e orientados ao receptor. Os mecanismos de controle de congestionamento orientados ao transmissor utilizam as informações da rede para que o transmissor ajuste sua taxa ou janela de transmissão de acordo com o nível de congestionamento inferido. A única função dos receptores é enviar uma informação de realimentação sobre as condições da rede conforme os dados chegam.

Nos mecanismos orientados ao receptor, é utilizada a transmissão em camadas e, de acordo com o nível de congestionamento da rede, o receptor decide a quantos grupos *multicast* ele deve se unir.

## 2.3 Mecanismos de Realimentação do Nível de Congestionamento

Existem diversos mecanismos utilizados para que o transmissor ou o receptor descubra o nível de congestionamento atual na rede e adapte a taxa de transmissão adequadamente. Alguns desses mecanismos de realimentação do nível de congestionamento estão implícitos na própria transmissão, como, por exemplo, o descarte de pacotes pelos roteadores ou o aumento no atraso da chegada dos mesmos. Outros mecanismos informam explicitamente o nível de congestionamento da rede, como o ECN (*Explicit Congestion Notification*) e o ICMP *source quench* (*Internet Control Message Protocol*). Tais mecanismos serão detalhados a seguir.

---

<sup>1</sup>Esse caso caracteriza o controle de fluxo fim-a-fim de laço aberto, pois a banda está descrita e supostamente garantida pelos equipamentos da rede, não necessitando realimentação. Basta utilizar um conformador de tráfego para manter a transmissão dentro dos valores reservados.

<sup>2</sup>Esse caso caracteriza o controle de fluxo fim-a-fim de laço fechado, pois as estações devem se adaptar conforme a realimentação informada pela rede.

### 2.3.1 Descarte de Pacotes

O procedimento de descarte de pacotes é atualmente a forma mais comum de realimentação do nível de congestionamento das redes. Entretanto, nem sempre um pacote descartado indica congestionamento, principalmente quando se trata de transmissão *wireless*, onde maiores taxas de erros são comuns [20].

O roteador descarta pacotes quando detecta uma condição de congestionamento, e o momento desse descarte pode variar, dependendo da política de descarte utilizada. *Tail drop* e RED (*Random Early Detection*) são exemplos de tais políticas, as quais serão descritas posteriormente.

### 2.3.2 Notificação de Congestionamento Explícito - ECN

Esse mecanismo de realimentação do nível de congestionamento propõe a utilização nos roteadores de uma política de controle de congestionamento que marca o pacote através dos bits de ECN (*Explicit Congestion Notification*) no cabeçalho IP, indicando que esse pacote passou por uma região congestionada. Quando o pacote chega no receptor, o mesmo envia uma mensagem avisando ao transmissor do congestionamento. O transmissor então reage à essa mensagem diminuindo sua taxa de transmissão. A vantagem deste mecanismo de realimentação é que não é necessário haver descarte de pacotes para reduzir o congestionamento. Tal mecanismo está descrito na RFC (do inglês *Request for Comments*) 2481 [21].

### 2.3.3 ICMP Source Quench

O ICMP (*Internet Control Message Protocol*) é um protocolo integrante do protocolo IP, definido pelo RFC 792 e utilizado em ocasiões onde o roteador deve enviar mensagens de erro ou de controle para o transmissor, como por exemplo, avisando que o IP destino é inacessível, ou que o TTL<sup>3</sup> (*Time To Live*) expirou. Além disso, o ICMP define mensagens de *echo*, as quais são utilizadas para testar a conectividade até um determinado destino [22].

Muitas mensagens ICMP são definidas na literatura e uma delas é o *source quench*. Essa mensagem é enviada para o transmissor em duas situações: quando os pacotes estão chegando muito rapidamente para serem processados ou quando o roteador descarta um pacote por não ter espaço suficiente no *buffer* para inserí-lo. O objetivo dessa mensagem é solicitar uma redução na taxa de

---

<sup>3</sup>É um campo do pacote IP que é utilizado para limitar o número de roteadores por onde um determinado pacote pode passar. Cada roteador por onde um determinado pacote IP trafega decrementa o número do campo TTL antes de passá-lo para frente. Se o valor do campo TTL chegar a zero, o roteador não envia mais o pacote IP, e sim um outro pacote ICMP para a origem avisando que o pacote IP original teve o seu TTL=0 e por isso não pôde ser mais transmitido. Esta é uma forma de evitar que os pacotes permaneçam na rede por tempo infinito.

transmissão. Nesse caso, o transmissor diminui sua taxa de transmissão até parar de receber mensagens de *source quench*, incrementando-a gradualmente até voltar a receber esse tipo de mensagem. Uma solução alternativa seria o envio dessas mensagens quando a capacidade da fila estiver se aproximando do limite, ao invés de esperar o esgotamento total dos recursos, evitando o descarte de pacotes [22].

## 2.4 Políticas de Gerência de Filas nos Roteadores

O modelo utilizado para estudar a gerência das filas dos roteadores consiste de um servidor e de uma seqüência de solicitações de serviço (pacotes chegando através das interfaces de entrada) [23]. Neste modelo, o servidor não possui qualquer controle sobre a taxa de chegada das solicitações, da mesma forma que o tempo utilizado para atender cada solicitação pode variar, dependendo, por exemplo, da capacidade de processamento do servidor e da velocidade das interfaces de saída. Assim, enquanto o servidor estiver atendendo uma solicitação, outras solicitações podem chegar, de modo que o servidor necessitará enfileirá-las, para atendê-las quando estiver liberado.

As políticas de gerência de filas podem variar dependendo do algoritmo utilizado no servidor, e sua função é determinar qual solicitação será atendida quando o servidor tiver terminado de atender a solicitação anterior. Algumas políticas preservam a ordem de chegada dos pacotes, como por exemplo a FIFO (*First-In First-Out*). Outras políticas alocam prioridades a certos tipos de fluxos de tráfego, como as do tipo CBQ (*Class-Based Queuing*). Dentro dessas políticas de gerência existem diversas variantes, as quais serão detalhadas nas próximas seções.

### 2.4.1 Fila FIFO

A forma mais simples de fila é a do tipo FIFO (*First-In First-Out*) também chamada de FCFS (*First Come First Served*) [24]. Nesse caso, o roteador analisa somente a ordem de chegada dos pacotes para decidir qual será o próximo pacote a ser transmitido. A ordem do processamento das solicitações é, obrigatoriamente, a mesma ordem na qual elas chegam.

### 2.4.2 Fila de Prioridade Estrita - SP

A fila de Prioridade Estrita ou *Strict Priority* (SP) considera o fato de que certas classes de tráfego podem ser identificadas e transferidas para o início da fila de saída, de tal forma que essas classes sejam sempre transmitidas antes de outras seguindo o critério de prioridade adotado.

No algoritmo de escalonamento SP, para cada prioridade existirá uma fila de serviço associada, e um pacote somente é retirado de uma fila quando todas as filas com prioridade superior estiverem

vazias. Dentro de cada prioridade os pacotes são processados como numa fila FIFO, obedecendo a ordem de chegada de cada pacote na fila.

Esse tipo de encaminhamento tem a vantagem de ser relativamente simples, porém os fluxos de baixa prioridade podem ficar longos períodos sem atendimento caso exista uma continuidade de pacotes de alta prioridade passando pelo roteador.

### 2.4.3 Fila Baseada em Classes - CBQ

A fila do tipo CBQ (*Class-Based Queuing*) é uma variação da fila de prioridade estrita SP que evita o não atendimento das filas de baixa prioridade. Ao invés de atribuir prioridades absolutas (A antes de B, B antes de C), o sistema CBQ agrupa o tráfego em classes, e atribui um peso ou métrica para cada classe de tráfego [24]. Ajustando o peso, é possível ajustar a prioridade de determinada classe em relação às outras. Caso haja banda não utilizada por alguma classe, esta banda pode ser aproveitada pelas outras classes, proporcionalmente ao peso de cada uma.

O CBQ é uma tentativa de prover justiça priorizando o serviço para certos tipos de tráfego, porém não permitindo que uma classe possa monopolizar os recursos do sistema. Deste modo, todas as classes receberão um mínimo do serviço oferecido.

### 2.4.4 Fila Justa - FQ e Fila Justa Ponderada - WFQ

O algoritmo de escalonamento justo ponderado WFQ (*Weighted Fair Queuing*), também conhecido com PGPS (*Packet Generalized Processor Sharing*), é uma variação da fila justa FQ (*Fair Queuing*). Em ambos os casos, cada classe de tráfego tem uma fila FIFO separada. No caso FQ, para uma capacidade de enlace  $R$ , em um determinado instante, as  $N$  classes de tráfego ativas (com filas não-vazias) serão servidas simultaneamente, cada uma com uma taxa média de  $R/N$ . Assim, desde que cada classe de tráfego tem sua própria fila, uma classe dita "mal comportada" ou seja, que enviou maiores pacotes ou mais pacotes por segundo, prejudicará somente a ela mesma e não às outras classes.

O algoritmo WFQ atribui pesos às classes de tráfego. Assim, se  $N$  classes de tráfego estão ativas com pesos  $w_1, w_2, \dots, w_N$ , uma determinada classe " $i$ " terá uma taxa média de  $Rw_i/(w_1 + w_2 + \dots + w_N)$ , ou seja, cada classe de tráfego será atendida em proporção direta ao seu peso. Se uma classe de tráfego necessita de menos taxa do que foi alocado à ela, o excedente será dividido pelas outras classes de tráfego, proporcionalmente ao seu peso.

## 2.5 Políticas de Descarte de Pacotes nas Filas dos Roteadores

Quando a taxa de chegada de pacotes excede a taxa de atendimento dos mesmos, a fila expande. Em um determinado instante, o número de pacotes na fila pode estar tão grande que o roteador deverá tomar a atitude de marcar ou descartar algum pacote. Esse comportamento do roteador pode servir como uma realimentação para que o transmissor diminua sua taxa de transmissão e restabeleça o equilíbrio na rede. O instante de início de marcação (ou descarte de pacotes) não ocorre necessariamente quando a fila estiver cheia, como será visto nas políticas de descarte a seguir.

### 2.5.1 Descarte Caudatório (*Tail Drop*)

O algoritmo de descarte caudatório ou *tail drop* normalmente é utilizado em conjunto com filas FIFO. Este algoritmo descarta os pacotes quando o *buffer* está cheio sem espaço para mais pacotes. O pacote escolhido para descarte é sempre o último a chegar, ou seja, o pacote que ficaria no fim da fila. A seqüência de descartes continua até que haja a liberação de espaço no *buffer*, conseguida quando um pacote é atendido e sai do início da fila.

O descarte no fim da fila é bastante eficiente em ambientes onde a capacidade da rede é superior à demanda, fazendo com que a fila seja utilizada somente para armazenar rajadas transientes de tráfego [23]. Entretanto, em ambientes com congestionamento, esse tipo de algoritmo possui três desvantagens principais [20][23]:

- Em períodos de congestionamento, o *buffer* se mantém cheio provocando atrasos até que as estações terminais detectem tal congestionamento, visto que o mesmo não é percebido até que seja detectada a perda do pacote. Em protocolos com realimentação, até que o transmissor receba a mensagem de realimentação de pacote perdido e reduza sua taxa de transmissão, o congestionamento persiste, ocasionando atraso e novas perdas;
- Outro problema é a questão do sincronismo entre os fluxos de tráfego. Quando o *buffer* enche, muito provavelmente serão descartados pacotes de diversas sessões, fazendo com que todas as sessões envolvidas reduzam sua taxa de transmissão simultaneamente. Isso pode levar à redução na utilização do enlace e à subutilização dos recursos da rede.
- A ausência de um comportamento de descarte buscando tráfego equitativo entre as sessões. Uma fonte mal comportada (com um grande número de rajadas de pacotes ou sem controle de fluxo) pode saturar a fila, forçando o escalonador a descartar pacotes de uma fonte bem comportada.

### 2.5.2 Detecção Precoce Aleatória (*Random Early Detection*) - RED

O algoritmo de detecção precoce aleatória ou RED é um algoritmo de gerenciamento ativo de filas que visa evitar a ocorrência do congestionamento agindo antes que o *buffer* fique cheio [20]. Esse comportamento é considerado pró-ativo e permite aos roteadores a informação de quando e quantos pacotes serão descartados. Tal algoritmo foi sugerido pelo IETF (*Internet Engineering Task Force*) para ser utilizado nos roteadores *best-effort* da Internet (RFC 2309)[25].

Dentre as vantagens do RED temos a diminuição do número de pacotes descartados no roteador, um atraso fim-a-fim menor e a eliminação do esgotamento do *buffer*, permitindo quase sempre a existência de espaço para a entrada de um novo pacote na fila.

No algoritmo RED, o roteador monitora a informação do tamanho médio da fila no *buffer*. Quando esse tamanho excede um determinado limite, alguns pacotes são marcados ou descartados, funcionando como um filtro. A probabilidade de descarte aumenta à medida que a média estimada do tamanho da fila aumenta. Pequenos picos de tráfego podem passar por tal filtro sem experimentar uma probabilidade de descarte muito alta (pois o algoritmo se baseia na média), enquanto que condições de maiores cargas causam continuamente maiores taxas de descarte [20] [26].

As decisões do momento e do número de pacotes a descartar são dadas por duas variáveis: um limiar mínimo do tamanho da fila no *buffer*, que determina o momento de iniciar o descarte de pacotes (com baixa probabilidade), e um limiar máximo, que determina o máximo de probabilidade de descarte que o algoritmo irá realizar. Caso a probabilidade máxima seja configurada em 100%, todos os pacotes acima do limiar máximo serão descartados ou marcados. Até que o tamanho médio da fila atinja o limiar mínimo, o roteador considera que não existe congestionamento, redirecionando os pacotes normalmente. Quando o tamanho médio da fila ultrapassa o limiar mínimo, e aumenta até atingir o tamanho máximo da fila, o algoritmo escolhe aleatoriamente um determinado número de pacotes e os descarta. O número de pacotes descartados aumenta linearmente à medida que o tamanho médio da fila aumenta.

## 2.6 Análise dos Tipos de Serviços em Redes de Dados

Nesta seção, inicialmente serão apresentadas as redes chaveadas por pacotes e as redes chaveadas por circuitos. Em seguida, o procedimento de configuração dos parâmetros de QoS de uma nova conexão será detalhado. Com o intuito de garantir o cumprimento desses parâmetros de QoS, a rede monitora a taxa da fonte através de um controle de fluxos. Tal controle pode ser de malha aberta ou de malha fechada, como veremos a seguir.

### 2.6.1 Redes de Pacotes e de Circuitos

Tradicionalmente, as redes de dados chaveadas por pacotes provêm os serviços chamados melhor esforço (*best effort*). Esse tipo de serviço tem como principal característica a não garantia de que os dados serão entregues em um tempo específico, nem mesmo que eles serão entregues por completo. Um dos exemplos mais interessantes de serviços *best effort* é o serviço oferecido pelo protocolo IP (*Internet Protocol*) na Internet, o qual é projetado para entrega sem conexão de dados de comprimentos variáveis. Em tal protocolo, não há garantias de qualidade de serviço para as conexões, e os parâmetros de QoS variam de acordo com o nível de congestionamento da rede. Desse modo, o controle de congestionamento é bastante simples: o tráfego é roteado para evitar pontos congestionados e os dados são simplesmente descartados quando necessário. Esse tipo de serviço melhor esforço IP é muito apropriado para aplicações de tempo não-real, tais como e-mail eletrônico e transferência de arquivos.

Por outro lado, existem também os serviços chaveados por circuitos com largura de banda fixa. Esses serviços são característicos das redes de telecomunicações e assumem a responsabilidade de garantir largura de banda e QoS para cada conexão, visto que é necessário um certo nível de qualidade de serviço para voz e vídeo. Para transportar esses serviços são utilizadas redes de alta velocidade que permitem pequeno atraso de chaveamento e altas vazões, tais como ATM (*Asynchronous Transfer Mode*), ISDN (*Integrated Services Digital Network*), MPLS (*Multiprotocol Label Switching*), entre outras. Para garantir QoS em suas conexões de tempo real, essas redes avaliam e controlam medidas de desempenho, como por exemplo, atraso e taxa de perda de pacotes [14] [27]. A importância do controle dessas medidas é percebida no exemplo de comunicações de voz, onde um sinal de fala é amostrado periodicamente e a informação da fala digitalizada é transmitida para o ouvinte, onde o sinal é reconstruído. O sinal de fala só será reconstruído de forma apropriada se as amostras forem continuamente disponibilizadas, em intervalos periódicos, no instante em que serão necessárias para a reprodução no ouvinte. As informações atrasadas não terão mais utilidade e serão descartadas, logo, é importante que os dados sejam entregues com atraso e taxa de perda de pacotes limitados. Além disso, dados que chegam adiantados precisam ser armazenados, exigindo assim a presença de *buffers* na rede.

### 2.6.2 Qualidade de Serviço - QoS

Os parâmetros de QoS de uma conexão são especificados durante o procedimento de configuração da mesma. Ao requisitar uma conexão, o usuário deve declarar suas características de tráfego (como por exemplo, a taxa de pico) e a rede, por sua vez, baseando-se nessas declarações, realiza uma estimativa dos recursos necessários para atender a conexão (*buffer*, largura de banda, processador).

Se os recursos existentes na rede não forem suficientes, a requisição é rejeitada, caso contrário, a conexão é estabelecida. Após a conexão ser aceita, a rede provê os parâmetros de QoS especificados e o usuário limita seu tráfego de acordo com os parâmetros declarados por ele. Esse acordo é o chamado contrato de tráfego entre a rede e o usuário [14] [27]. Uma vez que a conexão é estabelecida, a fonte de tráfego não é mais controlada e a rede apenas monitora a taxa da fonte para garantir que o contrato de tráfego seja obedecido. Esse monitoramento é realizado pelo UPC (Usage Parameter Control) ou Controle de Parâmetro de Uso, cujo controle descarta o excesso de tráfego ou classifica-o como tráfego de prioridade baixa.

Devido a ausência de realimentação no controle durante a conexão estabelecida através do contrato de tráfego rede-usuário, esta técnica é chamada de controle de laço aberto (*open-loop*). Entretanto, em algumas aplicações de tempo-real, tais como voz e vídeo, essa técnica de controle não é aplicável, visto que é muito difícil estimar as características de tráfego desses tipos de aplicações. O único parâmetro que se pode prever é a taxa pico, o qual não é o bastante para que a rede aloque recursos eficientemente. Desse modo, o controle de laço fechado (*closed-loop*) é considerado mais apropriado para essas aplicações de tempo-real. Além disso, para as aplicações de tempo não-real, as quais não necessitam de vazão garantida nem de limites de atraso específicos, um serviço de taxa de bits disponível (TBD) baseado no controle de laço fechado é definido [28].

Outros tipos de serviços também definidos são os serviços de taxa de bits variável (TBV) e taxa de bits constante (TBC), os quais são apropriados para um grande número de aplicações [28]. O serviço TBV, com característica de rajadas aleatórias, oferece conexões para aplicações de vídeo e áudio em tempo real e não-real com largura de banda variável. Os pacotes desse tipo de serviço são gerados em intervalos arbitrários (até a taxa pico) e entregues com atraso e TPB<sup>4</sup> (Taxa de Perda de Bytes) limitados de acordo com os valores requeridos pela aplicação. O serviço TBC pode ser considerado um caso especial do serviço TBV, onde sua taxa de pico é igual a taxa média. Esses serviços são usados em aplicações de tempo-real com largura de banda fixa, tais como, redes chaveadas por circuitos (Exemplo: canal de voz de 64 kbyte/seg). Os pacotes do serviço TBC são gerados em intervalos periódicos regulares e entregues com atrasos estritamente limitados.

Os serviços de taxa de bits disponível são utilizados em uma classe mais estreita de aplicações, especificamente aplicações de dados que podem ajustar sua largura de banda e tolerar atrasos imprevisíveis [14] [29]. Essas conexões TBD compartilham a largura de banda que está em excesso dos tráfegos de TBV e TBC, ou seja, dado que a capacidade total do enlace é constante, o tráfego TBD aumentará sua taxa se a taxa dos tráfegos TBC e TBV diminuírem e vice-versa. A Figura 2.1 mostra o comportamento da largura de banda dos vários serviços mencionados em um enlace de rede.

Uma parte da largura de banda do enlace é consumida pelos tráfegos TBC e TBV. O tráfego TBD

---

<sup>4</sup>Do termo em inglês *Byte Loss Rate* (BLR).

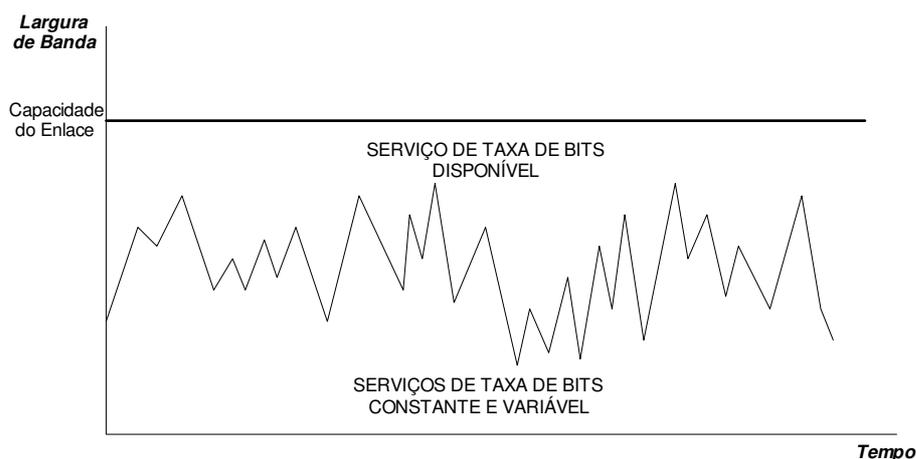


Fig. 2.1: Exemplo da Largura de Banda em Excesso para Serviços de Taxa de Bits Disponível.

somente tem acesso à largura de banda do enlace quando nenhum dos dois tráfegos anteriores estão esperando para transmitir. Podemos comparar esse tipo de situação com a de passageiros de uma companhia aérea que pega assentos vagos nos últimos instantes antes do voo. Deste modo, o tráfego TBD usa a largura de banda que está sem uso pelos demais tráfegos, aumentando a utilização do enlace sem comprometer os parâmetros de QoS das conexões TBC e TBV.

Como pode ser notado pela Figura 2.1, a largura de banda para cada conexão TBD irá flutuar dinamicamente devido às características aleatórias do tráfego TBC/TBV e também devido ao número de conexões TBD que irão compartilhar a largura de banda disponível. Assim, no serviço TBD, a largura de banda para cada conexão pode diminuir até uma taxa mínima especificada definida como MCR (*Minimum Cell Rate*). O MCR é requisitado pelo usuário e garantido pela rede de tal forma que a largura de banda de cada conexão TBD flutue entre a taxa de pico e o MCR mas nunca abaixo desse valor. O usuário não é obrigado a transmitir sempre na taxa MCR, porém isso pode acontecer algumas vezes. Tem sido discutido que algumas aplicações necessitam de garantias mínimas de vazão, por outro lado, tem-se afirmado também que as características dos serviços TBD tornam tal serviço inconsistente com garantias de recursos, visto que se trata de serviços que usam o excedente de banda tornando, assim, difícil garantir alguma vazão mínima [28].

Dado que a fonte TBD ajusta sua taxa periodicamente de acordo com a largura de banda disponível e variante no tempo, o desempenho da rede irá depender de como será realizado esse ajuste da taxa. Para isto, a fonte será informada das últimas condições da rede através de um canal de realimentação com atraso. Esse atraso de realimentação juntamente com o atraso de transmissão direta é chamado de atraso *round-trip*. Devido ao atraso, algum descasamento entre a taxa da fonte TBD e a largura de banda disponível será inevitável. Tal diferença irá exigir a presença de *buffers* na rede, onde os mesmos irão absorver o tráfego TBD durante o instante em que a taxa da fonte excederá a largura

de banda disponível. Logo, o tamanho do *buffer* necessário será proporcional a essa diferença de largura de banda e também ao atraso de realimentação. A Figura 2.2 mostra um modelo usual de realimentação no qual o atraso de *round-trip* foi considerado no caminho de realimentação.

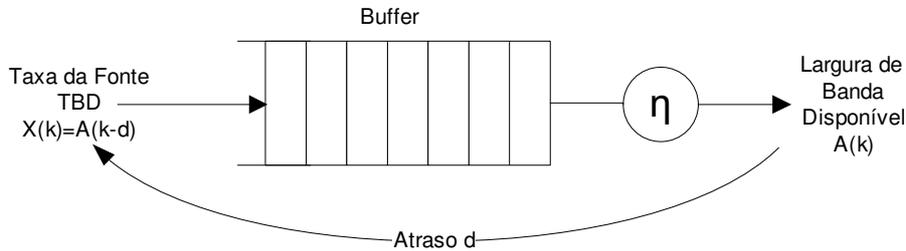


Fig. 2.2: Modelo de Realimentação da Fonte TBD com Atraso.

A largura de banda disponível  $A(k)$  pode variar arbitrariamente de zero até a capacidade do enlace  $\eta$  (supondo  $MCR=0$ ). A fonte TBD ajusta sua taxa  $X(k)$  como uma versão atrasada de  $A(k)$ , isto é,  $X(k) = A(k - d)$ , onde  $d$  é um parâmetro fixo denotando o atraso de *round-trip* da rede. Durante os períodos em que  $X(k) > A(k)$ , o tráfego excedente é absorvido pelo *buffer* provocando um aumento no seu tamanho, o qual será reduzido somente nos períodos em que  $X(k) < A(k)$ . Assim, temos que, no pior caso quando a largura de banda disponível é toda a capacidade do enlace, o *buffer* será enchido com  $\eta d$  [bits ou pacotes] antes que chegue a informação de diminuir a taxa da fonte TBD  $X(k)$ . Isso é comumente chamado de produto "largura de banda-atraso" da rede (*bandwidth-delay*) e esses dois fatores,  $\eta$  e  $d$ , representam respectivamente, a máxima diferença entre a largura de banda disponível e a taxa da fonte TBD, e a máxima duração dessa diferença.

Em redes de alta velocidade, a capacidade do enlace  $\eta$  é alta e, se a rede for extensa, o atraso  $d$  será longo, logo um grande *buffer* é necessário para evitar perda de dados, em contra partida, o tempo de espera dos pacotes no *buffer* aumenta. Dessa forma, o serviço TBD é apropriado somente para aplicações que podem, além de adaptar sua taxa de acordo com a largura de banda disponível, tolerar atrasos aleatórios.

Deve-se também ressaltar que esse tipo de serviço possui garantias de QoS e existe um acordo entre a rede e o usuário para garantir taxa de perda de pacotes mínima [28]. Contudo, esse acordo não é feito na forma de contrato de tráfego explícito, como no caso dos serviços TBV, que especifica limites superiores para a taxa da fonte e pretensões mínimas para o desempenho da rede. O serviço TBD é mais dinâmico e a natureza do acordo é mais implícita, no sentido de que a rede e o usuário cooperam juntos para controlar a dinâmica do serviço. Desse modo, é necessário um bom mecanismo de controle para ajustar os parâmetros da rede (tais como o tamanho da fila no *buffer* e a taxa de entrada) de tal forma a minimizar os problemas de congestionamento e garantir os parâmetros de QoS.

## 2.7 Considerações Finais

Neste capítulo, foram detalhadas as características dos mecanismos de controle de congestionamento. Dentre elas podemos citar as políticas de gerência de filas, as quais determinam a ordem de atendimento dos dados que esperam nas filas. Foram apresentados também alguns mecanismos de descarte de dados, cujo objetivo de tais mecanismos é manter o equilíbrio da rede em situações de congestionamento marcando ou descartando pacotes. Em seguida, uma análise das redes de dados e dos tipos de serviços foi realizada no contexto do uso da largura de banda dos enlaces da rede. O controle de laço fechado foi detalhado e o atraso de *round-trip*  $d$ , característico de controles desse tipo, foi apresentado. Por fim, foi percebido a importância de um mecanismo de controle para ajustar os parâmetros da rede (tais como o tamanho da fila no *buffer* e a taxa de entrada) de tal forma a minimizar os problemas de congestionamento e garantir os parâmetros de QoS.

No capítulo seguinte, serão apresentados alguns conceitos relevantes sobre a teoria da lógica nebulosa, visto que a mesma será utilizada como ferramenta para modelar os fluxos de tráfego do trabalho em questão.



# Capítulo 3

## Sistemas Nebulosos

### 3.1 Introdução

Aristóteles, filósofo grego (384-322 a.C.), foi o fundador da ciência da lógica, e estabeleceu um conjunto de regras rígidas para que conclusões pudessem ser aceitas como logicamente válidas. O emprego da lógica de Aristóteles levava a uma linha de raciocínio baseado em premissas e conclusões. Como por exemplo: se é observado que "*Todo ser vivo é mortal*" (premissa 1), a seguir é constatado que "*João é um ser vivo*" (premissa 2), como conclusão temos que "*João é mortal*". Desde então, a lógica tem sido binária, isto é, uma declaração é falsa ou verdadeira, não podendo ser ao mesmo parcialmente verdadeira e parcialmente falsa. Esta suposição e a lei da não contradição, que diz que "*U e não U*" cobrem todas as possibilidades, formam a base do pensamento lógico de Aristóteles. No entanto, muitas das experiências humanas não podem ser classificadas simplesmente como verdadeiras ou falsas, sim ou não, branco ou preto. Por exemplo, "*Aquele homem é alto ou baixo?*", "*A taxa de risco para aquele empreendimento é grande ou pequena?*". Um "*sim*" ou um "*não*" como resposta a estas questões é, na maioria das vezes, incompleta. Esta imperfeição intrínseca à informação representada numa linguagem natural, tem sido tratada com o uso da **lógica nebulosa**.

De forma mais objetiva e preliminar, podemos definir lógica nebulosa como sendo uma ferramenta capaz de capturar informações vagas, em geral descritas em uma linguagem humana, e convertê-las para um formato numérico, de fácil manipulação pelos computadores [30]. Considere a seguinte afirmativa: Se o tempo de um investimento é *longo* e a economia tem sido *não muito estável*, então a taxa de risco do investimento é *muito alta*. Os termos "*longo*", "*não muito estável*" e "*muito alta*" trazem consigo informações vagas. A representação destas informações vagas se dá através do uso de conjuntos nebulosos.

No presente capítulo serão apresentadas definições e exemplos de conjuntos nebulosos bem como suas particularidades e operações algébricas. A lógica nebulosa será detalhada mostrando os con-

ceitos de variável linguística, variável nebulosa e regra de inferência. E por fim, os sistemas nebulosos serão apresentados.

## 3.2 Conjuntos Nebulosos

O conceito formal de conjunto nebuloso foi introduzido por Lofti A. Zadeh em 1965 [31]. Na teoria clássica dos conjuntos, os conjuntos são ditos "*crisp*" se um dado elemento do universo em discurso<sup>1</sup> pertence ou não pertence ao referido conjunto. Na teoria dos conjuntos nebulosos existe um grau de pertinência de cada elemento a um determinado conjunto. Uma consequência direta desse fato é que, ao contrário dos conjuntos clássicos, a fronteira que separa os conjuntos nebulosos não é bem definida. Ou seja, na teoria clássica dos conjuntos, dado qualquer subconjunto  $A$  de um universo de discurso  $U$ , e qualquer elemento  $x$  pertencente a  $U$ , existe uma função

$$f_A : U \rightarrow A \quad (3.1)$$

$$\text{tal que: } \begin{cases} f_A(x) = 1 & \text{se } x \in A \\ f_A(x) = 0 & \text{se } x \notin A \end{cases}$$

Essa função  $f_A$  é chamada de função característica, pois define os elementos de  $A$ , sendo única para cada conjunto. No caso dos conjuntos nebulosos, tal função toma um valor real no intervalo fechado  $[0, 1]$ , passando a ser chamada de função de pertinência (*membership function*).

### 3.2.1 Definição de Conjuntos Nebulosos

Um conjunto nebuloso  $A$  definido num universo de discurso  $U$  é caracterizado por uma função de pertinência  $\mu_A : U \rightarrow [0, 1]$  que mapeia os elementos de  $U$  para o intervalo  $[0, 1]$ , e é representado por um conjunto de pares ordenados

$$A = \{\mu_A(x); x\} \quad (3.2)$$

Dessa forma, a função de pertinência  $\mu_A$  associa cada elemento  $x \in U$  a um número real  $\mu_A(x)$  no intervalo  $[0, 1]$ , o qual representa o grau de possibilidade de que o elemento  $x$  venha a pertencer ao conjunto  $A$ , isto é, o quanto é possível para o elemento  $x$  pertencer ao conjunto  $A$  [32] [33].

---

<sup>1</sup>O termo universo de discurso refere-se ao domínio do conjunto.

### 3.2.2 Funções de Pertinência

As funções de pertinência podem ter diferentes formas, dependendo do conceito que se deseja representar e do contexto em que serão utilizadas. Na maioria das aplicações práticas, as funções de pertinência são dos tipos gaussianas, triangulares, trapezoidais, crescentes ou decrescentes. A Figura 3.1 mostra alguns tipos básicos de funções de pertinência, onde  $a$ ,  $b$ ,  $c$  e  $d$  são parâmetros que definem uma característica particular para cada tipo de função.

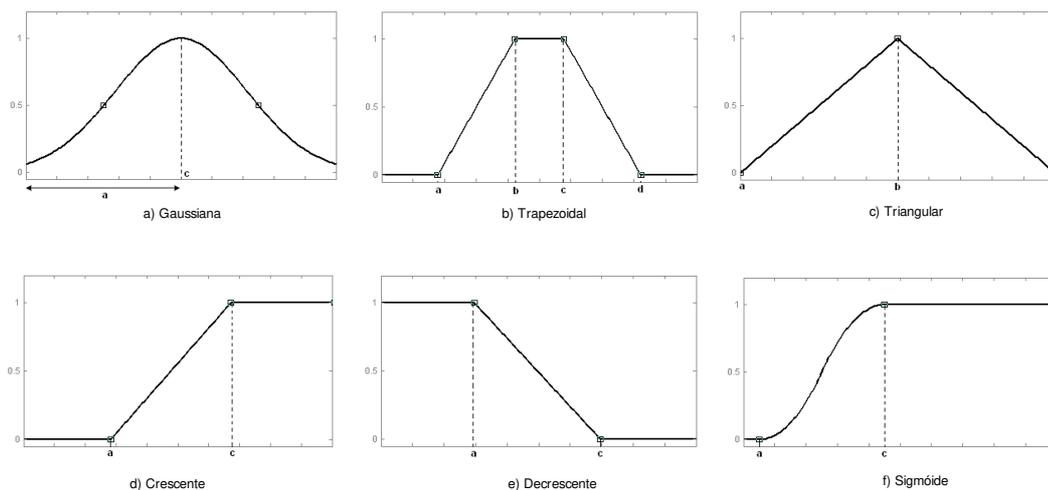


Fig. 3.1: Tipos Básicos de Funções de Pertinência.

As expressões matemáticas para as funções de pertinência apresentadas são dadas a seguir:

**Função Gaussiana:**  $f_{\text{gau}}(x, a, c) = \exp\left(-\frac{(x-c)^2}{a}\right)$

**Função Triangular:**  $f_{\text{tr}}(x, a, b, c) = \min\left(\max\left(0, \left(\frac{x-a}{b-a}\right)\right), \max\left(0, \left(\frac{c-x}{c-b}\right)\right)\right)$

**Função Trapezoidal:**  $f_{\text{tra}}(x, a, b, c, d) = \min(\text{tra}_1, \text{tra}_2)$

em que  $\text{tra}_i = \max\left(0, \min\left(1, \left(\frac{x-a}{b-a}\right)\right)\right)$  e  $\text{tra}_2 = \max\left(0, \min\left(1, \left(\frac{d-x}{d-c}\right)\right)\right)$

**Função Crescente:**  $f_c(x, a, c) = \min\left(1, \max\left(0, \left(\frac{x-a}{c-a}\right)\right)\right)$

**Função Decrescente:**  $f_d(x, a, c) = \min\left(1, \max\left(0, \left(\frac{c-x}{c-a}\right)\right)\right)$

**Função Sigmóide:**  $f_{\text{sig}}(x, a, c) = \frac{1}{1+e^{-\alpha(x-\beta)}}$  em que  $\alpha = \frac{2}{a-c} \ln\left(\frac{1-\gamma}{\gamma}\right)$  e  $\beta = \frac{a+c}{2}$

### 3.2.3 Definições e Operações com Conjuntos Nebulosos

Nessa seção, apresentamos algumas definições e operações da lógica nebulosa. Sejam  $A$  e  $B$  conjuntos nebulosos definidos no universo de discurso  $U$ , com funções de pertinência  $\mu_A(x)$  e  $\mu_B(x)$  respectivamente, então [31][34][32]:

- Dois conjuntos nebulosos são iguais,  $A = B$ , se e somente se:  $\mu_A(x) = \mu_B(x), \forall x \in U$ .
- Um conjunto nebuloso  $A$  está contido num conjunto nebuloso  $B$ , se e somente se:  $\mu_A(x) \leq \mu_B(x), \forall x \in U$ .
- A altura de um conjunto nebuloso denotado por  $\text{hgt}(A)$  é definido por:  $\text{hgt}(A) = \text{Sup}_{x \in U} \mu_A(x)$ , em que  $\text{Sup} \mu_A(x)$  é o valor máximo da função de pertinência  $\mu_A(x)$ .
- O conjunto de corte- $\alpha$ , definido como  $A_\alpha$ , é o conjunto contendo todos os elementos  $x \in U$  que possuem graus de pertinência em  $A$  maiores ou iguais do que o valor especificado em  $\alpha$ , ou seja,  $A_\alpha = \{x | \mu_A(x) \geq \alpha\}$ .
- O conjunto de nível- $\beta$ , definido como  $A_\beta$ , é o conjunto de todos os graus de pertinência  $\mu_A(x)$  de  $A$  que possuem valor maior ou igual ao valor especificado  $\beta$ , isto é,  $A_\beta = \{\{\mu_A(x); x\} | \mu_A(x) \geq \beta\}$  ou

$$\mu_\beta(x) = \begin{cases} \mu_A(x) & \text{se } \mu_A(x) \geq \beta \\ 0 & \text{c.c.} \end{cases}$$

- Um conjunto nebuloso *singleton*,  $A_S$ , é um conjunto cuja função de pertinência é nula para todo  $x$  pertencente a  $U$ , exceto em  $x_i$  onde  $\mu_{A_S}(x_i) = 1$ .
- Um conjunto nebuloso é dito convexo se somente se:  $\mu_A(kx_1 + (1-k)x_2) \geq \min(\mu_A(x_1), \mu_A(x_2))$ ,  $\forall x_1, x_2 \in U$  e  $k \in [0, 1]$ .
- A cardinalidade de um conjunto nebuloso é uma medida do tamanho do conjunto nebuloso dada por:  $\text{Card}(A) = \sum_U \mu_A(x)$ .
- O complemento de um conjunto nebuloso  $A$  denotado por  $\bar{A}$  é tal que:  $\mu_{\bar{A}}(x) = 1 - \mu_A(x), \forall x \in U$ .
- O conjunto nebuloso resultante da união de dois conjuntos nebulosos,  $A$  e  $B$ , é aquele cuja função de pertinência é composta pelos máximos das funções de pertinência de  $A$  e de  $B$ ,  $\forall x \in U$ . Ou seja,  $\mu_A(x) \vee \mu_B(x) = \max(\mu_A(x), \mu_B(x))$ .

- O conjunto nebuloso resultante da intersecção de dois conjuntos nebulosos,  $A$  e  $B$ , é aquele cuja função de pertinência é composta pelos mínimos das funções de pertinência de  $A$  e de  $B$ ,  $\forall x \in U$ . Ou seja,  $\mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x), \mu_B(x))$ .

### 3.2.4 Operações Algébricas

Dado que  $A$  e  $B$  são conjuntos nebulosos definidos no universo de discurso  $U$ , com funções de pertinência  $\mu_A(x)$  e  $\mu_B(x)$  respectivamente, as principais operações algébricas envolvendo os conjuntos nebulosos são [32]:

**Soma Algébrica:**  $A+B = \{\{\mu_{A+B}(x); x\} | x \in U\}$  em que  $\mu_{A+B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$

**Produto por um escalar  $\alpha$ :**  $\alpha A = \{\{\alpha \mu_A(x); x\} | x \in U\}$  com  $\alpha > 0$  tal que  $\text{Sup}(\alpha \mu_A(x)) \leq 1$ .

**Produto Algébrico:**  $A.B = \{\{\mu_{A.B}(x); x\} | x \in U\}$  em que  $\mu_{A.B}(x) = \mu_A(x) \cdot \mu_B(x)$ .

**Potenciação:**  $A^\alpha = \{\{\mu_A(x)^\alpha; x\} | x \in U\}$ .

## 3.3 Relações Nebulosas

Genericamente, uma relação qualquer exprime a presença ou a ausência de uma associação (ou interação) entre elementos de dois ou mais conjuntos. As relações nebulosas generalizam o conceito dessas relações e representam o grau da associação entre elementos de dois ou mais conjuntos *fuzzy*.

### 3.3.1 Definição

Uma relação nebulosa binária  $R$ , definida no espaço  $X \times Y = \{(x, y) | x \in X, y \in Y\}$  é um subconjunto nebuloso no espaço  $X \times Y$  caracterizado pela função de pertinência  $\mu_R(x, y)$ , a qual associa cada par  $(x, y)$  pertencente a  $R$  a um valor real no intervalo  $[0, 1]$ , ou seja:

$$R : X \times Y \longrightarrow [0, 1]$$

$$R = \int_{X \times Y} \{\mu_R(x, y); (x, y)\} \quad (3.3)$$

no qual o símbolo " $\int_{X \times Y}$ " representa a operação de união lógica em todo o espaço  $X \times Y$ . O grau de pertinência  $\mu_R(x, y)$  expressa a força da relação entre os elementos do par  $(x, y)$ .

### 3.3.2 Norma-T e Conorma-T (ou S-Norma)

O conceito de Norma-T e Conorma-T foi introduzido na teoria dos conjuntos nebulosos por Dubois e Prade em 1982 [35]. Uma Norma-T triangular é uma função de duas variáveis reais  $T : [0, 1] \times [0, 1] \rightarrow [0, 1]$ , e que satisfaz as seguintes condições [34]:

**Condição de Contorno:**  $t(0, 0) = 0$  e  $t(a, 1) = a$

**Monotônica:**  $t(a, b) \leq t(c, d)$ ;  $a \leq c$  e  $b \leq d$

**Simétrica:**  $t(a, b) = t(b, a)$

**Associativa:**  $t(a, t(b, c)) = t(b, t(a, c))$

O mapeamento Conorma-T, ou S-norma,  $S : [0, 1] \times [0, 1] \rightarrow [0, 1]$  satisfaz as condições[36]:

**Condição de Contorno:**  $s(1, 1) = 1$  e  $s(a, 0) = a$

**Monotônica:**  $s(a, b) \leq s(c, d)$ ;  $a \leq c$  e  $b \leq d$

**Simétrica:**  $s(a, b) = s(b, a)$

**Associativa:**  $s(a, s(b, c)) = s(b, s(a, c))$

Alguns exemplos de operadores da Norma-T e Conorma-T são descritos abaixo:

Norma-T	Conorma-T
Operador Mínimo $\rightarrow \min(a, b)$	Operador Máximo $\rightarrow \max(a, b)$
Operador Produto $\rightarrow a.b$	Operador Soma Prob. $\rightarrow a + b - a.b$
Operador Tm $\rightarrow \max(0, a + b - 1)$	Operador Soma Limit. $\rightarrow \min(1, a + b)$

Inúmeras Normas-T e Conormas-T têm sido propostas na literatura, porém as mais utilizadas em aplicações, como na engenharia, são os operadores *min* e *produto* para a interseção nebulosa, e o operador *max* para a união nebulosa.

## 3.4 Lógica Nebulosa (*Fuzzy Logic*)

A lógica nebulosa tem por objetivo modelar aproximadamente o modo de raciocínio humano, visando desenvolver sistemas computacionais capazes de tomar decisões racionais em um ambiente de incerteza e imprecisão. Anterior à lógica nebulosa, surgiu a lógica multi-valor que buscava ser

uma generalização à lógica clássica de dois valores  $[0, 1]$ . Dentre as lógicas multi-valores, tornaram-se mais conhecidas as lógicas tri-valores, onde o terceiro estado é representado por  $1/2$ . O foco primário da lógica nebulosa está na linguagem natural humana, onde há predominância de raciocínio aproximado e preposições vagas.

### 3.4.1 Variável Nebulosa e Variável Linguística

Uma variável nebulosa é uma variável cujo valor é um rótulo (*label*), o qual está associado a um conjunto nebuloso. *Jovem, alto, pequeno, muito grande*, etc, são exemplos de variáveis nebulosas.

Uma variável linguística é uma variável cujos valores são sentenças na forma de linguagem "natural" [32]. Uma variável linguística tem como valores variáveis nebulosas. *Idade, Altura, Temperatura, Distância, Velocidade*, etc, são exemplos de variáveis linguísticas. E os termos *pequeno, grande, muito jovem, muito próximo*, etc, são os valores para as variáveis linguísticas.

### 3.4.2 Regra de Inferência

As relações entre as variáveis linguísticas são descritas através de declarações condicionais nebulosas do tipo "*SE X É A, ENTÃO Y É B*", onde  $X$  e  $Y$  são variáveis linguísticas, e  $A$  e  $B$  são conjuntos nebulosos. O termo " $X \text{ É } A$ " é a parte antecedente ou premissa da regra de inferência e o termo " $Y \text{ É } B$ " é dito parte consequente. Os conjuntos  $A$  e  $B$  pertencem a universos de discurso distintos, ou seja, falam de coisas relacionadas mas diferentes. Como exemplo, podemos citar: *SE o verão é muito quente, ENTÃO o consumo de picolé é alto; SE a demanda aumenta, ENTÃO os preços sobem*, etc. A regra de inferência estabelece uma sistemática para se determinar o valor da variável  $Y$ , se a variável  $X$  for diferente de  $A$ .

### 3.4.3 Sistemas Nebulosos

Um sistema nebuloso é formado pela agregação de um conjunto de regras de inferência no formato "*SE  $X_1 \text{ é } A_1 \text{ e } X_2 \text{ é } A_2 \text{ e } \dots \text{ e } X_n \text{ é } A_n$ , ENTÃO  $Y \text{ é } B$* ". Os sistemas nebulosos mais populares podem ser classificados em três tipos: sistema lógico nebuloso puro, sistema nebuloso Takagi-Sugeno e sistema nebuloso com fuzzificador e defuzzificador [37].

#### Sistema Lógico Nebuloso Puro

Um sistema nebuloso puro mapeia regiões no espaço de entradas  $U_1 \times U_2 \times \dots \times U_r$  para regiões no espaço das saídas  $V_1 \times V_2 \times \dots \times V_m$  [37]. Este mapeamento se dá através da relação de conjuntos nebulosos para outros conjuntos nebulosos. A Figura 3.2 mostra um sistema nebuloso puro com duas

entradas e uma saída. Esse sistema mapeia regiões do espaço  $\mathbb{R}^2$  para o espaço  $\mathbb{R}^1$  e é formado por  $C$  regras de inferência que codificam o conhecimento de um especialista descrito como:

$$R^i : \text{SE } X_1 \text{ é } A_1^i \text{ e } X_2 \text{ é } A_2^i \text{ ENTÃO } Y \text{ é } B^i, \text{ para } 1 \leq i \leq C. \quad (3.4)$$

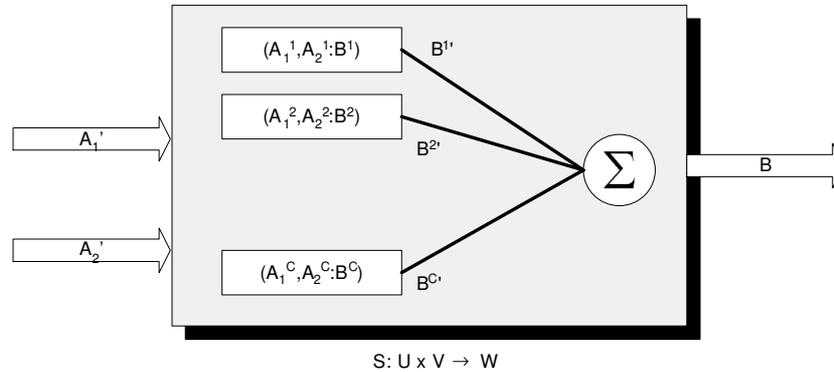


Fig. 3.2: Sistema Nebuloso Puro com Duas Entradas e Uma Saída

Do ponto de vista da lógica nebulosa, a variável linguística  $X_1$  e os conjuntos nebulosos  $A_1^i$  pertencem ao universo de discurso  $U$ , a variável linguística  $X_2$  e os conjuntos nebulosos  $A_2^i$  pertencem ao universo de discurso  $V$ , enquanto que a variável linguística  $Y$  e os conjuntos nebulosos  $B^i$  pertencem ao universo de discurso  $W$ , onde  $1 \leq i \leq C$ .

As duas entradas  $A_1^i$  e  $A_2^i$  interagem com as  $C$  regras, aqui representadas na forma compacta por  $(A_1^i, A_2^i : B^i)$ , produzindo  $C$  conjuntos nebulosos  $B^{i'}$  que são uma versão dos  $C$  conjuntos  $B^i$  modificados pelo grau de interação das entradas com as respectivas regras de inferência. Esses conjuntos nebulosos são então somados logicamente para produzir o conjunto nebuloso de saída  $B$ .

A principal desvantagem do sistema nebuloso puro é que suas entradas e saídas são conjuntos nebulosos, e nos principais sistemas de engenharia, as entradas e saídas são variáveis de valores reais. Para superar essa desvantagem, Takagi e Sugeno propuseram um sistema nebuloso cujas entradas e saídas são valores reais [8].

### Sistema Nebuloso Takagi-Sugeno - TSK

O modelo nebuloso TSK tem atraído uma grande atenção devido seu bom desempenho em várias aplicações [37]. O modelo proposto consiste de regras SE-ENTÃO que têm a seguinte forma:

$$R^i : \text{SE } x_1 \text{ é } A_1^i \text{ e } x_2 \text{ é } A_2^i \text{ e } \dots \text{ e } x_n \text{ é } A_n^i \quad (3.5)$$

$$\text{ENTÃO } y^i = a_0^i + a_1^i x_1 + a_2^i x_2 + \dots + a_n^i x_n,$$

em que  $\vec{x} = (x_1, x_2, \dots, x_n)$  é o vetor de variáveis de entrada no sistema,  $n$  é o número de elementos do vetor de entrada,  $A_j^i$  são conjuntos nebulosos, com  $i = 1, 2, \dots, C$  e  $j = 1, 2, \dots, n$ ,  $\vec{a}^i = [a_0^i, a_1^i, a_2^i, \dots, a_n^i]$  é o vetor de parâmetros consequentes (valores reais) e  $y^i$  é a saída do sistema para a regra  $i$  dada pela combinação linear das variáveis de entrada. Assim, temos que a parte antecedente "SE" é nebulosa mas a parte consequente "ENTÃO" é dita "crisp" (exata).

Para um vetor de entrada real  $\vec{x} = (x_1, x_2, \dots, x_n)$ , a saída  $y$  do sistema Takagi-Sugeno será a média ponderada das saídas de cada regra *Se-Então*  $y^i$  dada pela equação abaixo:

$$y = \frac{\sum_{i=1}^C w^i y^i}{\sum_{i=1}^C w^i} \quad (3.6)$$

onde  $w^i(k) = \min_{j=1 \dots n} A_j^i(x_j)$

O sistema *fuzzy* TSK é mostrado na Figura 3.3. A vantagem desse sistema é que ele provê uma equação de sistema compacta dada pela equação (3.6).

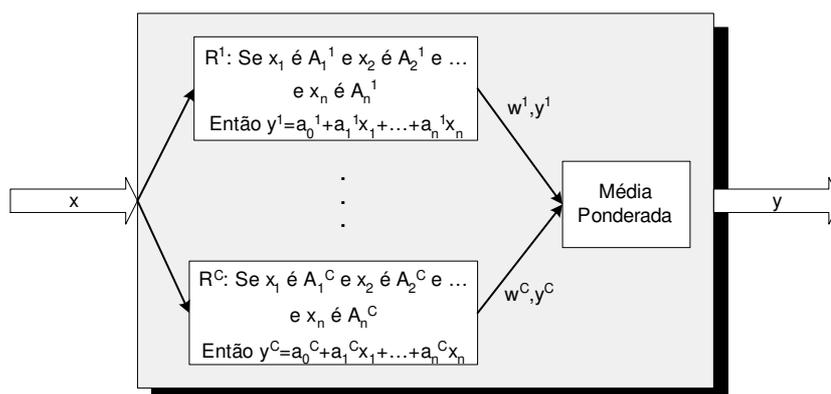


Fig. 3.3: Sistema Nebuloso Takagi-Sugeno.

### Sistema Lógico *Fuzzy* com Fuzzificador e Defuzzificador

Na grande maioria das aplicações dos sistemas nebulosos em engenharia, a troca de informações entre o ambiente e o sistema se dá por meio de variáveis cujo valores são números reais. Como foi citado anteriormente, o sistema nebuloso TSK é capaz de trabalhar diretamente com esses valores sem a necessidade de uma conversão para conjuntos nebulosos. Uma solução alternativa seria o uso do sistema nebuloso puro da Figura 3.2 modificado pelo acréscimo de dois blocos funcionais chamados de *Fuzzificador* e *Defuzzificador*, como pode ser visto na Figura 3.4.

O bloco *Fuzzificador* tem a função de converter um valor real em um conjunto nebuloso, e o bloco *Defuzzificador* realiza o processo inverso.

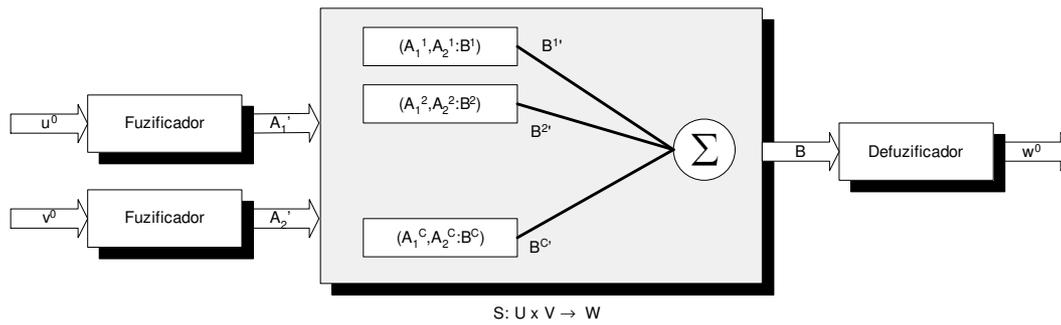


Fig. 3.4: Sistema Nebuloso com Fuzificador e Defuzificador

Na literatura, esse sistema nebuloso é comumente chamado de controlador nebuloso devido à grande aplicação em controladores. Foi primeiramente proposto por Mamdani [38] e tem sido aplicado a uma variedade de processos industriais devido suas vantagens[39]. Dentre elas, podemos citar a liberdade na escolha do fuzzificador e do defuzzificador de tal forma a se obter o sistema nebuloso mais preciso para um problema particular. Os métodos de fuzzificação e defuzzificação mais usados são: centro da gravidade, centro da soma, método dos máximos e método das alturas.

### 3.5 Considerações Finais

Neste capítulo, foram apresentados alguns aspectos relevantes sobre a teoria da lógica nebulosa, objetivando-se fornecer conceitos básicos dessa teoria. Como se pode concluir, a teoria dos conjuntos nebulosos pode ser vista como uma ponte, que une modelos matemáticos tradicionais de sistemas físicos reais (precisos) à representação mental (geralmente imprecisa) desses sistemas.

A teoria nebulosa é de fundamental importância no presente trabalho pois será utilizada para modelar o comportamento dos fluxos de tráfego. O modelo *fuzzy* TSK será utilizado de forma preditiva para prever o comportamento do buffer e controlar a taxa de entrada.

No capítulo seguinte, será detalhado o esquema proposto de controle de congestionamento adaptativo baseado na modelagem *fuzzy* TSK.

## Capítulo 4

# Esquema de Controle de Congestionamento Adaptativo Baseado em Modelagem *Fuzzy*

### 4.1 Introdução

Nesse capítulo, o esquema de controle de congestionamento proposto será detalhado. A modelagem *fuzzy* TSK é utilizada na forma preditiva com o intuito de prever o comportamento dos fluxos de tráfego, mais especificamente, do tamanho da fila no *buffer*. Tal modelagem possui vantagens sobre os modelos lineares em descrever o comportamento de processos reais desconhecidos com características não-lineares, como é o caso dos fluxos de tráfego de redes [7][8][9].

O modelo TSK é treinado com o algoritmo de agrupamento proposto ARFARC (Agrupamento Regressivo *Fuzzy* Adaptativo com Reset da Covariância), o qual é dividido em duas etapas. Na primeira etapa, os parâmetros das partes antecedentes e consequentes das regras nebulosas são identificados e o modelo TSK preditivo inicial é obtido. Na segunda etapa, um algoritmo de refinamento é utilizado para melhorar a precisão dos resultados obtidos na primeira etapa.

Com os parâmetros da predição, propomos uma taxa ótima para regular a taxa da fonte de tal forma que o tamanho da fila no *buffer* seja igual ou inferior à um nível desejado e a variância do tamanho da fila se torne mínima possível.

### 4.2 Descrição do Sistema de Controle de Congestionamento Proposto

A estrutura básica do sistema de controle de congestionamento proposto é mostrada na Figura 4.1, na qual estão identificados todos os elementos do sistema: os fluxos de tráfego  $\mu(k)$  e  $\nu(k)$ , o



Devido ao surgimento de diversos tipos de serviços e aplicações, tais como, dados, voz e vídeo, e ao aumento do chaveamento e multiplexação dos mesmos, o tráfego de pacotes em redes de alta velocidade pode ser considerado um processo não-linear [40]. Um exemplo de sistema altamente não-linear é a dinâmica do tamanho da fila no *buffer* do sistema da Figura 4.1, descrita pela equação abaixo:

$$b(k+1) = \min\{\max[b(k) + (\mu(k-d) + \nu(k) - \eta) \times T, 0], B_{max}\} \quad (4.1)$$

em que  $B_{max}$  é o tamanho do *buffer*.

Muitos estudos mostram que o modelo *fuzzy* TSK possui vantagens sobre os modelos lineares em descrever o comportamento de processos reais com características não-lineares e variantes no tempo, como é o caso de sistemas como esse descrito pela equação (4.1) [7][8][9]. Desse modo, propomos um preditor adaptativo do tamanho da fila no *buffer* baseado na modelagem *fuzzy* TSK, o qual leva vantagem ao explorar as propriedades de não-linearidade e incerteza da dinâmica do tamanho da fila no *buffer* quando comparado a outros preditores adaptativos.

### 4.3 Preditor *Fuzzy* TSK Adaptativo

Nesta seção, um preditor *fuzzy* adaptativo para o tamanho da fila no *buffer*  $d$  passos a frente  $b(k+d)$  é proposto, baseado nas informações passadas e atuais da taxa de fonte controlável  $\mu(s)_{s \leq k}$  e do tamanho da fila no *buffer*  $b(s)_{s \leq k}$ .

O algoritmo de agrupamento proposto ARFARC (Agrupamento Regressivo *Fuzzy* Adaptativo com Reset da Covariância), utilizado para treinar o preditor *fuzzy* TSK, identifica simultaneamente os subespaços *fuzzy* e os parâmetros das partes consequentes das regras nebulosas. Na primeira parte do algoritmo, obtemos uma inicialização aproximada do modelo TSK. E na segunda parte, um algoritmo de refinamento baseado no gradiente descendente [41] é utilizado para fazer um ajuste fino do modelo inicial obtido na primeira parte, melhorando a sua precisão.

#### 4.3.1 Modelo *Fuzzy* TSK Aplicado à Predição do Tamanho da Fila no *Buffer*

Algumas técnicas para modelagem de regras nebulosas têm sido propostas na literatura e muitas delas definem seus subespaços *fuzzy* baseado no treinamento com os dados de entrada [7] [9] [8]. O modelo *fuzzy* TSK é um exemplo de tais técnicas, o qual interpola localmente relações entrada-saída lineares através de funções de pertinência resultando em sistemas não-lineares. Esse modelo é descrito por regras *SE-ENTÃO* conforme apresentado na seção 3.4.3, e é usado neste trabalho para tratar a dinâmica do tamanho da fila no *buffer*.

Seja um conjunto de observações  $\{(\vec{x}(1), y(1)), (\vec{x}(2), y(2)), \dots, (\vec{x}(N), y(N))\}$  com  $\vec{x}(k) \in \mathbb{R}^n$  e  $y(k) \in \mathbb{R}$ , onde  $N$  é o número de dados de treinamento,  $\vec{x}(k) = [x_1(k), x_2(k), \dots, x_n(k)]$  é o  $k$ -ésimo vetor de entrada e  $y(k)$  é a saída desejada para a entrada  $\vec{x}(k)$ , com  $1 \leq k \leq N$ . Supondo que essas observações são relacionadas por uma função desconhecida  $y(k) = f(x_1(k), x_2(k), \dots, x_n(k))$ , deseja-se construir uma função  $\hat{f}$  que possa representar essa relação entrada-saída  $f$  através de um modelo fuzzy TSK. No nosso caso, como já foi citado, os vetores de entrada consistem dos valores passados e atuais do tamanho da fila no *buffer* e do tráfego controlável e a saída consiste do tamanho da fila no *buffer*  $d$  passos a frente, isto é,

$$\vec{x}(k) = [b(k-p), b(k-p+1), \dots, b(k), \mu(k-q), \mu(k-q+1), \dots, \mu(k)] \quad \text{e} \quad y(k) = b(k+d). \quad (4.2)$$

Nosso modelo TSK aplicado à predição do tamanho da fila no *buffer* consiste de regras *SE-ENTÃO* que têm a seguinte forma:

$$\begin{aligned} R^i : & \text{ Se } b(k-p) \text{ é } A_1^i(\vec{\theta}_1^i) \text{ e } b(k-p-1) \text{ é } A_2^i(\vec{\theta}_2^i) \text{ e } \dots \text{ e } b(k) \text{ é } A_{p+1}^i(\vec{\theta}_{p+1}^i) \\ & \text{ e } \mu(k-q) \text{ é } A_{p+2}^i(\vec{\theta}_{p+2}^i) \text{ e } \mu(k-q-1) \text{ é } A_{p+3}^i(\vec{\theta}_{p+3}^i) \text{ e } \dots \text{ e } \mu(k) \text{ é } A_{p+q+2}^i(\vec{\theta}_{p+q+2}^i) \\ & \text{ Então } y^i(k) = b^i(k+d) = \alpha_0^i + \alpha_1^i b(k-p) + \alpha_2^i b(k-p+1) + \dots + \alpha_{p+1}^i b(k) + \\ & + \beta_{p+2}^i \mu(k-q) + \beta_{p+3}^i \mu(k-q+1) + \dots + \beta_{p+q+2}^i \mu(k) \end{aligned} \quad (4.3)$$

para  $i = 1, 2, \dots, C$ , em que  $C$  é o número total de regras,  $p+1$  é o número de amostras de entrada referente aos valores atuais e passados do tamanho da fila no *buffer*,  $q+1$  é o número de amostras de entradas referente aos valores atuais e passados do tráfego controlável,  $n = p+q+2$  é o número de entradas total,  $A_j^i(\vec{\theta}_j^i)$  é o conjunto nebuloso da  $i$ -ésima regra para o  $j$ -ésimo elemento do  $k$ -ésimo vetor de entrada  $\vec{x}(k)$ , sendo  $\vec{\theta}_j^i$  o conjunto de parâmetros ajustáveis das partes antecedentes com  $1 \leq j \leq n$ , e  $\vec{a}^i(k) = (\alpha_0^i, \alpha_1^i, \dots, \alpha_{p+1}^i, \beta_{p+2}^i, \beta_{p+3}^i, \dots, \beta_{p+q+2}^i)$  é o conjunto de parâmetros das partes consequentes para o instante  $k$ .

A saída do modelo nebuloso TSK é dada por:

$$\hat{y}(k) = \hat{b}(k+d) = \frac{\sum_{i=1}^C b^i(k+d) w^i(k)}{\sum_{i=1}^C w^i(k)} \quad (4.4)$$

em que  $b^i(k+d)$  é a saída da  $i$ -ésima regra,  $w^i(k) = \min_{j=1, \dots, n} A_j^i(\vec{\theta}_j^i; x_j(k))$  é o grau de ativação da  $i$ -ésima regra, o qual é obtido pelo mínimo das funções de pertinência de todas as variáveis nebulosas.

A especificação completa do modelo fuzzy TSK descrito na equação (4.3) consiste na identificação dos parâmetros das partes antecedentes (isto é,  $\vec{\theta}_j^i$ ) e das parte consequentes (isto é,  $\vec{a}^i$ ). Para isto, utilizaremos o algoritmo de agrupamento proposto ARFARC (Agrupamento Regressivo Fuzzy

Adaptativo com Reset da Covariância) que leva em consideração o erro de regressão e a distribuição espacial dos dados, analisando a distância de cada vetor de entrada ao centro de cada grupo<sup>1</sup>. Na primeira parte do algoritmo, obteremos uma inicialização aproximada do modelo TSK. Na segunda parte, um algoritmo de refinamento baseado no gradiente descendente será utilizado para ajustar o modelo inicial obtido melhorando a sua precisão.

### 4.3.2 Algoritmo ARFARC - Parte 1: Modelo TSK Inicial

Nesta seção, desenvolvemos a primeira parte do algoritmo de agrupamento proposto ARFARC que identifica simultaneamente os parâmetros das partes antecedentes ( $\vec{\theta}_j^i$ ) e das partes consequentes ( $\vec{a}^i$ ).

Diferente do algoritmo de agrupamento FCRM (*Fuzzy C-Regression Model*), definido em [42], que não considera a distribuição dos dados de treinamento, o algoritmo de agrupamento proposto ARFARC leva em conta tal distribuição analisando a distância entre os dados de entrada e os centros dos grupos. Além disso, o algoritmo proposto ARFARC considera também o erro de regressão, ou seja, a diferença entre a saída desejada e a saída obtida pelo modelo TSK.

Seja a função custo  $J$  do algoritmo ARFARC, definida como:

$$J = \sum_{i=1}^C \sum_{k=1}^N (u_{ik} r_{ik} d_{ik})^2 \quad (4.5)$$

com

$$\sum_{i=1}^C u_{ik} = 1, \quad \text{for } 1 \leq k \leq N \quad (4.6)$$

em que  $u_{ik}$  é o grau de ativação da  $i$ -ésima regra para o  $k$ -ésimo padrão de treinamento (vetor de entrada  $\vec{x}(k)$ ),  $C$  é o número de regras nebulosas e  $N$  é o número de dados de treinamento. Em (4.5),  $r_{ik}$  é o erro entre a  $k$ -ésima saída desejada do sistema modelado e a saída da  $i$ -ésima regra com a  $k$ -ésima entrada, isto é,

$$r_{ik} = y(k) - f_i(\vec{x}(k); \vec{a}^i(k)), \quad (4.7)$$

com  $i = 1, 2, \dots, C$  e  $k = 1, 2, \dots, N$ . Na mesma equação (4.5),  $d_{ik}$  é a distância entre a  $k$ -ésima entrada e o centro do  $i$ -ésimo grupo definido como  $\beta_i$ , isto é,

$$d_{ik} = \vec{x}(k) - \beta_i \quad (4.8)$$

---

<sup>1</sup>Os grupos (*clusters*) são formados pelos dados de entrada do sistema de acordo com o critério adotado pelo algoritmo de agrupamento. Cada grupo é representado pelo seu centro e o grau de pertinência de uma determinada entrada ao grupo vai diminuindo com a distância da mesma ao centro do grupo.

Para minimizar a função  $J$  em (4.5), aplicamos o método de multiplicadores de Lagrange [43]. Deste modo, temos a função Lagrange definida como:

$$L = \sum_{i=1}^C \sum_{k=1}^N u_{ik}^2 (r_{ik} d_{ik})^2 - \sum_{k=1}^N \lambda_k \left( \sum_{i=1}^C u_{ik} - 1 \right) \quad (4.9)$$

onde  $\lambda_k$  são multiplicadores de Lagrange e  $k = 1, 2, \dots, N$ .

A minimização da função custo  $J$  implica na imposição das seguintes condições:

$$\frac{\partial L}{\partial \vec{a}^i(k)} = \sum_{k=1}^N (u_{ik})^2 (d_{ik})^2 \frac{\partial r_{ik}^2}{\partial \vec{a}^i(k)} = 0 \quad (4.10)$$

$$\frac{\partial L}{\partial u_{ik}} = 2u_{ik}(r_{ik}d_{ik})^2 - \lambda_k = 0 \quad (4.11)$$

$$\frac{\partial L}{\partial \beta_i} = \sum_{k=1}^N (u_{ik})^2 (r_{ik})^2 \frac{\partial d_{ik}^2}{\partial \beta_i} = 0 \quad (4.12)$$

Note que a derivada parcial  $\frac{\partial r_{ik}^2}{\partial \vec{a}^i(k)}$  em (4.10) pode ser obtida através da equação (4.7) da seguinte forma:

$$\frac{\partial r_{ik}^2}{\partial \vec{a}^i(k)} = 2r_{ik} \frac{\partial r_{ik}}{\partial \vec{a}^i(k)} = 2[y(k) - f_i(\vec{x}(k); \vec{a}^i(k))] \frac{\partial r_{ik}}{\partial \vec{a}^i(k)}. \quad (4.13)$$

Substituindo (4.13) na equação (4.10) temos:

$$\sum_{k=1}^N (u_{ik})^2 (d_{ik})^2 \frac{\partial r_{ik}}{\partial \vec{a}^i(k)} y(k) - \sum_{k=1}^N (u_{ik})^2 (d_{ik})^2 \frac{\partial r_{ik}}{\partial \vec{a}^i(k)} f_i(\vec{x}(k); \vec{a}^i(k)) = 0, \quad (4.14)$$

em que  $\frac{\partial r_{ik}}{\partial \vec{a}^i(k)}$  é obtido derivando a equação (4.7) e, assim, obtemos:

$$\frac{\partial r_{ik}}{\partial \vec{a}^i(k)} = \vec{x}(k). \quad (4.15)$$

Para o  $k$ -ésimo padrão de treinamento, com  $k = 1, 2, \dots, N$ , definimos  $X(k) \in \mathbb{R}^{k \times (n+1)}$  como uma matriz cujas colunas são os valores de  $x_j$  na  $(j+1)$ -ésima coluna. A primeira coluna de  $X(k)$  é toda composta por 1. São também definidos  $Y \in \mathbb{R}^k$  como um vetor onde o  $k$ -ésimo elemento é o valor de  $y(k)$  e  $Q_i(k) \in \mathbb{R}^{k \times k}$  uma matriz diagonal onde o  $k$ -ésimo elemento da diagonal é dado por  $\omega(k) = u_{ik}^2 d_{ik}^2$ . Assim, a equação (4.14) escrita na forma matricial é dada por:

$$X^T(k) Q_i(k) Y(k) - (X^T(k) Q_i(k) X(k)) \vec{a}^i(k) = 0. \quad (4.16)$$

Denotaremos por  $P_i(k)$ , a matriz de covariância de  $X(k)$  e  $Q_i(k)$ , dada por:

$$P_i(k) = (X^T(k)Q_i(k)X(k))^{-1}, \quad i = 1, 2, \dots, C \quad (4.17)$$

Assim, o vetor de parâmetros  $\vec{a}^i$  das partes consequentes da  $i$ -ésima regra para o  $k$ -ésimo padrão de treinamento é dado por:

$$\vec{a}^i(k) = P_i(k)X^T(k)Q_i(k)Y(k). \quad (4.18)$$

Para evitar a inversão matricial do termo  $(X^T(k)Q_i(k)X(k))$  em (4.17) no cálculo da matriz de covariância, propomos um algoritmo recursivo de estimação do vetor de parâmetros das partes consequentes  $\vec{a}^i$ . Para obtermos as equações utilizadas no algoritmo recursivo proposto, um certo desenvolvimento matemático é detalhado no Apêndice A e os resultados obtidos são dados por:

$$\vec{a}^i(k+1) = \vec{a}^i(k) + P_i(k+1)x^T(k+1)\omega(k+1)(y(k+1) - x(k+1)\vec{a}^i(k)), \quad (4.19)$$

em que  $P_i(k+1)$  é dado por:

$$P_i(k+1) = P_i(k) - \left( \frac{P_i(k)x^T(k+1)\omega(k+1)x(k+1)P_i(k)}{1 + \omega(k+1)x(k+1)P_i(k)x^T(k+1)} \right), \quad (4.20)$$

em que  $x(k+1)$  é a  $(k+1)$ -ésima linha da matriz  $X(k)$  e  $w(k+1)$  é o  $(k+1)$ -ésimo elemento diagonal da matriz  $Q_i(k+1)$ .

Esse método recursivo de estimação dos parâmetros das partes consequentes apresenta um alta taxa de convergência inicial nas simulações, contudo o ganho do algoritmo é reduzido pois a matriz de covariância  $P_i$  diminui rapidamente dentro de poucas iterações. Para evitar esse comportamento, a matriz  $P_i$  é inicializada a cada 10 iterações de modo que o ganho não é reduzido e a alta convergência também é obtida.

O grau de ativação da  $i$ -ésima regra para o  $k$ -ésimo padrão de treinamento,  $u_{ik}$ , pode ser obtido através da segunda condição necessária para minimizar a função custo  $J$ , definida na equação (4.11), ou seja:

$$u_{ik} = \frac{\lambda_k}{2r_{ik}^2 d_{ik}^2} \quad (4.21)$$

O parâmetro  $\lambda_k$  pode ser obtido substituindo a equação (4.21) na equação (4.6) obtendo:

$$\sum_{i=1}^C \lambda_k \frac{1}{2r_{ik}^2 d_{ik}^2} = 1 \Rightarrow \lambda_k \sum_{i=1}^C \frac{1}{2r_{ik}^2 d_{ik}^2} = 1 \Rightarrow \lambda_k = \frac{1}{\sum_{i=1}^C \frac{1}{2r_{ik}^2 d_{ik}^2}} \quad (4.22)$$

Substituindo o parâmetro  $\lambda_k$  da equação (4.22) na equação (4.21) obtemos:

$$u_{ik} = \frac{\frac{1}{\sum_{i=1}^C \frac{1}{2r_{ik}^2 d_{ik}^2}}}{2r_{ik}^2 d_{ik}^2} \Rightarrow u_{ik} = \frac{1}{\sum_{i=1}^C \frac{1}{2r_{ik}^2 d_{ik}^2}} \frac{1}{2r_{ik}^2 d_{ik}^2}$$

$$u_{ik} = \frac{1/(2r_{ik}^2 d_{ik}^2)}{\sum_{i=1}^C \frac{1}{2r_{ik}^2 d_{ik}^2}} \quad (4.23)$$

Para simplificar a terceira condição, descrita pela equação (4.12), temos que a derivada parcial  $\frac{\partial d_{ik}}{\partial \beta_i}$  nesta equação pode ser calculada através da distância entre a  $k$ -ésima entrada e o centro do  $i$ -ésimo grupo, dada pela equação (4.8), da seguinte forma:

$$\frac{\partial d_{ik}^2}{\partial \beta_i} = 2d_{ik} \frac{\partial d_{ik}}{\partial \beta_i} = 2(\vec{x}(k) - \beta_i) \cdot (-1) = 2\beta_i - 2\vec{x}(k) \quad (4.24)$$

Substituindo (4.24) em (4.12) temos:

$$\sum_{k=1}^N u_{ik}^2 r_{ik}^2 (2\beta_i - 2\vec{x}(k)) = 0 \Rightarrow \sum_{k=1}^N u_{ik}^2 r_{ik}^2 \beta_i = \sum_{k=1}^N u_{ik}^2 r_{ik}^2 \vec{x}(k)$$

$$\beta_i = \frac{\sum_{k=1}^N u_{ik}^2 r_{ik}^2 \vec{x}(k)}{\sum_{k=1}^N u_{ik}^2 r_{ik}^2} \quad (4.25)$$

Desse modo, resumindo a primeira parte do algoritmo ARFARC proposto, para cada padrão de treinamento  $(\vec{x}(k), y(k))$ , a saída do modelo TSK será calculada pela equação (4.4) e os parâmetros  $r_{ik}$ ,  $d_{ik}$ ,  $u_{ik}$ ,  $\beta_i$  e  $\vec{a}^i$  serão atualizados, respectivamente, pelas equações (4.7), (4.8), (4.23), (4.25) e (4.19), caracterizando assim uma modelagem adaptativa. Vale também ressaltar que, a atualização do vetor de parâmetros consequentes,  $\vec{a}^i(k)$ , é recursiva, ou seja, ela usa a amostra anterior para calcular a amostra atual.

Nessa primeira parte do algoritmo de agrupamento proposto ARFARC, os parâmetros das partes antecedentes também são obtidos. Supomos que funções gaussianas são utilizadas para as funções de pertinência das partes premissas (antecedentes), assim, temos que:

$$A_j^i(\theta_{j1}^i; \theta_{j2}^i) = \exp \left\{ - \left( \frac{x_j(k) - \theta_{j1}^i}{\theta_{j2}^i} \right)^2 \right\} \quad (4.26)$$

Para o cálculo das funções de pertinência, devemos, portanto obter os parâmetros  $\theta_{j1}^i$  e  $\theta_{j2}^i$ . Tais parâmetros correspondem, respectivamente, à média e ao desvio padrão da  $j$ -ésima função de pertinência da  $i$ -ésima regra nebulosa, onde  $1 \leq j \leq n$  e  $1 \leq i \leq C$ . Assim, temos as seguintes equações

para o cálculo dos parâmetros  $\theta_{j1}^i$  e  $\theta_{j2}^i$ :

$$\theta_{j1}^i(k) = \frac{\sum_{z=1}^N (u_{iz})^2 x_j(z)}{\sum_{z=1}^N (u_{iz})^2} \quad (4.27)$$

$$\theta_{j2}^i(k) = \sqrt{\frac{\sum_{z=1}^N (u_{iz})^2 (x_j(z) - \theta_{j1}^i)^2}{\sum_{z=1}^N (u_{iz})^2}} \quad (4.28)$$

### 4.3.3 Algoritmo ARFARC - Parte 2: Algoritmo de Refinamento

Na segunda parte do algoritmo de agrupamento ARFARC, os parâmetros das partes antecedentes ( $\bar{a}^i(k)$ ) e consequentes ( $\theta_{j1}^i$  e  $\theta_{j2}^i$ ) são ajustados por um algoritmo de refinamento baseado em um procedimento de aprendizagem supervisionado para melhorar a precisão da modelagem. O algoritmo de aprendizagem supervisionado considerado é o gradiente descendente e está definido em [41].

Seja a função custo  $E_R$  definida como:

$$E_R = \frac{1}{N} \sum_{j=1}^N e^2(k) \quad (4.29)$$

em que  $e(k) = y(k) - \hat{y}(k)$ ,  $y(k)$  é a saída desejada e  $\hat{y}(k)$  é a saída do modelo nebuloso TSK. Sabendo que o modelo TSK é descrito pela equação (4.3) e sua saída é obtida pela equação (4.4), então, para o  $k$ -ésimo padrão de treinamento, o parâmetro das partes antecedentes e consequentes serão atualizados pelas Equações I e II, respectivamente, detalhadas abaixo.

#### Equação I (Ajuste da Parte Antecedente)

Os parâmetros das partes antecedentes do modelo nebuloso,  $\theta_{jl}^i$ , com  $l = 1$  e  $2$ , podem ser ajustados precisamente através da equação abaixo:

$$\Delta \theta_{jl}^i(k) = \eta (y(k) - \hat{y}(k)) (y^i(k) - \hat{y}(k)) \frac{1}{\sum_{i=1}^C w^i(k)} \frac{\partial w^i(k)}{\partial \theta_{jl}^i(k)} \quad (4.30)$$

em que  $\eta$  é uma taxa de aprendizagem,  $y(k)$  é a saída desejada,  $\hat{y}(k)$  é a saída do modelo nebuloso TSK e  $y^i(k)$  é a saída da  $i$ -ésima regra do modelo TSK. A obtenção dessa equação de ajuste está detalhada no Apêndice C.

Para calcularmos  $\frac{\partial w^i(k)}{\partial \theta_{jl}^i(k)}$ , primeiro é definido que  $w^i(k)$  é dado por  $\min_{j=1,2,\dots,n} A_j^i(x_j(k))$  e que  $j^*$  é igual ao índice  $j$  que define a minimização em  $w^i(k)$ , isto é,  $j^* = \arg \min_j A_j^i(x_j(k))$ . Outra

definição para  $j^*$  seria,  $j$  tal que  $w^i = \min_{j=1,2,\dots,n} A_j^i(x_j(k)) = A_{j^*}^i(x_{j^*}(k))$ . Então, quando  $j = j^*$  teremos, para  $l = 1$ :

$$\begin{aligned} \frac{\partial w^i(k)}{\partial \theta_{j_1}^i(k)} &= \frac{\partial A_j^i}{\partial \theta_{j_1}^i(k)} \\ \frac{\partial w^i(k)}{\partial \theta_{j_1}^i(k)} &= \frac{\partial}{\partial \theta_{j_1}^i(k)} \left( \exp \left\{ - \left( \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \right)^2 \right\} \right) \\ \frac{\partial w^i(k)}{\partial \theta_{j_1}^i(k)} &= \frac{2}{\theta_{j_2}^i(k)} \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \exp \left\{ - \left( \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \right)^2 \right\} \end{aligned}$$

E para  $l = 2$ , temos:

$$\begin{aligned} \frac{\partial w^i(k)}{\partial \theta_{j_2}^i(k)} &= \frac{\partial A_j^i}{\partial \theta_{j_2}^i(k)} \\ \frac{\partial w^i(k)}{\partial \theta_{j_2}^i(k)} &= \frac{2}{\theta_{j_2}^i(k)} \left( \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \right)^2 \exp \left\{ - \left( \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \right)^2 \right\} \\ \frac{\partial w^i(k)}{\partial \theta_{j_2}^i(k)} &= \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \frac{\partial w^i(k)}{\partial \theta_{j_1}^i(k)} \end{aligned}$$

Quando  $j \neq j^*$ ,  $\frac{\partial w^i(k)}{\partial \theta_{j_2}^i(k)} = \frac{\partial w^i(k)}{\partial \theta_{j_1}^i(k)} = 0$ , o que significa que nenhum ajuste será feito para esses casos.

### **Equação II (Ajuste da Parte Consequente)**

Os parâmetros das partes consequentes do modelo nebuloso definido em (4.3) podem ser ajustados através da equação:

$$\Delta a_j^i(k) = \zeta (y(k) - \hat{y}(k)) \frac{w^i(k) x_j(k)}{\sum_{i=1}^C w^i(k)} \quad (4.31)$$

em que  $\zeta$  é outra taxa de aprendizagem,  $y(k)$  é a saída desejada e  $\hat{y}(k)$  é a saída do modelo nebuloso TSK. A obtenção da equação de ajuste da parte consequente também está detalhada no Apêndice C.

Para analisar a validade das equações (4.30) e (4.31) obtidas, assumimos um modelo nebuloso com 3 regras e um dado amostral  $(\vec{x}(k), y(k))$ , como mostrado na Figura 4.2, onde  $y^1(k)$ ,  $y^2(k)$  e  $y^3(k)$  são as saídas respectivas das regras nebulosas  $R^1$ ,  $R^2$  e  $R^3$  para a entrada  $\vec{x}(k)$ ,  $\hat{y}(k)$  é a saída do modelo nebuloso e  $y(k)$  é a saída desejada.

A equação (4.30) é analisada analiticamente e intuitivamente, e para o caso intuitivo temos que, para  $\hat{y}(k)$  aproximar-se de  $y(k)$ , os parâmetros das partes premissas devem ser ajustados de tal forma que  $R^1$  e  $R^2$  sejam enfatizados e  $R^3$  seja enfraquecido. Agora, analiticamente falando da mesma situação, temos que  $\eta(y(k) - \hat{y}(k))(y^1(k) - \hat{y}(k))(1/\sum_{i=1}^C w^i(k))$  de  $R^1$  e  $\eta(y(k) - \hat{y}(k))(y^2(k) -$

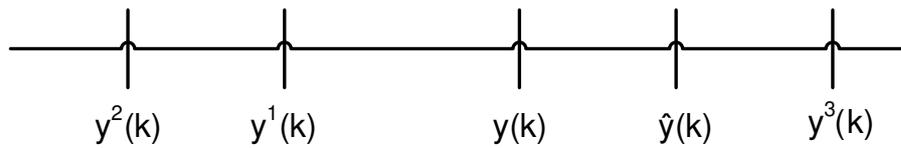


Fig. 4.2: Situação Exemplar de um Modelo Nebuloso com 3 Regras.

$\hat{y}(k))(1/\sum_{i=1}^C w^i(k))$  de  $R^2$  são valores positivos e a equação (4.30) será " $\Delta\theta_{jl}^i(k) = \text{um número positivo} \times \partial w^i(k)/\partial \theta_{jl}^i(k)$ ", o que significa que os parâmetros das partes premissas são ajustados de tal forma a enfatizar as regras  $R^1$  e  $R^2$ . No caso da regra nebulosa  $R^3$ ,  $\eta(y(k) - \hat{y}(k))(y^3(k) - \hat{y}(k))(1/\sum_{i=1}^C w^i(k))$  é negativo e a equação (4.30) torna-se, portanto, " $\Delta\theta_{jl}^i(k) = \text{um número negativo} \times \partial w^i(k)/\partial \theta_{jl}^i(k)$ ", o que significa que  $R^3$  é enfraquecido, concordando com a análise intuitiva. Uma análise similar pode ser feita para validar a equação (4.31).

## 4.4 Cálculo da Taxa Ótima de Controle de Fluxo

Esta seção é dedicada à derivação da taxa ótima de controle para regular a taxa da fonte controlável do esquema de controle de congestionamento adaptativo proposto, a partir da minimização da função custo  $J$  abaixo:

$$J(k+d) = E \left[ \frac{1}{2} (b(k+d) - b^\tau)^2 + \frac{\lambda}{2} \mu^2(k) \right], \quad (4.32)$$

em que  $\lambda$  é o fator de ponderação e  $E[\cdot]$  denota a esperança matemática. O efeito do fator de ponderação  $\lambda$  no desempenho do sistema será analisado posteriormente nas simulações. Na equação (4.32), a função custo  $J$  leva em consideração o erro de ajuste do tamanho da fila no *buffer*, isto é, considera a diferença entre o valor do tamanho da fila no *buffer*  $d$  passos à frente e o nível desejado para o tamanho da fila no *buffer*  $b^\tau$ .

**Proposição:** A taxa ótima de controle,  $\mu^o$ , que minimiza a função custo  $J$ , dada pela equação (4.32), e regula a taxa da fonte controlável é:

$$\mu^o(k) = \beta_0(k) \times \left( b^\tau - \sum_{i=1}^C \left( \alpha_0^i + \sum_{t=1}^{p+1} \alpha_t^i b(k-p-1+t) + \sum_{t=1}^q \beta_{t+p+1}^i \mu(k-q-1+t) \right) h^i(k) \right) \quad (4.33)$$

com

$$\beta_0(k) = \frac{\sum_{i=1}^C \beta_{p+q+2}^i h^i(k)}{\left( \sum_{i=1}^C \beta_{p+q+2}^i h^i(k) \right)^2 + \lambda}$$

Da equação (4.33), é possível perceber que a taxa ótima de controle de fluxo  $\mu^o(k)$  é obtida a

partir do vetor de entrada  $\vec{x}(k)$ , do vetor de parâmetros consequentes  $\vec{a}^i(k)$  e do nível desejado do tamanho da fila no *buffer*  $b^\tau$ , ou seja, o cálculo dessa taxa ótima utiliza os parâmetros obtidos do modelo *fuzzy* preditivo proposto e o nível desejado para o tamanho da fila no *buffer*.

**Prova:** Sabe-se que a saída do preditor *fuzzy* do tamanho da fila no *buffer* é dada por:

$$\hat{b}(k+d) = \sum_{i=1}^C b^i(k+d)h^i(k) \quad (4.34)$$

em que

$$h^i(k) = \frac{w^i(k)}{\sum_{i=1}^C w^i(k)} \quad \text{com} \quad w^i(k) = \min_{j=1 \dots n} A_j^i(\vec{\theta}_j^i; x_j(k)) \quad (4.35)$$

e

$$b^i(k+d) = \alpha_0^i + \alpha_1^i b(k-p) + \alpha_2^i b(k+p+1) + \dots + \alpha_{p+1}^i b(k) + \beta_{p+2}^i \mu(k-q) + \beta_{p+3}^i \mu(k-p+1) + \dots + \beta_{p+q+2}^i \mu(k) \quad (4.36)$$

Da função custo (4.32) temos:

$$J(k+d) = \frac{1}{2} E \left[ (b(k+d) - b^\tau)^2 + \lambda \mu^2(k) \right] \quad (4.37)$$

$$J(k+d) = \frac{1}{2} \left[ E (b(k+d) - b^\tau)^2 + \lambda \mu^2(k) \right] \quad (4.38)$$

Desde que o valor esperado para o tamanho da fila no *buffer*  $d$  passos à frente  $b(k+d)$  é a estimativa dada pelo preditor nebuloso proposto  $\hat{b}(k+d)$ , temos:

$$J(k+d) = \frac{1}{2} \left[ (\hat{b}(k+d) - b^\tau)^2 + \lambda \mu^2(k) \right] \quad (4.39)$$

Para obter a taxa de controle de tráfego ótima através da minimização da função custo  $J(k+d)$ , diferenciamos a equação (4.39) com respeito a  $\mu(k)$  e obtemos a seguinte igualdade:

$$\left[ \hat{b}(k+d) - b^\tau \right] \frac{\partial \hat{b}(k+d)}{\partial \mu(k)} + \lambda \mu(k) = 0 \quad (4.40)$$

A expressão  $\frac{\partial \hat{b}(k+d)}{\partial \mu(k)}$ , que aparece em (4.40), pode ser obtida derivando a equação (4.34), assim temos:

$$\frac{\partial \hat{b}(k+d)}{\partial \mu(k)} = \sum_{i=1}^C \beta_{p+q+2}^i h^i(k) \quad (4.41)$$

Desenvolveremos a equação (4.40) com base nos parâmetros do preditor proposto. Assim, substituindo (4.41) na equação (4.40) temos:

$$\begin{aligned} & \left[ \hat{b}(k+d) - b^\tau \right] \sum_{i=1}^C \beta_{p+q+2}^i h^i(k) + \lambda \mu(k) = 0 \\ & \left[ \hat{b}(k+d) - b^\tau \right] + \frac{\lambda}{\sum_{i=1}^C \beta_{p+q+2}^i h^i(k)} \mu(k) = 0 \end{aligned} \quad (4.42)$$

A substituição da equação (4.34) em (4.42) resulta da seguinte igualdade:

$$\left[ \left( \sum_{i=1}^C b^i(k+d) h^i(k) \right) - b^\tau \right] + \frac{\lambda}{\sum_{i=1}^C \beta_{p+q+2}^i h^i(k)} \mu(k) = 0 \quad (4.43)$$

Note que  $b^i(k+d)$  é dado pela equação (4.36). Isolando o termo  $\beta_{p+q+2}^i \mu(k)$  desta equação obtemos:

$$b^i(k+d) = \alpha_0^i + \sum_{t=1}^{p+1} \alpha_t^i b(k-p-1+t) + \sum_{t=1}^q \beta_{t+p+1}^i \mu(k-q-1+t) + \beta_{p+q+2}^i \mu(k) \quad (4.44)$$

Substituindo (4.44) em (4.43), obtemos:

$$\begin{aligned} & \left[ \left( \sum_{i=1}^C \left( \alpha_0^i + \sum_{t=1}^{p+1} \alpha_t^i b(k-p-1+t) + \sum_{t=1}^q \beta_{t+p+1}^i \mu(k-q-1+t) + \beta_{p+q+2}^i \mu(k) \right) h^i(k) \right) - \right. \\ & \left. - b^\tau \right] + \frac{\lambda}{\sum_{i=1}^C \beta_{p+q+2}^i h^i(k)} \mu(k) = 0 \end{aligned} \quad (4.45)$$

$$\begin{aligned} & \left( \sum_{i=1}^C \beta_{p+q+2}^i \mu(k) h^i(k) + \sum_{i=1}^C \left( \alpha_0^i + \sum_{t=1}^{p+1} \alpha_t^i b(k-p-1+t) + \sum_{t=1}^q \beta_{t+p+1}^i \mu(k-q-1+t) \right) \right. \\ & \left. \times h^i(k) \right) - b^\tau + \frac{\lambda}{\sum_{i=1}^C \beta_{p+q+2}^i h^i(k)} \mu(k) = 0 \end{aligned} \quad (4.46)$$

Colocando os termos que contém  $\mu(k)$  para o lado esquerdo da equação e os demais para o lado direito, obtemos:

$$\sum_{i=1}^C \beta_{p+q+2}^i \mu(k) h^i(k) + \frac{\lambda}{\sum_{i=1}^C \beta_{p+q+2}^i h^i(k)} \mu(k) = b^\tau - \sum_{i=1}^C \left( \alpha_0^i + \sum_{t=1}^{p+1} \alpha_t^i b(k-p-1+t) + \right.$$

$$+ \sum_{t=1}^q \beta_{t+p+1}^i \mu(k - q - 1 + t) \Big) h^i(k) \quad (4.47)$$

$$\begin{aligned} \mu(k) \left( \sum_{i=1}^C \beta_{p+q+2}^i h^i(k) + \frac{\lambda}{\sum_{i=1}^C \beta_{p+q+2}^i h^i(k)} \right) &= b^\tau - \sum_{i=1}^C \left( \alpha_0^i + \sum_{t=1}^{p+1} \alpha_t^i b(k - p - 1 + t) + \right. \\ &\left. + \sum_{t=1}^q \beta_{t+p+1}^i \mu(k - q - 1 + t) \right) h^i(k) \end{aligned} \quad (4.48)$$

Finalmente, temos que a taxa ótica de controle que regula a taxa da fonte controlável é dada por:

$$\mu^o(k) = \beta_0(k) \times \left( b^\tau - \sum_{i=1}^C \left( \alpha_0^i + \sum_{t=1}^{p+1} \alpha_t^i b(k - p - 1 + t) + \sum_{t=1}^q \beta_{t+p+1}^i \mu(k - q - 1 + t) \right) h^i(k) \right), \quad (4.49)$$

em que

$$\beta_0(k) = \frac{\sum_{i=1}^C \beta_{p+q+2}^i h^i(k)}{\left( \sum_{i=1}^C \beta_{p+q+2}^i h^i(k) \right)^2 + \lambda}.$$

Uma consideração importante deve ser feita: a taxa de controle de fluxo não pode ser negativa, contudo, na simulação isso pode ocorrer. Para evitar esse tipo de situação, um limite mínimo e máximo para a taxa deve ser estabelecido. Caso a taxa  $\mu^o(k)$  seja negativa, ela será configurada como zero. Visando evitar flutuações abruptas na taxa que causam grandes variações ou transbordamento no *buffer*, considera-se um limite superior  $\mu_h^o(k)$ . Esse limite é dado por  $\eta$  (capacidade do enlace de saída) menos o termo médio  $(1/N_w) \sum_{i=1}^{N_w} \nu(k - i)$  de um janela de tamanho  $N_w$ . Esse termo médio pode ser visto como uma estimativa da largura de banda consumida (dentro da janela  $N_w$ ) pelo tráfego não-controlável  $\nu(k)$ .

## 4.5 Considerações Finais

Neste capítulo, o esquema proposto de controle de fluxos não-linear, baseado na modelagem *fuzzy* linear local com adaptação em tempo real, foi apresentado. Inicialmente, é proposto um preditor adaptativo, via modelagem *fuzzy* TSK, que visa prever o comportamento da fila no *buffer*. Tal predição é composta de duas etapas: na primeira etapa, com a ajuda de um algoritmo de treinamento, os agrupamentos nebulosos e os parâmetros do modelo são obtidos de uma forma adaptativa. Na segunda etapa, um algoritmo de refinamento baseado no gradiente descendente foi usado para ajustar os parâmetros do modelo melhorando a sua precisão. Em seguida, com os parâmetros obtidos na predição, uma taxa

---

ótima para a fonte controlável é deduzida de tal forma que o tamanho da fila no *buffer* permaneça em torno de um nível desejado. Os resultados das simulações serão mostrados nos capítulos que seguem.



# Capítulo 5

## Avaliação do Preditor Proposto

### 5.1 Introdução

No capítulo anterior apresentamos o esquema de controle de congestionamento proposto, o qual faz uso de um algoritmo de predição para prever o comportamento da rede, mais especificamente, o tamanho da fila no *buffer*. Em seguida, com os parâmetros obtidos na predição, esse esquema controla a taxa da fonte de tal forma que o tamanho da fila no *buffer* se aproxime de um nível desejado e a variância do tamanho da fila se torne mínima. O presente capítulo busca avaliar particularmente o preditor proposto e, no capítulo seguinte, o esquema de controle de congestionamento como um todo será analisado. Nas seções seguintes deste capítulo, serão realizadas avaliações comparativas entre o desempenho do preditor proposto e o desempenho de outros dois diferentes tipos de preditores, quando aplicados à predição de traços de tráfego reais.

Inicialmente, uma introdução à predição linear é detalhada e os preditores utilizados nas análises comparativas são descritos. Em seguida, as medidas de desempenho de predição utilizadas são apresentadas. Quanto à caracterização do traço de tráfego a ser predito, avaliamos as características fractais através do diagrama multiescala linear. Uma descrição da taxa média e da taxa de pico de tais fluxos de tráfego também é apresentada. Por fim, o desempenho de predição do preditor proposto é analisado utilizando traços de tráfego monofractais e multifractais e os resultados obtidos são exibidos e comentados.

### 5.2 Predição Linear

A predição linear de um processo estocástico nada mais é do que uma particularização da filtragem linear. Um filtro é dito linear se obedece ao princípio da superposição, que diz: a resposta do filtro à entrada  $av_1(t) + bw_1(t)$  é o sinal  $av_2(t) + bw_2(t)$ , onde  $a$  e  $b$  são valores constantes e  $v_2(t)$  e  $w_2(t)$  são

as respostas do filtro quanto este for excitado, separadamente, pelas entradas  $v_1(t)$  e  $w_1(t)$ . No caso de processos estocásticos a tempo discreto, os filtros projetados serão filtros com resposta ao impulso de tempo discreto.

O problema da predição consiste de projetar um filtro linear de tempo discreto, dado um conjunto de amostras de entrada  $u(0), u(1), \dots$ , cuja saída  $y(k)$  fornece uma estimação da saída desejada  $d(k)$ .

Uma abordagem bastante utilizada para o problema de otimização de filtro é minimizar o valor do erro quadrático médio, onde o erro é definido pela diferença entre a saída desejada e a saída real do filtro. Para entradas estacionárias, a solução resultante é comumente denominada de filtro de Wiener, o qual é ótimo no sentido quadrático médio [41].

Considere o filtro linear transversal de tempo discreto da Figura 5.1. As entradas do filtro correspondem ao vetor  $U(k) = [u(k), u(k-1), \dots, u(k-n+1)]$  e o filtro é caracterizado pelos pesos  $w_0, w_1, \dots, w_{n-1}$ . Em um dado instante  $k$ , a saída do filtro é dada pela seguinte equação:

$$y(k) = \sum_{j=0}^{n-1} w_j u(k-j), \quad (5.1)$$

ou seja, a saída do filtro consiste do produto escalar entre as entradas do filtro  $U(k) = [u(k), u(k-1), \dots, u(k-n+1)]$  e o vetor de pesos  $W = [w_0, w_1, \dots, w_{n-1}]$ .

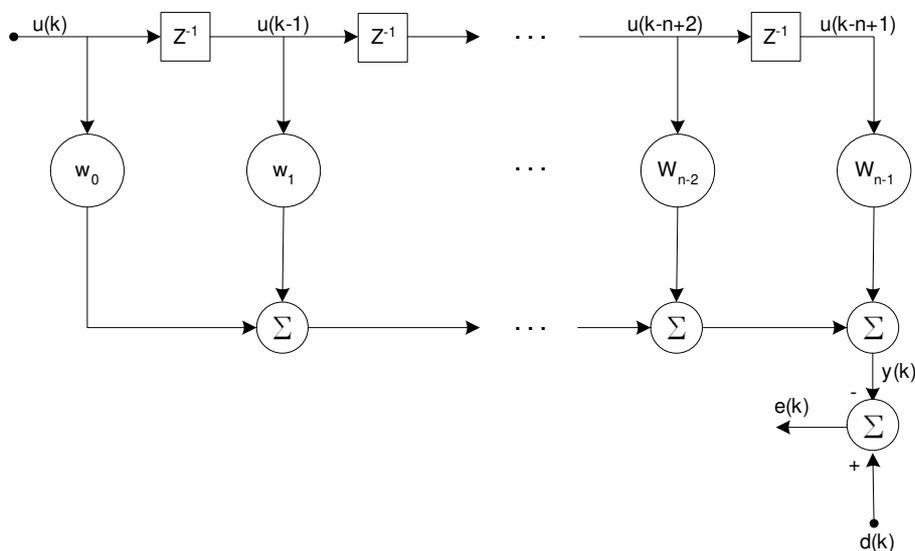


Fig. 5.1: Filtro Linear Transversal.

A saída do filtro é utilizada para estimar a saída desejada  $d(k)$ . Essa estimativa apresenta um erro

$e(k)$ , definido pela diferença entre a resposta desejada e a saída do filtro, ou seja,

$$e(k) = d(k) - \sum_{j=0}^{n-1} w_j u(k-j). \quad (5.2)$$

Desejamos então tornar esse erro tão pequeno quanto possível segundo algum critério estatístico. O critério utilizado será a minimização do erro quadrático médio. Dessa forma, a função custo do filtro transversal é dada por:

$$J = \frac{1}{2} E[e^2(k)] = \frac{1}{2} E[d^2(k)] - E \left[ \sum_{j=0}^{n-1} w_j u(k-j) d(k) \right] + \frac{1}{2} E \left[ \sum_{j=0}^{n-1} \sum_{i=0}^{n-1} w_j w_i u(k-j) u(k-i) \right]. \quad (5.3)$$

Dado que a esperança matemática  $E[\cdot]$  é uma operação linear, a equação (5.3) pode ser reescrita como:

$$J = \frac{1}{2} E[d^2(k)] - \sum_{j=0}^{n-1} w_j E[u(k-j)d(k)] + \frac{1}{2} \sum_{j=0}^{n-1} \sum_{i=0}^{n-1} w_j w_i E[u(k-j)u(k-i)]. \quad (5.4)$$

Analisando a equação acima e identificando as três esperanças, temos:

- Para a esperança  $E[d^2(k)]$  temos que:

$$\sigma_d^2 = E[d^2(k)], \quad (5.5)$$

onde  $\sigma_d^2$  é a variância da resposta desejada  $d(k)$ , assumida como tendo média zero.

- Para a segunda esperança, temos:

$$p(-j) = E[u(k-j)d(k)], \quad (5.6)$$

onde  $p(-j)$  é a correlação cruzada entre a entrada  $u(k-j)$  e a resposta desejada  $d(k)$ . É importante ressaltar que  $u(k-j)$  e  $d(k)$  são considerados conjuntamente estacionários<sup>1</sup>.

- Para a terceira esperança temos:

$$r_u(i-j) = E[u(k-j)u(k-i)], \quad (5.7)$$

<sup>1</sup>Dois processos  $X(t)$  e  $Y(t)$  são conjuntamente estacionários se, e somente se,  $X(t)$  e  $Y(t)$  forem estacionários e a correlação cruzada entre eles,  $R_{XY}(t_1, t_2)$ , só depender explicitamente de  $\tau = t_1 - t_2$ .

onde  $r_u(i-j)$  é a função de auto-correlação das entradas do filtro para uma distância entre elas de  $i-j$ .

Deste modo, a equação (5.4) pode ser reescrita da seguinte forma:

$$J = \frac{1}{2}\sigma_d^2 - \sum_{j=0}^{n-1} w_j p(-j) + \frac{1}{2} \sum_{j=0}^{n-1} \sum_{i=0}^{n-1} w_j w_i r_u(i-j). \quad (5.8)$$

Logo, quando as entradas do filtro e a resposta desejada forem conjuntamente estacionárias, a função custo, ou o erro quadrático médio  $J$ , será precisamente uma função de segunda ordem dos pesos do filtro. Consequentemente, a dependência da função custo  $J$  em relação aos pesos  $w_0, w_1, \dots, w_{n-1}$  poderá ser vista como uma superfície no espaço de dimensão  $(n+1)$  com  $n$  graus de liberdade, representado pelos pesos do filtro. Essa superfície é comumente chamada de superfície de erro e é caracterizada por um mínimo global. É justamente nesse ponto que o filtro obtém o seu estado ótimo, ou seja, o erro quadrático médio atinge o seu valor mínimo  $J_{min}$ . Deste modo, o vetor gradiente  $\nabla(J)$  nesse ponto de mínimo será identicamente igual a zero, ou seja:

$$\nabla_j(J) = 0, \quad j = 0, 1, \dots, n-1, \quad (5.9)$$

onde  $\nabla_j(J)$  é a derivada da função custo  $J$  com relação ao  $j$ -ésimo peso  $w_j$ .

Derivando a equação (5.8) em relação ao peso  $w_j$ , temos:

$$\nabla_j(J) = \frac{\partial J}{\partial w_j} = -p(-j) + \sum_{i=0}^{n-1} w_i r_u(i-j). \quad (5.10)$$

Substituindo a equação (5.10) na equação (5.9), encontramos o conjunto ótimo de pesos  $w_0^o, w_1^o, \dots, w_{n-1}^o$  para o filtro da Figura 5.1:

$$\sum_{i=0}^{n-1} w_i^o r_u(i-j) = p(-j), \quad j = 0, 1, \dots, n-1. \quad (5.11)$$

Esse sistema representado pela equação 5.11 é conhecido como as equações de *Wiener-Hopf* e o filtro cujos pesos satisfazem essas equações é denominado filtro de *Wiener*.

As expressões do filtro de *Wiener* também podem ser expressas na forma matricial:

$$RW^o = P, \quad (5.12)$$

onde

$$R = E[U(k)U(k)^T] = \begin{bmatrix} r(0) & r(1) & \dots & r(n-1) \\ r(1) & r(0) & \dots & r(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(n-1) & r(n-2) & \dots & r(0) \end{bmatrix},$$

é a matriz de correlação  $M \times M$  do vetor de entrada  $U(k) = [u(k), \dots, u(k-n+1)]^T$ ,

$$P = E[U(k)d(k)] = [p(0), p(-1), \dots, p(1-n)], \quad (5.13)$$

é o vetor de correlação cruzada  $M \times 1$  entre as entradas do filtro e a resposta desejada  $d(k)$ , e

$$W^o = [w_0^o, w_1^o, \dots, w_{n-1}^o], \quad (5.14)$$

é o vetor  $M \times 1$  de pesos ótimos do filtro transversal da Figura 5.1.

Para solucionar as equações de *Wiener-Hopf*, descrito em (5.12), para  $W^o$ , assume-se que a matriz de correlação  $R$  é não-singular. Então, multiplicando os membros da equação (5.12) pela inversa da matriz de correlação ( $R^{-1}$ ), temos:

$$W^o = R^{-1}P. \quad (5.15)$$

Da equação (5.15), pode-se concluir que, para obter os pesos do filtro através das equações de *Wiener-Hopf*, devemos calcular a inversa da matriz de correlação  $R$ . Essa inversão pode se tornar muito complexa se o número de pesos do filtro for muito grande. Sendo assim, uma maneira de evitarmos essa inversão é utilizando o método do gradiente descendente [41]. Nesse método, os pesos do filtro assumem uma forma variante no tempo e seus valores são ajustados de uma forma iterativa, ao longo da superfície de erro, com o objetivo de movê-los progressivamente em direção à solução ótima.

Seja  $w_j(k)$  o valor do peso  $w_j$  calculado na iteração ou tempo discreto  $k$  pelo método do gradiente descendente. De uma forma correspondente, o gradiente da superfície de erro do filtro em relação aos pesos, assume uma forma variante no tempo, dada por:

$$\nabla_j(J(k)) = -p(-j) + \sum_{i=0}^{n-1} w_i(k)r_u(i-j). \quad (5.16)$$

De acordo com o método do gradiente descendente, o ajuste aplicado ao peso  $w_j(k)$ , no instante  $k$ , é definido por [41]:

$$\Delta w_j(k) = -\eta \nabla_j(J(k)), \quad j = 0, 1, \dots, n-1, \quad (5.17)$$

onde  $\eta$  é uma constante positiva denominada de taxa de aprendizagem. Dado o valor do  $j$ -ésimo peso  $w_j(k)$  no instante  $k$ , o valor atualizado para esse peso no instante  $k + 1$  será:

$$w_j(k + 1) = w_j(k) + \Delta w_j(k) = w_j(k) - \eta \nabla_j(J(k)), \quad j = 0, 1, \dots, n - 1. \quad (5.18)$$

Escrevendo na forma matricial, temos:

$$W(k + 1) = W(k) - \eta \nabla J(k), \quad (5.19)$$

onde

$$\nabla J(k) = -P + RW(k). \quad (5.20)$$

Logo, substituindo (5.20) em (5.19) temos que o vetor de pesos para o instante  $k + 1$  é dado por:

$$W(k + 1) = W(k) + \eta[P - RW(k)]. \quad (5.21)$$

### 5.2.1 Algoritmo LMS (*Least Mean Square*)

O algoritmo dos mínimos quadrados (LMS) é um importante membro da família de algoritmos baseados em gradiente estocástico[41]. Uma característica importante desse algoritmo é a sua simplicidade, visto que ele não requer medições das funções de correlação pertinentes nem requer inversão matricial. Na verdade, é a simplicidade do algoritmo LMS que o torna o algoritmo padrão dentre todos os algoritmos de filtragem adaptativa [41].

O algoritmo LMS é um algoritmo de filtragem linear adaptativa que consiste de dois processos básicos: um processo de filtragem, que executa o produto da entrada com os pesos correspondentes produzindo a estimativa da resposta desejada, e em seguida calcula o erro de estimação através da comparação dessa estimativa com o valor da resposta desejada; e um processo adaptativo, que ajusta automaticamente o conjunto de pesos de acordo com o erro de estimação.

Como vimos anteriormente, para que os valores dos pesos converjam para os valores correspondentes à solução ótima de *Wiener*, através do algoritmo gradiente descendente, é necessário que se tenha uma medição exata do vetor gradiente  $\nabla J(k)$  em cada iteração e que o parâmetro  $\eta$ , da equação (5.19), seja escolhido apropriadamente. Na prática, é muito difícil de se obter esse cálculo de  $\nabla J(k)$ , visto que é exigido o conhecimento da matriz de correlação  $R$  das entradas e do vetor de correlação cruzada  $P$  entre as entradas e a resposta desejada (equação (5.20)). Desse modo, o vetor gradiente será estimado através dos dados de entrada, ou seja, os pesos serão atualizados através de um algoritmo que se adapta de acordo com os dados de entrada disponíveis.

A fim de obtermos uma estimação do vetor gradiente  $\nabla J(k)$ , a estratégia utilizada é introduzir

estimações da matriz de correlação  $R$  e do vetor de correlação cruzada  $P$  na equação (5.20). A escolha mais simples para as estimativas de  $R$  e  $P$  são as baseadas nos valores amostrais do vetor de entradas e na resposta desejada, essas estimativas são chamadas de estimativas instantâneas e são definidas abaixo:

$$R_{est}(k) = U(k)U^T(k) \quad (5.22)$$

e

$$P_{est}(k) = U(k)d(k). \quad (5.23)$$

Substituindo (5.22) e (5.23) em (5.20), a estimativa instantânea do vetor gradiente será:

$$\nabla_{est}(J(k)) = -U(k)d(k) + U(k)U^T(k)W_{est}(k) \quad (5.24)$$

Aplicando a estimativa do vetor gradiente  $\nabla_{est}(J(k))$ , dada pela equação (5.24), no algoritmo do gradiente descendente representado pela equação (5.21), obtemos a seguinte relação recursiva para atualização do vetor de pesos:

$$W_{est}(k+1) = W_{est}(k) + \eta U(k)[d(k) - U^T(k)W_{est}(k)] \quad (5.25)$$

Equivalentemente, a equação (5.25) pode ser reescrita na forma de três relações básicas, como segue:

### 1. Saída do Filtro

$$y(k) = U^T(k)W_{est}(k) \quad (5.26)$$

### 2. Estimativa do erro

$$e(k) = d(k) - y(k) \quad (5.27)$$

### 3. Adaptação do Vetor de Pesos

$$W_{est}(k+1) = W_{est}(k) + \eta U(k)e(k) \quad (5.28)$$

O procedimento iterativo do cálculo do vetor de pesos é inicializado com um valor inicial  $W_{est}(0) = 0$ . No método do gradiente descendente, o vetor de pesos  $W(k)$  começa no valor inicial  $W(0)$  e segue uma trajetória bem definida (ao longo da superfície de erro) que eventualmente termina na solução ótima  $W^o$ . Para isto, o parâmetro taxa de aprendizagem  $\eta$  deve ser escolhido apropriadamente. No algoritmo LMS, o vetor de pesos  $W_{est}(k)$  representa uma estimativa de  $W(k)$ , e segue uma trajetória

aleatória em direção à solução ótima  $W^o$ . É por essa razão que o LMS é conhecido como um "algoritmo de gradiente estocástico". Quando o número de iterações aproxima-se do infinito, o valor de  $W_{est}(k)$  torna-se muito próximo ao valor da solução ótima  $W^o$ .

### 5.2.2 Algoritmo RLS (*Recursive Least Square*)

O algoritmo RLS utiliza o método dos mínimos quadrados para calcular recursivamente os pesos do filtro, ou seja, a partir da estimação mínima quadrada do vetor de pesos no instante  $k - 1$ , a estimação atualizada do vetor de pesos do filtro no instante  $k$  é calculada à medida que novas observações vão sendo obtidas. A taxa de convergência do algoritmo RLS é mais rápida do que a obtida no algoritmo LMS, em contrapartida, os custos requeridos com o aumento da complexidade computacional são maiores [41].

Na implementação recursiva do método dos mínimos quadrados, a partir de condições iniciais e da nova observação obtida, atualiza-se a estimação anterior do vetor de pesos do filtro. A função custo do filtro RLS é definida em [41] e dada por:

$$J(k) = \sum_{i=1}^k \beta(k, i) |e(i)|^2 \quad 1 \leq i \leq k, \quad (5.29)$$

onde  $k$  é o comprimento variável dos dados de entrada,  $\beta(k, i)$  é o fator de esquecimento e  $e(i)$  é a diferença entre a resposta desejada  $d(i)$  e a saída  $y(i)$  produzida pelo filtro transversal cujas entradas no instante  $i$  é  $u(i), u(i - 1), \dots, u(i - n + 1)$ . Ou seja,  $e(i)$  é dado por:

$$e(i) = d(i) - y(i) = d(i) - W^T(k)U(i), \quad (5.30)$$

onde  $U(i)$  é o vetor de entrada dado por  $U(i) = [u(i), u(i - 1), \dots, u(i - n + 1)]^T$  e  $W(k)$  é o vetor de pesos no instante  $k$  dado por  $W(k) = [w_0(k), w_1(k), \dots, w_{n-1}(k)]$ . O vetor de pesos do filtro transversal permanece fixo durante o intervalo de observação  $1 \leq i \leq k$ .

O fator de esquecimento  $\beta(k, i)$  é usado para garantir que os dados de um passado distante sejam "esquecidos" de tal forma a acompanhar as variações estatísticas dos dados de entrada quando o filtro opera em um ambiente não-estacionário. O fator de esquecimento exponencial é o mais utilizado e é definido por:

$$\beta(k, i) = \lambda^{k-i} \quad i = 1, 2, \dots, k, \quad (5.31)$$

onde  $\lambda$  é uma constante positiva próxima e menor do que 1. Quando  $\lambda$  é igual a 1, temos o caso do método ordinário dos mínimos quadrados. A inversa de  $1 - \lambda$  é considerada como uma medida de memória do algoritmo, onde o caso especial de  $\lambda = 1$  corresponde à memória infinita. Assim, no

método dos mínimos quadrados ponderados exponencialmente, a função custo é dada por:

$$J(k) = \sum_{i=1}^k \lambda^{k-i} |e(i)|^2. \quad (5.32)$$

O valor ótimo do vetor de pesos,  $W_{est}(k)$ , com o qual a função custo atinge seu mínimo é definido pela equação matricial

$$\Phi(k)W_{est}(k) = \Theta(k). \quad (5.33)$$

O termo  $\Phi(k)$ , denominado matriz de correlação,  $n \times n$ , é definido pela seguinte equação:

$$\Phi(k) = \sum_{i=1}^k \lambda^{k-i} U(i)U^H(i). \quad (5.34)$$

O vetor de correlação cruzada  $\Theta(k)$ ,  $n \times 1$ , entre as entradas do filtro e a resposta desejada é dado por:

$$\Theta(k) = \sum_{i=1}^k \lambda^{k-i} U(i)d^*(i), \quad (5.35)$$

onde  $d^*(i)$  indica o complexo conjugado de  $d(i)$ .

Isolando o termo correspondente a  $i = k$  do resto do somatório no lado direito da equação (5.34), temos:

$$\Phi(k) = \lambda \left[ \sum_{i=1}^{k-1} \lambda^{k-1-i} U(i)U^H(i) \right] + U(k)U^H(k). \quad (5.36)$$

Por definição, o termo entre os colchetes é igual a matriz de correlação  $\Phi(k-1)$ . Assim, a matriz de correlação da entrada pode ser calculada de forma recursiva através da seguinte equação:

$$\Phi(k) = \lambda\Phi(k-1) + U(k)U^H(k), \quad (5.37)$$

onde  $\Phi(k-1)$  é o valor anterior da matriz de correlação e o produto  $U(k)U^H(k)$  corresponde a um termo de correção na operação de atualização.

Da mesma forma, pode-se usar a equação (5.35) para derivar uma expressão recursiva para atualizar o vetor de correlação cruzada  $\Theta(k)$  entre a entrada e a resposta desejada:

$$\Theta(k) = \lambda\Theta(k-1) + U(k)d^*(k). \quad (5.38)$$

Para calcular o vetor de pesos estimado  $W_{est}(k)$  da equação (5.33), é preciso primeiro determinar a inversa da matriz de correlação  $\Phi(k)$ . Na prática, tenta-se evitar tal cálculo devido a complexidade

computacional exigida, principalmente quando  $n$  é um valor alto. Em adição, é ideal que se efetue a estimação mínima quadrada do vetor de pesos recursivamente para  $k = 1, 2, \dots, \infty$ . Para atingir esses dois objetivos, é usado o teorema da inversão matricial, conforme descrito no Apêndice B. Assim, aplicando o teorema da inversão matricial na equação recursiva (5.37), obtemos a inversa da matriz de correlação  $\Phi(k)$  dada por:

$$\Phi^{-1}(k) = \lambda^{-1}\Phi^{-1}(k-1) - \frac{\lambda^{-2}\Phi^{-1}(k-1)U(k)U^H(k)\Phi^{-1}(k-1)}{1 + \lambda^{-1}U^H(k)\Phi^{-1}(k-1)U(k)}. \quad (5.39)$$

Fazendo a inversa da matriz de correlação igual a:

$$P(k) = \Phi^{-1}(k) \quad (5.40)$$

e

$$K(k) = \frac{\lambda^{-1}P(k-1)U(k)}{1 + \lambda^{-1}U^H(k)P(k-1)U(k)}, \quad (5.41)$$

a equação (5.39) pode ser reescrita como:

$$P(k) = \lambda^{-1}P(k-1) - \lambda^{-1}K(k)U^H(k)P(k-1). \quad (5.42)$$

Reescrevendo a equação (5.41) obtemos:

$$K(k) = \lambda^{-1}P(k-1)U(k) - \lambda^{-1}K(k)U^H(k)P(k-1)U(k)$$

e

$$K(k) = [\lambda^{-1}P(k-1) - \lambda^{-1}K(k)U^H(k)P(k-1)]U(k). \quad (5.43)$$

O termo entre colchetes da equação (5.43) é igual a  $P(k)$  conforme a equação (5.42). Deste modo, temos:

$$K(k) = P(k)U(k). \quad (5.44)$$

Substituindo  $P(k) = \Phi^{-1}(k)$  na equação acima temos:

$$K(k) = \Phi^{-1}(k)U(k). \quad (5.45)$$

Ou seja, o vetor  $K(k)$ , conhecido como ganho, é definido como o vetor entrada  $U(k)$  modificado pela inversa da matriz de correlação  $\Phi(k)$ .

Para desenvolver uma equação recursiva para atualizar a estimação do vetor de pesos  $W_{est}(k)$ ,

inicialmente a equação (5.33) pode ser escrita da seguinte forma:

$$W_{est}(k) = \Phi^{-1}(k)\Theta(k). \quad (5.46)$$

Substituindo  $P(k) = \Phi^{-1}(k)$  na equação (5.46) obtemos:

$$W_{est}(k) = P(k)\Theta(k). \quad (5.47)$$

Substituindo o vetor de correlação cruzada  $\Theta(k)$  da equação (5.38) em (5.47):

$$W_{est}(k) = \lambda P(k)\Theta(k-1) + P(k)U(k)d^*(k). \quad (5.48)$$

E substituindo a equação (5.42) apenas no primeiro termo  $P(k)$  do lado direito da equação (5.48), temos:

$$\begin{aligned} W_{est}(k) &= P(k-1)\Theta(k-1) - K(k)U^H(k)P(k-1)\Theta(k-1) + P(k)U(k)d^*(k), \\ W_{est}(k) &= \Phi^{-1}(k-1)\Theta(k-1) - K(k)U^H(k)\Phi^{-1}(k-1)\Theta(k-1) + P(k)U(k)d^*(k), \\ W_{est}(k) &= W_{est}(k-1) - K(k)U^H(k)W_{est}(k-1) + P(k)U(k)d^*(k). \end{aligned} \quad (5.49)$$

Finalmente, sabendo que  $K(k) = P(k)U(k)$ , a equação (5.49) pode ser reescrita como:

$$\begin{aligned} W_{est}(k) &= W_{est}(k-1) + K(k)[d^*(k) - U^H(k)W_{est}(k-1)], \\ W_{est}(k) &= W_{est}(k-1) + K(k)\alpha^*(k), \end{aligned} \quad (5.50)$$

onde  $\alpha(k)$  é a inovação [41] definida por

$$\begin{aligned} \alpha(k) &= d(k) - U^T(k)W_{est}^*(k-1), \\ \alpha(k) &= d(k) - W_{est}^H(k-1)U(k). \end{aligned} \quad (5.51)$$

O produto  $W_{est}^H(k-1)U(k)$  representa uma estimativa da resposta desejada  $d(k)$  baseado na estimativa do vetor de pesos feito no instante  $k-1$ . Assim,  $\alpha(k)$  pode ser considerado como um erro de estimativa do instante anterior.

Resumindo, as equações (5.41), (5.51), (5.50) e (5.42), nessa ordem, formam o algoritmo RLS.

---



---

 Algoritmo RLS
 

---

$$K(k) = \frac{\lambda^{-1}P(k-1)U(k)}{1+\lambda^{-1}U^H(k)P(k-1)U(k)} \quad (5.41)$$

$$\alpha(k) = d(k) - W_{est}^H(k-1)U(k) \quad (5.51)$$

$$W_{est}(k) = W_{est}(k-1) + K(k)\alpha^*(k) \quad (5.50)$$

$$P(k) = \lambda^{-1}P(k-1) - \lambda^{-1}K(k)U^H(k)P(k-1) \quad (5.42)$$


---



---

A equação (5.51) descreve a operação de filtragem do algoritmo, onde o filtro transversal é excitado para calcular o erro de estimação anterior  $\alpha(k)$ . A equação (5.50) representa a operação adaptativa do algoritmo, onde o vetor de pesos é atualizado através do incremento do seu valor anterior por um valor igual ao complexo conjugado do erro de estimação anterior  $\alpha(k)$  vezes o vetor ganho  $K(k)$ , daí o nome vetor ganho. As equações (5.41) e (5.42) são utilizadas para atualizar o vetor ganho  $K(k)$ .

### 5.3 Medidas de Desempenho de Predição

Para avaliarmos o desempenho do preditor proposto, adotamos a medida de erro de predição do tipo Erro Quadrático Médio (EQM).

#### 5.3.1 Erro Quadrático Médio - EQM

Seja  $\hat{y}_i$  o valor predito da amostra de um processo  $Y$  em um determinado instante e  $y_i$  o valor desejado para esta amostra. O erro quadrático médio (EQM) é dado pelo valor médio dos desvios do valor predito  $\hat{y}_i$  em relação ao seu valor desejado  $y_i$ , ou seja,

$$EQM = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2, \quad (5.52)$$

onde  $N$  é o número de amostras observadas.

O erro quadrático médio é uma medida de erro absoluto, portanto é bastante influenciado pela amplitude do sinal predito. Para evitar esse problema, são adotadas medidas de erro relativas, conhecidas como erros quadráticos médios normalizados (EQMN). Neste trabalho, duas medidas de erro relativas são adotadas: o EQM normalizado em relação à variância da série predita, e o EQM nor-

malizado em relação ao erro quadrático médio do preditor ótimo para o processo de passeio aleatório (*random walk*) [44].

### 5.3.2 Erro Quadrático Médio Normalizado 1 - EQMN1

Seja  $\sigma_y^2$  a variância do processo  $Y$ , dada por  $\sigma_y^2 = \frac{1}{N} \sum_{i=1}^N (\mu - y_i)^2$ , onde  $\mu$  é a média do processo  $Y$ . O erro quadrático médio do tipo 1 é dado por:

$$EQMN1 = \frac{EQM}{\sigma_y^2} = \frac{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2}{\frac{1}{N} \sum_{i=1}^N (\mu - y_i)^2} \quad (5.53)$$

Analisando a equação (5.53), o EQMN1 pode ser interpretado como a razão entre os EQM's de dois preditores, onde, no numerador, o valor predito é dado por  $\hat{y}_i$ , enquanto no denominador o valor predito é dado pela média do processo  $\mu$ . Logo, se um preditor apresenta um valor de EQMN1 igual ou inferior a 1, podemos concluir que esse preditor apresenta desempenho igual ou superior a um preditor que apenas estime o valor futuro como a média do processo.

### 5.3.3 Erro Quadrático Médio Normalizado 2 - EQMN2

Considerando o preditor simplificado descrito por:

$$\hat{y}_i = y_{i-1} \quad (5.54)$$

Ou seja, o valor predito é igual ao valor da amostra imediatamente anterior. Para o processo de passeio aleatório (*random walk*), esse é o preditor ótimo no sentido do erro quadrático médio [44].

Seja  $\hat{y}_i^{pa}$  o valor predito da amostra  $y_i$  do processo  $Y$ , estimado através do preditor descrito pela equação (5.54). O erro quadrático médio do tipo 2 será dado por:

$$EQMN2 = \frac{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2}{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i^{pa} - y_i)^2} \quad (5.55)$$

Analisando a equação acima, um EQMN2 próximo de 1 nos indica um preditor com desempenho próximo ao de um preditor que estima o valor futuro como o valor da amostra imediatamente anterior.

A partir das análises e consideração feitas, concluímos que o desempenho de um determinado preditor só é considerado satisfatório quando as medidas EQMN1 e EQMN2 forem inferiores a 1.

## 5.4 Caracterização das Séries de Tráfego Utilizadas

Antes de analisarmos o desempenho do preditor proposto, os traços utilizados nas simulações devem ser caracterizados. Para tal, faremos uma análise dos traços sob uma perspectiva da teoria fractal, verificando o comportamento fractal dos traços de tráfego considerados.

### 5.4.1 A Característica Multifractal

Com origem no Latim *fractus*, significando "quebrado", a palavra fractal foi originalmente utilizada para descrever objetos que eram muito irregulares para se adequarem à geometria euclidiana tradicional [45]. Em um sentido determinístico, uma forma geométrica é fractal ou auto-similar caso preserve a mesma aparência quando observada sobre todas as escalas no espaço ou no tempo.

Pesquisas sobre tráfego de redes encontraram-se com a teoria dos fractais a partir da publicação do trabalho de Leland, Taqqu, Willinger e Wilson [46]. Leland et al. constataram experimentalmente que o tráfego coletado na rede Ethernet do *Bellcore Morristown Research and Engineering Center* exibia propriedades fractais tais como auto-similaridade e dependência a longo prazo. Após ter sido evidenciada por Leland et al., estudos posteriores verificaram a presença de propriedades fractais também no tráfego gerado por transmissão de vídeo a taxa variável [47], tráfego em redes WAN [48], tráfego *World Wide Web* [49], dentre outros. Foi constatado que tais propriedades, com destaque para a dependência de longa duração, influenciam fortemente no desempenho das redes, não sendo adequadamente modeladas por processos estocásticos markovianos. Vários estudos abordaram a medida e a modelagem da característica de dependência de longo prazo existente nos dados. Constatou-se que tal dependência é uma importante característica do tráfego, possuindo relevantes implicações em diversas áreas, tais como teoria das filas [50] e projeto de redes.

Diferentes modelos matemáticos foram propostos na literatura com o objetivo de representar a característica auto-similar constatada no tráfego de redes. Identificado como o modelo que, de maneira mais simples, incorpora matematicamente as características auto-similares observadas no tráfego, o modelo de tráfego Browniano fracionário (*fractional Brownian traffic*, fBt) tornou-se amplamente utilizado [51]. Entretanto, observou-se que, enquanto em escalas de tempo da ordem de centenas de milisegundos e maiores, o comportamento do tráfego era bem representado por modelos auto-similares, em escalas de tempo da ordem de centenas de milisegundos e menores, as características de tais modelos afastavam-se das apresentadas pelo tráfego real. Tal constatação motivou a busca por modelos de tráfego mais abrangentes, que possibilitassem uma descrição mais completa do tráfego de redes. Investigações envolvendo tráfego WAN TCP/IP [52][53] verificaram que diferentes propriedades do tráfego observadas em pequenas escalas de tempo eram convenientemente descritas utilizando-se a análise multifractal. Tais propriedades, presentes em pequenas escalas de tempo, são

oriundas da ação dos protocolos predominantes nas redes em questão, e dos mecanismos fim-a-fim de controle de congestionamento existentes, determinantes no comportamento do fluxo de informações entre diferentes camadas na hierarquia de protocolos TCP/IP [54].

Ao variar-se a escala temporal de observação de processos auto-similares, o comportamento observado é único e invariante no tempo, ou seja, tais processos apresentam um comportamento em escala monofractal. Esses processos dependem apenas de um único parâmetro, o parâmetro de Hurst ( $H$ ), que define a intensidade da auto-similaridade do processo. Por outro lado, processos multifractais são vistos como uma generalização dos processos monofractais, possibilitando leis de comportamento em escala variantes no tempo e, portanto, proporcionando maior flexibilidade em descrever processos irregulares que são localizados no tempo.

Um processo estocástico  $Z(t)$  é dito multifractal se satisfaz

$$E(|Z(t)|^q) = c(q)t^{\tau(q)+1}, \quad \forall t \in T, q \in Q, \quad (5.56)$$

onde  $T$  e  $Q$  são intervalos de números reais, e  $\tau(q)$  e  $c(q)$  são funções com o domínio  $Q$ . Além disso, assume-se que  $T$  e  $Q$  possuem comprimentos não nulos, e que  $0 \in T$ ,  $[0, 1] \subseteq Q$ . A função  $\tau(q)$  é chamada de função de escala do processo multifractal ou função partição. Trata-se de uma função côncava com  $\tau(0) = -1$ . Para processos monofractais temos que  $\tau(q) = Hq - 1$  e  $c(q) = E(|X(1)|^q)$ , ou seja, a função de escala  $\tau(q)$  é linear e completamente determinada pelo índice  $H$  [45].

### 5.4.2 Verificação do Comportamento Multifractal do Tráfego - Diagrama Multiescala Linear

A estimação da característica multifractal de um processo pode ser baseada na estimação da função de partição  $\tau(q)$  do processo, na estimação da regularidade do processo e no seu espectro multifractal. A partir da definição de um processo multifractal dada pela equação (5.56), é possível perceber que a descrição de um processo multifractal envolve tanto o conhecimento da função  $\tau(q)$  quanto da função  $c(q)$ . Um método que verifica a característica fractal de um processo e estima as funções  $\tau(q)$  e  $c(q)$  é apresentado a seguir e pode ser encontrado em [55].

Considere uma escala de referência de tempo  $T_0$  e defina uma escala de tempo  $T_j$  como  $T_j = 2^{-j}T_0$ . Em cada escala de tempo  $T_j$  temos que  $t_{j,k} = (kT_j, (k+1)T_j)$  denota a  $k$ -ésima unidade de tempo de comprimento  $T_j$ . Considere também um processo de tráfego estacionário  $Y$ . Na escala de tempo  $j$ ,  $Y_{j,k}$  é a quantidade de tráfego que chega na unidade de tempo  $t_{j,k}$ . Então  $Y_j = Y_{j,k}$  com  $k \in \mathbb{Z}$  representa o processo de tráfego que é observado na escala de tempo  $j$ . Os coeficientes de

wavelet Haar  $W_{j,k}$  do processo de tráfego na escala de tempo  $j$  são definidos como [55]:

$$W_{j,k} = 2^{j/2}(Y_{j+1,2k} - Y_{j+1,2k+1}). \quad (5.57)$$

Assim, a função energia do processo de tráfego na escala de tempo  $j$  é dada por:

$$\varepsilon_j = E[W_{j,k}^2], \quad (5.58)$$

onde  $E[.]$  denota a esperança matemática.

A análise multifractal de um processo compara o escalonamento de diferentes momentos wavelets para estimar a regularidade local no processo em questão. A principal ferramenta para tal estimação é a função partição wavelet  $S_j(q)$ , a qual é uma generalização da função energia wavelet dada em (5.58). Tal função partição wavelet é definida como:

$$S_j(q) = E[W_{j,k}^q]. \quad (5.59)$$

Para certos processos multifractais, a função partição wavelet  $S_j(q)$  se comporta assintoticamente ( $j \rightarrow \infty$ ) da seguinte forma:

$$\log_2 S_j(q) \sim q \times \text{constante} + j\alpha_q \quad (5.60)$$

Assim, a inclinação de  $\log_2 S_j(q)$  em relação a  $j$  provê uma estimativa de  $\alpha_q$ . Em processos monofractais,  $\alpha_q$  varia linearmente com relação a  $q$ , já em processos multifractais essa variação é não-linear.

Uma ferramenta disponível em [56] conhecida como Diagrama Multiescala mostra se um processo tem tendência a apresentar comportamento monofractal ou multifractal em um intervalo  $[j1, j2]$ . Para realizar tal teste, dois parâmetros foram definidos,  $\zeta_q = \alpha_q - q/2$  e  $h_q = \zeta_q/q$ . O Diagrama Multiescala (MD) mostra no eixo vertical os expoentes  $\zeta_q$  e no eixo horizontal os valores de  $q$ . O Diagrama Multiescala Linear (LMD) mostra no eixo vertical os expoentes  $h_q$  e no eixo horizontal os valores de  $q$ . Se um processo é monofractal, o diagrama de  $\zeta_q$  resulta em uma reta e  $h_q$  em uma constante. E no caso de um processo multifractal, o diagrama de  $\zeta_q$  resulta em uma curva e  $h_q$  em uma inconstância [55].

### 5.4.3 Traços Utilizados nas Simulações

Nas simulações serão utilizados traços WAN TCP/IP que correspondem ao registro do tráfego de pacotes IP transmitidos em períodos de uma hora, entre a *Digital Equipment Corporation* e o resto

do mundo (*dec-pkt-1.tcp*, *dec-pkt-2.tcp* e *dec-pkt-3.tcp*)[57]. Tais traços foram agregados na escala de  $512ms$  por apresentarem comportamento multifractal [58].

Outros dois traços utilizados correspondem à amostras de tráfego LAN Ethernet disponibilizadas pela Bellcore. Tais arquivos referem-se ao tráfego trocado entre a Bellcore e o ambiente externo, via internet (traço *OctExt*) e ao tráfego interno circulado nos laboratórios da Bellcore (traço *OctInt*) [59]. O traço *OctExt* possui escala de agregação de  $1min$  para verificar uma agregação maior do tráfego, e o traço *OctInt* possui escala de  $100ms$ .

Outros traços também utilizados são *10-7-S-1*, *3-7-I-1*, *4-7-I-1*, *4-7-I-2*, *4-7-I-5* e *4-7-I-9*, os quais foram capturados em um servidor da rede da Petrobrás através de um analisador de dados da Acterna modelo DA350, com a resolução (*time stamp*) de 32 microsegundos. Tais traços de tráfego foram agregados em uma escala de  $100ms$ . Os arquivos de tráfego agregado designados com a letra "S" significam que tais arquivos foram capturados junto aos servidores de aplicações. Já os arquivos de tráfego agregado designados pela letra "I" foram capturados em roteador de acesso Internet.

Na Tabela 5.1 é especificada a taxa média e a taxa de pico de cada um dos traços de tráfego citados. A escolha de tais traços de tráfego, WAN TCP/IP e LAN Ethernet, justifica-se por terem sido empregados em outros trabalhos relacionados à controle de congestionamento e predição [60][61] [62] [63] [64], portanto, servindo bem ao propósito de comparação.

Tab. 5.1: Taxa Média e Taxa de Pico dos Traços de Tráfego Utilizados nas Simulações.

<b>Traço de Tráfego</b>	<b>Taxa Média [bytes/seg]</b>	<b>Taxa de Pico [bytes/seg]</b>
<i>dec-pkt-1.tcp</i>	$1.2168 \times 10^5$	$34.0859 \times 10^4$
<i>dec-pkt-2.tcp</i>	$6.4457 \times 10^4$	$27.8661 \times 10^4$
<i>dec-pkt-3.tcp</i>	$9.2496 \times 10^4$	$27.5649 \times 10^4$
<i>OctExt</i>	$1.4487 \times 10^3$	$2.2958 \times 10^4$
<i>OctInt</i>	776.8357	$2.2410 \times 10^3$
<i>10-7-S-1</i>	648.6298	$4.3771 \times 10^3$
<i>3-7-I-1</i>	$9.2533 \times 10^3$	$15.5123 \times 10^4$
<i>4-7-I-1</i>	$1.2113 \times 10^4$	$15.9012 \times 10^4$
<i>4-7-I-2</i>	$1.1342 \times 10^4$	$16.1603 \times 10^4$
<i>4-7-I-5</i>	$1.3586 \times 10^4$	$17.2204 \times 10^4$
<i>4-7-I-9</i>	$1.2198 \times 10^4$	$15.0736 \times 10^4$

### 5.4.4 Verificação do Comportamento Multifractal dos Traços de Tráfego Utilizados

Como já citamos anteriormente, utilizamos o algoritmo disponível em [56] para analisar a tendência dos traços quanto ao seu comportamento fractal. Para cada traço analisado o algoritmo simulado na linguagem MatLab gera o gráfico de  $\zeta_q$  e  $h_q$ . Dos gráficos gerados pelo algoritmo para os traços WAN TCP/IP da *Digital Equipment* e mostrados na Figura 5.2 temos que, segundo o diagrama multiescala linear, a série de tráfego *dec-pkt-2* apresenta um comportamento que tende a ser mais monofractal do que as séries *dec-pkt-1* e *dec-pkt-3*. Da Figura 5.2 é possível observar que, para a série de tráfego *dec-pkt-2*, o gráfico de  $\zeta_q$  se aproxima de uma reta e  $h_q$  de uma constante.

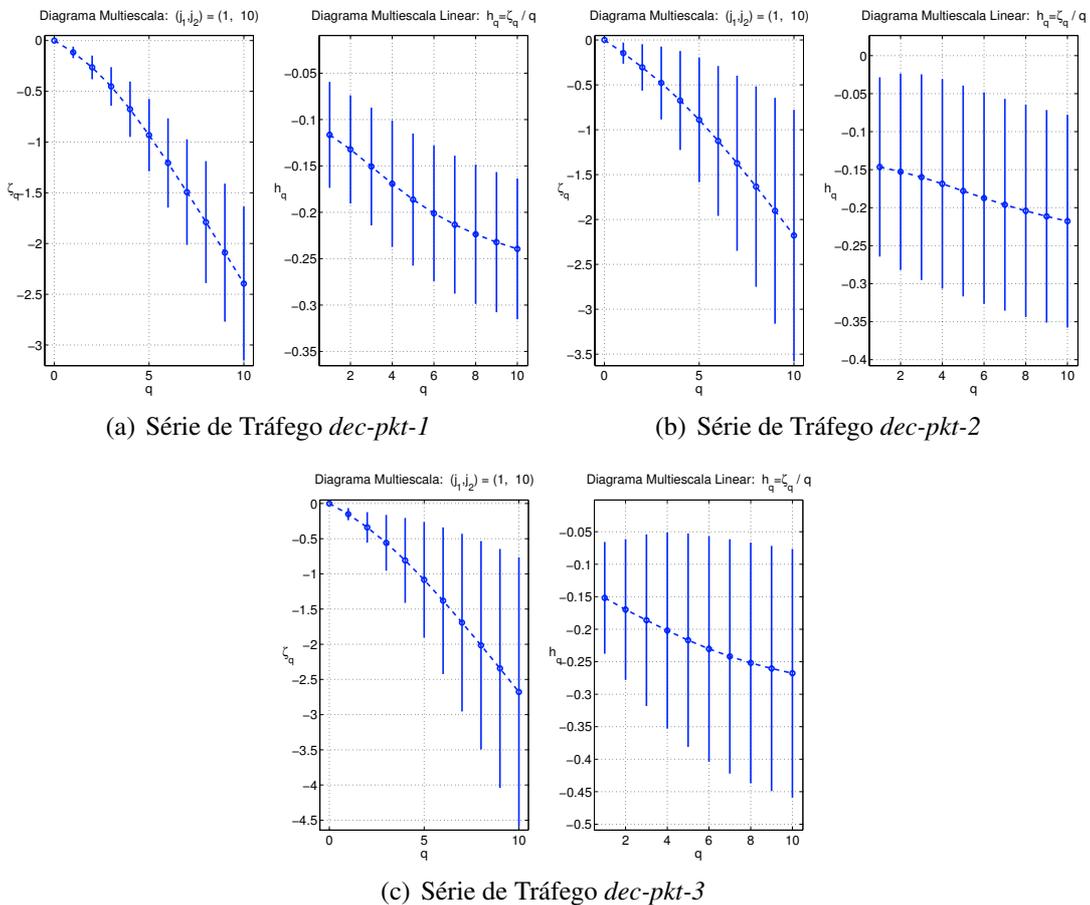
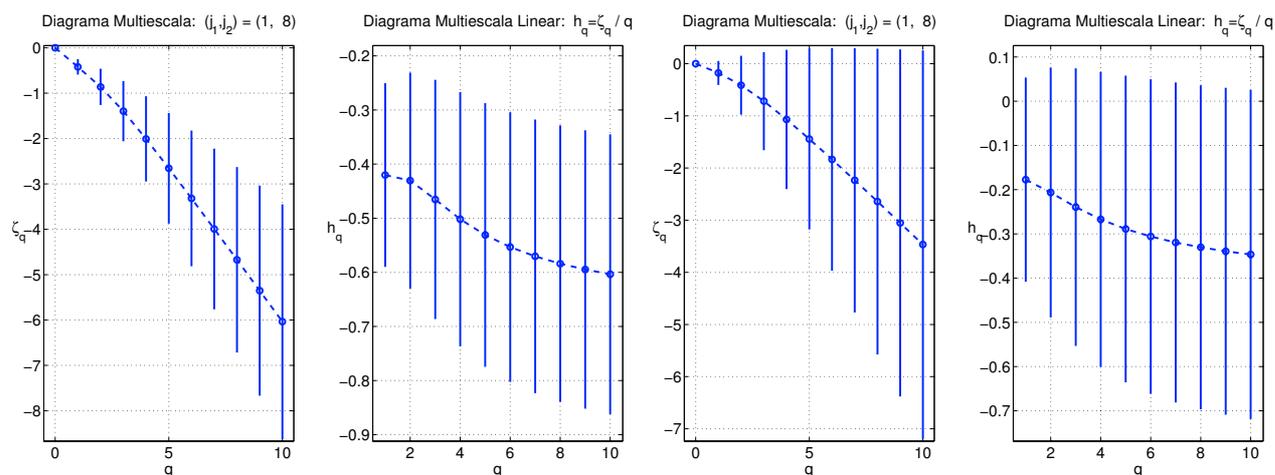


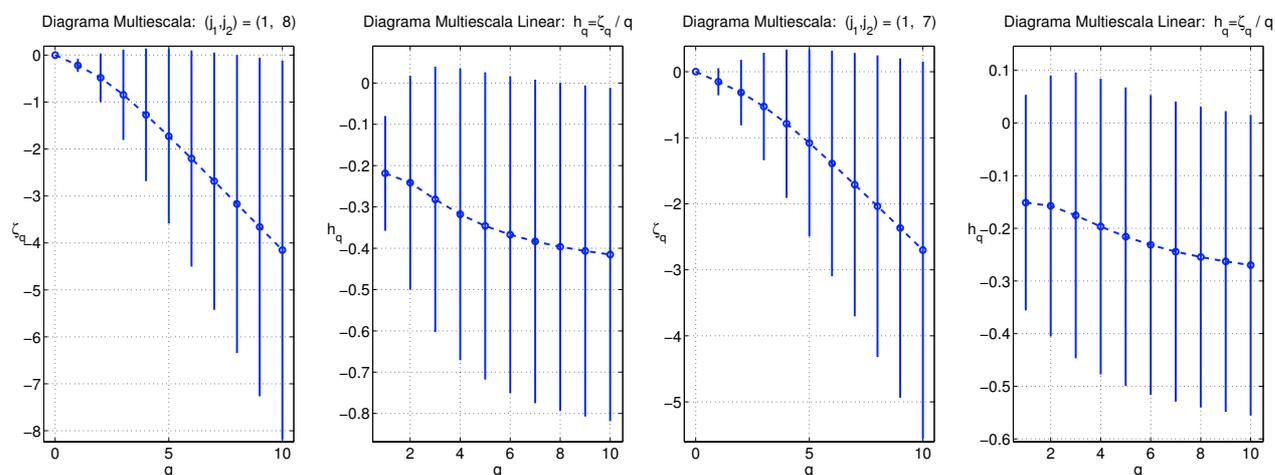
Fig. 5.2: Análise da Característica Fractal dos Traços TCP/IP.

Os diagramas multiescala linear para os traços da Petrobrás são mostrados na Figura 5.3 onde é possível observar não-linearidade em MD e inconstância em LMD para todos os traços, indicando evidências de "multifractalidade".



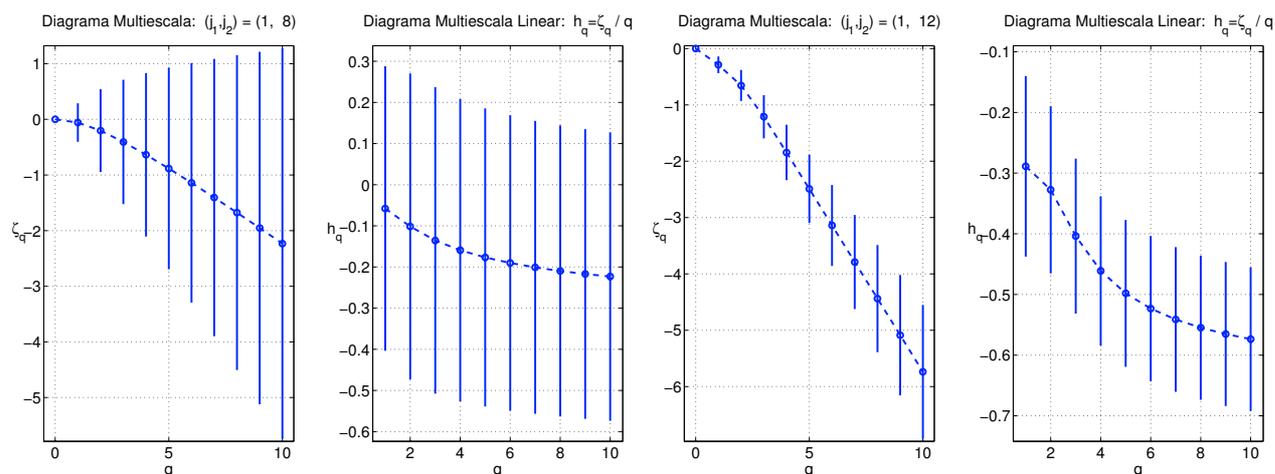
(a) Série de Tráfego 3-7-I-1

(b) Série de Tráfego 4-7-I-1



(c) Série de Tráfego 4-7-I-2

(d) Série de Tráfego 4-7-I-5



(e) Série de Tráfego 4-7-I-9

(f) Série de Tráfego 10-7-S-1

Fig. 5.3: Análise da Característica Fractal dos Traços da Petrobrás.

Dos diagramas multiescala linear para os traços LAN Ethernet da Bellcore, conforme mostrados na Figura 5.4, é possível notar que a série de tráfego *OctInt* possui características monofractais e a série de tráfego *OctExt* pode ser considerada multifractal. Pesquisas realizadas nesse tráfego detectaram a presença de surtos ao longo de diversas escalas de tempo, o que caracteriza uma manifestação da auto-similaridade, confirmando o que o diagrama multiescala linear nos mostra [46].

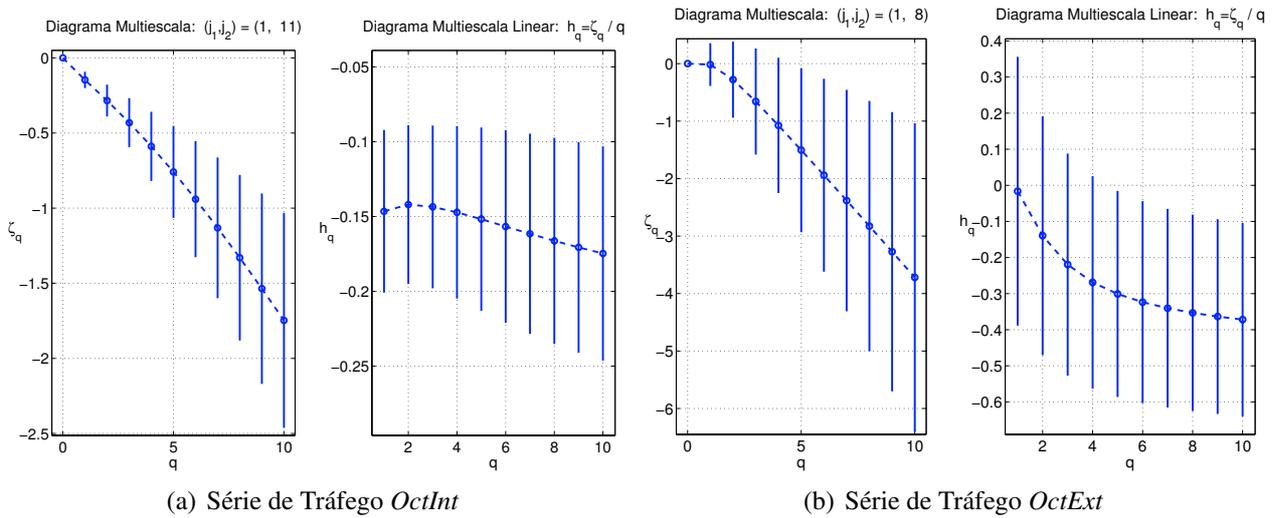


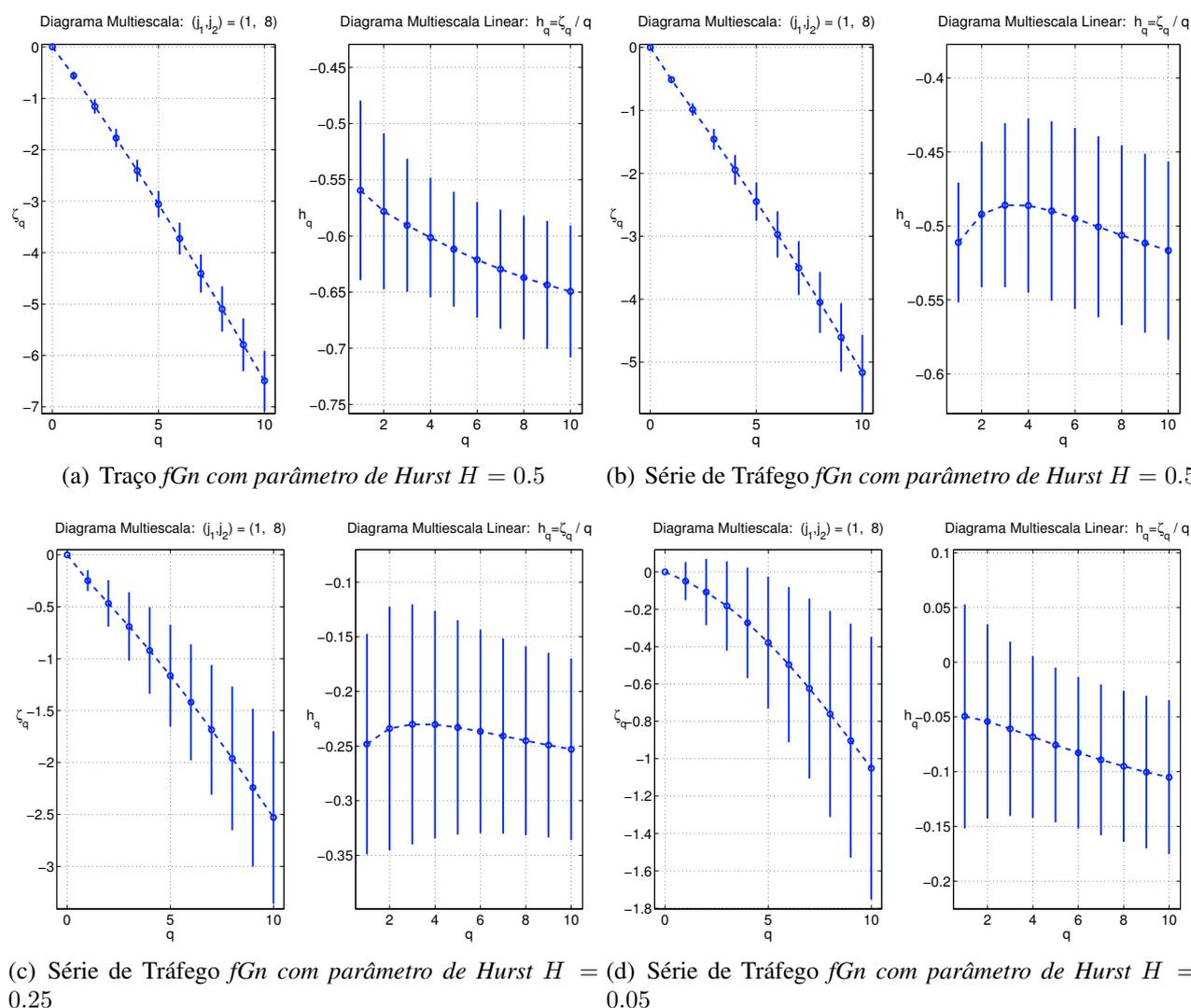
Fig. 5.4: Análise da Característica Fractal dos Traços LAN Ethernet.

Os comportamentos fractais de algumas séries fGn (*fractional Gaussian noise*) são verificados através dos diagramas multiescala linear mostrados na Figura 5.5. Sabe-se que um processo fGn com parâmetro de Hurst  $H$  possui  $\alpha_q = q(H + 1/2)$ , e sabendo que  $\zeta_q = \alpha_q - q/2$  e  $h_q = \zeta_q/q$ , temos que para o processo fGn,  $\zeta_q = qH$  e  $h(q) = H$ , ou seja,  $\zeta_q$  é linear e  $h(q)$  é uma constante igual ao parâmetro de Hurst  $H$  [55]. Logo, trata-se de um processo monofractal. Através da Figura 5.5 essa característica pode ser verificada.

## 5.5 Análise de Desempenho de Predição

Para a análise do desempenho de predição do preditor proposto, através de simulações na linguagem MatLab, são utilizados os traços WAN TCP/IP *dec-pkt-1.tcp* e *dec-pkt-2.tcp*. Considerando os aspectos fractais apresentados na seção anterior, para tais traços, será utilizada a escala de  $512ms$  nos experimentos a seguir, pois nessa tal escala, o traço *dec-pkt-1.tcp* apresentou comportamento multifractal e o traço *dec-pkt-2.tcp* apresentou comportamento monofractal.

Outros dois traços utilizados referem-se aos tráfego LAN Ethernet disponibilizadas pela Bellcore *OctExt* e *OctInt* na escala de  $1min$  para o *OctExt* e  $100ms$  para o *OctInt*.

Fig. 5.5: Análise da Característica Fractal dos Traços  $fGn$ .

Os traços da Petrobrás  $10-7-S-1$ ,  $4-7-I-9$  também foram utilizados para avaliar o desempenho do preditor proposto. Esses traços de tráfego foram agregados em uma escala de  $100ms$  e, conforme visto na seção anterior, possuem características multifractais.

Dois preditores lineares são utilizados para comparar o desempenho do preditor proposto. O primeiro é o filtro linear adaptativo LMS (*Least Mean Square*), pertencente à família de algoritmos de gradiente estocástico, que não requer medições das funções de correlação nem tampouco inversão da matriz de auto-correlação. O segundo é o filtro linear RLS (*Recursive Least Square*), o qual é visto como um caso especial do filtro de Kalman [41]. Tais filtros são bastante utilizados na literatura como métodos comparativos, servindo bem ao nosso propósito.

Para a escolha do número de amostras passadas do tráfego, ou seja, a ordem do preditor, sabe-se que

que se utilizarmos poucas amostras temos um menor tempo de processamento das informações, em contrapartida, esse pequeno número de amostras utilizado pode comprometer a qualidade da predição. Uma análise da ordem do preditor foi realizada utilizando o traço de tráfego da Petrobrás 4-7-I-9 e os resultados são mostrados na Tabela 5.2. Dos resultados obtidos, é possível notar que as medidas de desempenho se tornam melhores quando a ordem aumenta até  $n = 6$ , e a partir desse valor essas medidas começam a aumentar caracterizando uma queda na qualidade da predição. Deste modo, a ordem escolhida para o preditor foi  $n = 6$ . As taxas de aprendizagem do algoritmo de refinamento foram estipuladas de tal forma a obter melhores resultados possíveis, assim o valor encontrado foi  $\eta = 0.62$  e  $\zeta = 0.72$ . Para os preditores utilizados na comparação, foi utilizado, no caso do RLS, um fator de esquecimento de  $\lambda = 0.99$ . E para o caso do LMS, foi utilizado uma taxa de aprendizagem de  $\eta = 0.03$ . A ordem para os preditores de comparação foi a mesma utilizada pelo preditor proposto, ou seja,  $n = 6$ . Os resultados obtidos são mostrados a seguir.

Tab. 5.2: Resultados de Desempenho do Preditor Proposto para Diferentes Ordens  $n$ .

Ordem	EQM	EQMN1	EQMN2
$n = 2$	0.0284	1.3183	0.5311
$n = 4$	0.0093	0.4301	0.1733
$n = 6$	0.0066	0.3045	0.1227
$n = 8$	0.0217	1.0161	0.4095
$n = 10$	0.0258	1.1952	0.4815

### 5.5.1 Predição dos Traços de Tráfego

As Figuras 5.6 e 5.7 exibem as respostas do preditor de tráfego proposto para os tráfegos TCP/IP *dec-pkt-1* e *dec-pkt-2*, para o tráfego Ethernet *OctExt* e *OctInt* e para os traços da Petrobras *10-7-S-1* e *4-7-I-9*, respectivamente. Para uma melhor visualização, são mostradas as predições em duas janelas de 200 amostras para cada traço de tráfego. Analisando as Figuras 5.6 e 5.7 vemos a capacidade do preditor em acompanhar as variações do traço de tráfego real.

As Tabelas 5.3, 5.4, 5.5, 5.6, 5.7 e 5.8 mostram as medidas de desempenho (EQM, EQMN1 e EQMN2) obtidas utilizando o preditor nebuloso proposto assim como para os outros preditores utilizados na análise comparativa.

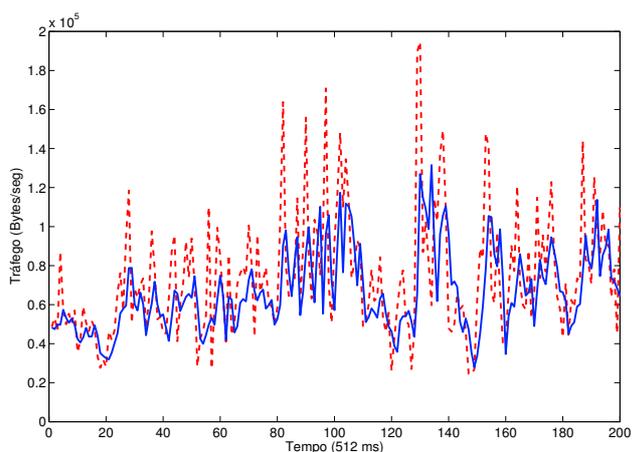
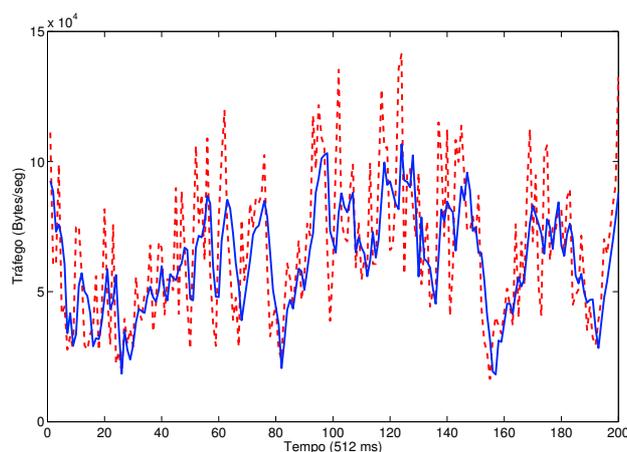
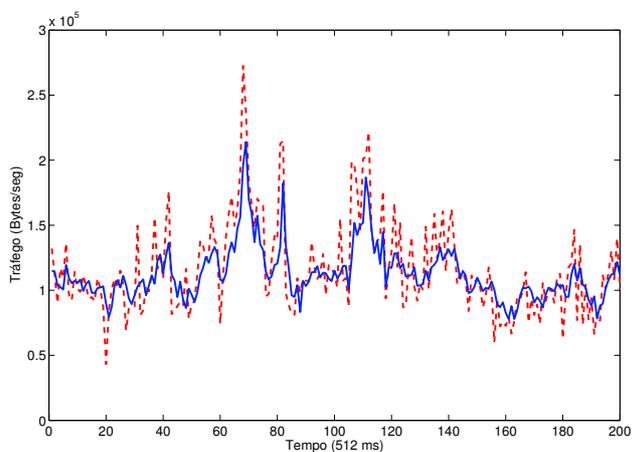
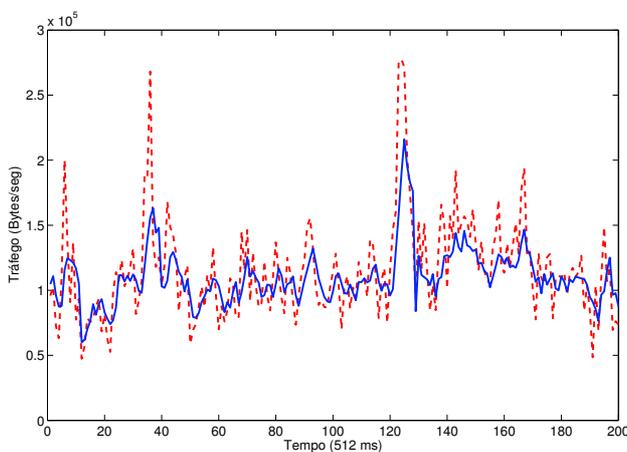
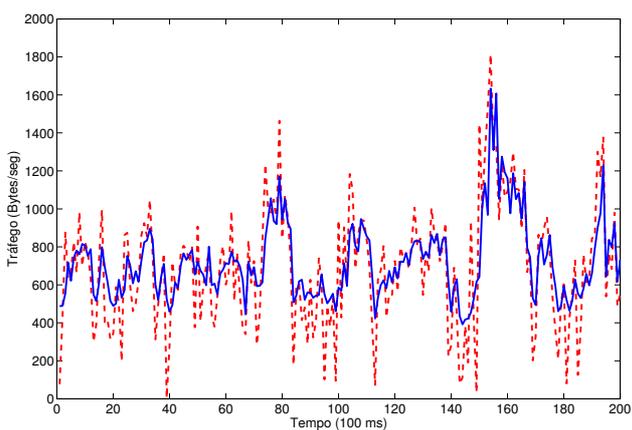
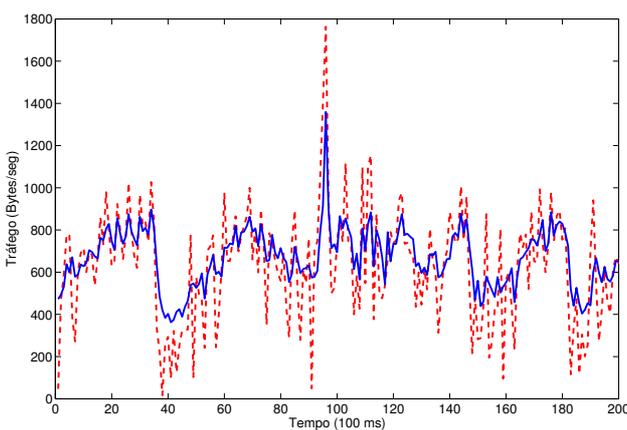
(a) Traço de Tráfego *dec-pkt-1*(b) Traço de Tráfego *dec-pkt-1*(c) Traço de Tráfego *dec-pkt-2*(d) Traço de Tráfego *dec-pkt-2*(e) Traço de Tráfego *OctInt*(f) Traço de Tráfego *OctInt*

Fig. 5.6: Predição a um Passo das Séries Temporais de Tráfego. Série Real (Linha Tracejada), Série Predita (Linha Cheia).

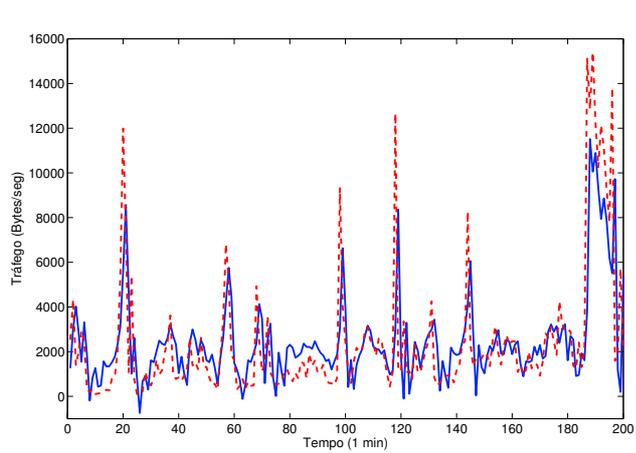
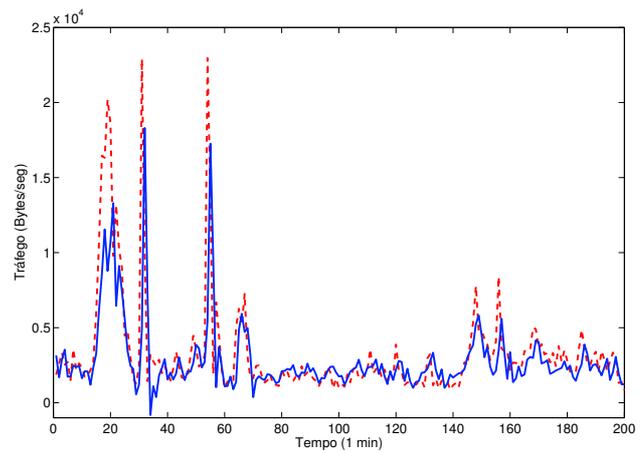
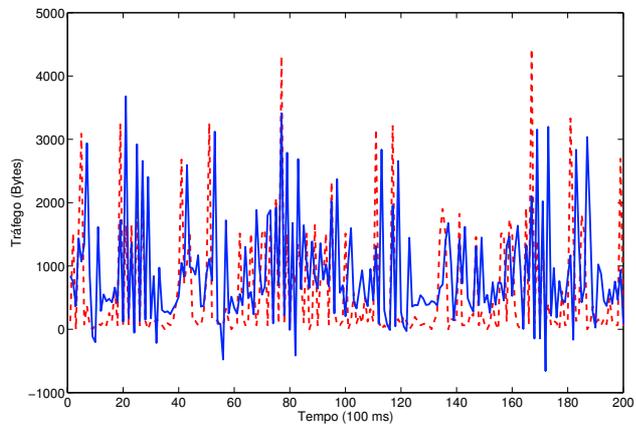
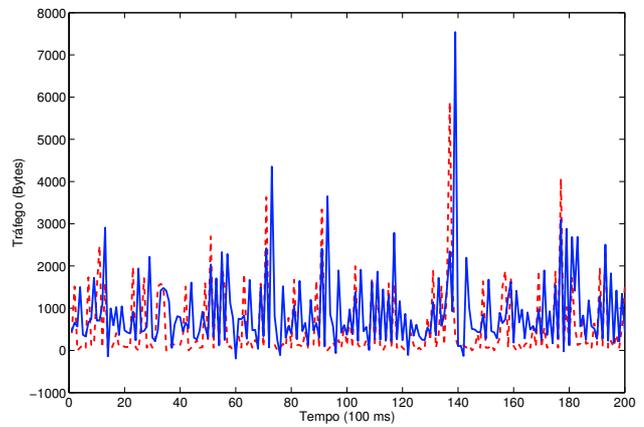
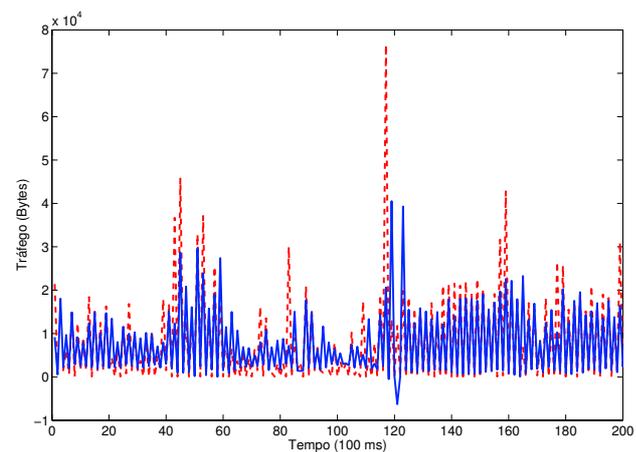
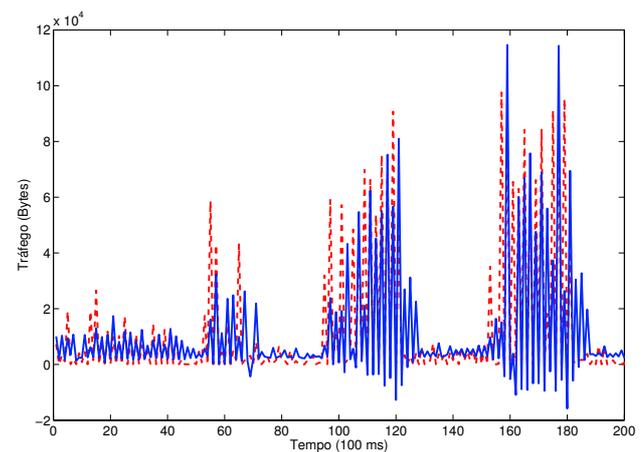
(a) Traço de Tráfego *OctExt*(b) Traço de Tráfego *OctExt*(c) Traço de Tráfego *FL10-7-S-1*(d) Traço de Tráfego *FL10-7-S-1*(e) Traço de Tráfego *4-7-I-9*(f) Traço de Tráfego *4-7-I-9*

Fig. 5.7: Predição a um Passo das Séries Temporais de Tráfego. Série Real (Linha Tracejada), Série Predita (Linha Cheia).

Tab. 5.3: Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Traço de Tráfego Real *dec-pkt-1*.

<b>Preditor</b>	<b>EQM</b>	<b>EQMN1</b>	<b>EQMN2</b>
Proposto	0.0067	0.5340	0.6106
RLS	0.0092	0.7312	0.8363
LMS	0.0108	0.8578	0.9811

Tab. 5.4: Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Traço de Tráfego Real *dec-pkt-2*.

<b>Preditor</b>	<b>EQM</b>	<b>EQMN1</b>	<b>EQMN2</b>
Proposto	0.0071	0.4113	0.5577
RLS	0.0107	0.6193	0.8331
LMS	0.0125	0.7204	0.9691

Tab. 5.5: Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Traço de Tráfego Real *OctExt*.

<b>Preditor</b>	<b>EQM</b>	<b>EQMN1</b>	<b>EQMN2</b>
Proposto	0.0058	0.5385	0.9283
RLS	0.0065	0.6010	1.0365
LMS	0.0078	0.7211	1.2436

Tab. 5.6: Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Traço de Tráfego Real *OctInt*.

<b>Preditor</b>	<b>EQM</b>	<b>EQMN1</b>	<b>EQMN2</b>
Proposto	0.0156	0.4691	0.4787
RLS	0.0250	0.7529	0.7652
LMS	0.0263	0.7927	0.8057

Tab. 5.7: Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Traço de Tráfego Real *FL10-7-S-1*.

Preditor	EQM	EQMN1	EQMN2
Proposto	0.0025	1.7997	0.8601
RLS	0.0034	2.4427	1.1666
LMS	0.0035	2.4903	1.1893

Tab. 5.8: Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Traço de Tráfego Real *4-7-I-9*.

Preditor	EQM	EQMN1	EQMN2
Proposto	0.0066	0.3045	0.1227
RLS	0.0083	0.3839	0.1548
LMS	0.0119	0.5503	0.2220

Analisando os resultados apresentados nas Tabelas 5.3, 5.4, 5.5, 5.6, 5.7 e 5.8, verifica-se que o preditor proposto apresentou os menores valores para EQM, EQMN1 e EQMN2, ou seja, o melhor desempenho entre os preditores analisados. Desse modo, é possível observar que a aplicação do modelo *fuzzy* adaptativo na predição de tráfego conduz a ganhos expressivos de desempenho.

### 5.5.2 Análise do Desempenho do Algoritmo de Refinamento

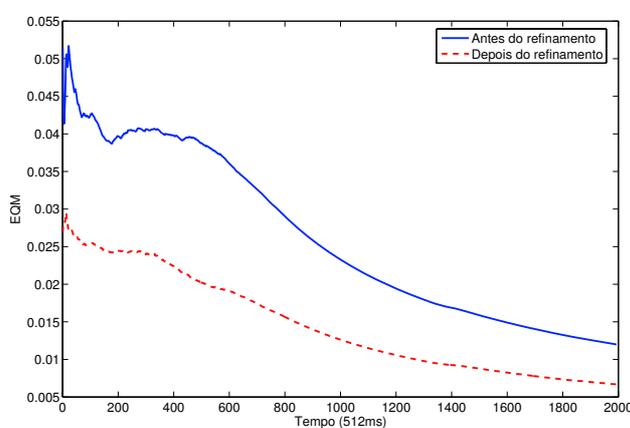
No Capítulo 4, mencionamos que o algoritmo de predição proposto é composto por duas etapas. Na primeira etapa, os parâmetros das partes antecedentes e consequentes do modelo *fuzzy* são calculados e assim, uma inicialização do modelo TSK é obtida. Na segunda etapa, um algoritmo de refinamento baseado no gradiente descendente é utilizado para fazer um ajuste fino do modelo inicial obtido melhorando a sua precisão.

Analisando o desempenho do algoritmo de refinamento, temos que as medidas de erro de predição utilizadas são bastante reduzidas após a aplicação desse algoritmo, conforme se pode observar pela Tabela 5.9.

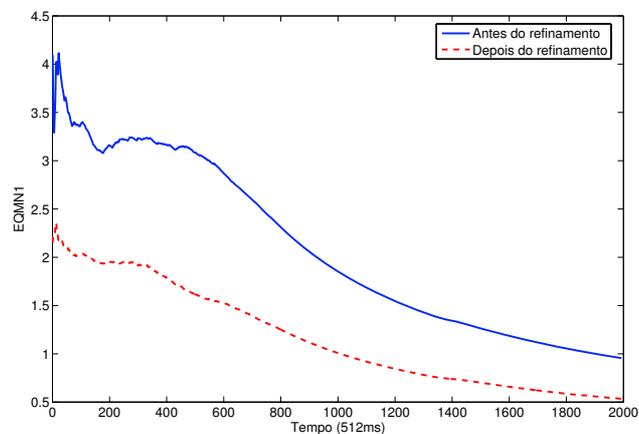
As Figuras 5.8 e 5.9 mostram as medidas de desempenho de predição (EQM, EQMN1 e EQMN2) para os traços de tráfego *dec-pkt-1* e *OctInt*, onde podemos observar, além da rápida convergência do algoritmo, a sua robustez no processo de refinamento proporcionando melhorias no desempenho.

Tab. 5.9: Comparação dos Resultados de Desempenho do Preditor Proposto Antes e Depois da Aplicação do Algoritmo de Refinamento .

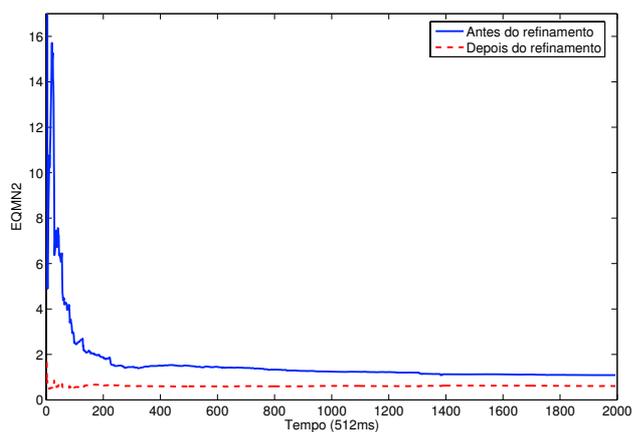
Traço	EQM		EQMN1		EQMN2	
	Antes	Depois	Antes	Depois	Antes	Depois
<i>dec-pkt-1</i>	0.0120	0.0067	0.9547	0.5340	1.0915	0.6106
<i>dec-pkt-2</i>	0.0112	0.0071	0.6515	0.4113	0.8746	0.5577
<i>OctExt</i>	0.0162	0.0058	1.5048	0.5385	2.5954	0.9283
<i>OctInt</i>	0.0436	0.0156	1.3179	0.4691	1.3401	0.4787
<i>FL-10-7-S-1</i>	0.0029	0.0025	2.0708	1.7997	0.9885	0.8601
<i>4-7-I-9</i>	0.0153	0.0066	0.7098	0.3045	0.2863	0.1227



(a) EQM



(b) EQMN1



(c) EQMN2

Fig. 5.8: Comparação das Medidas de Erro de Predição Antes e Depois da Aplicação do Algoritmo de Refinamento para o Traço *dec-pkt-1*.

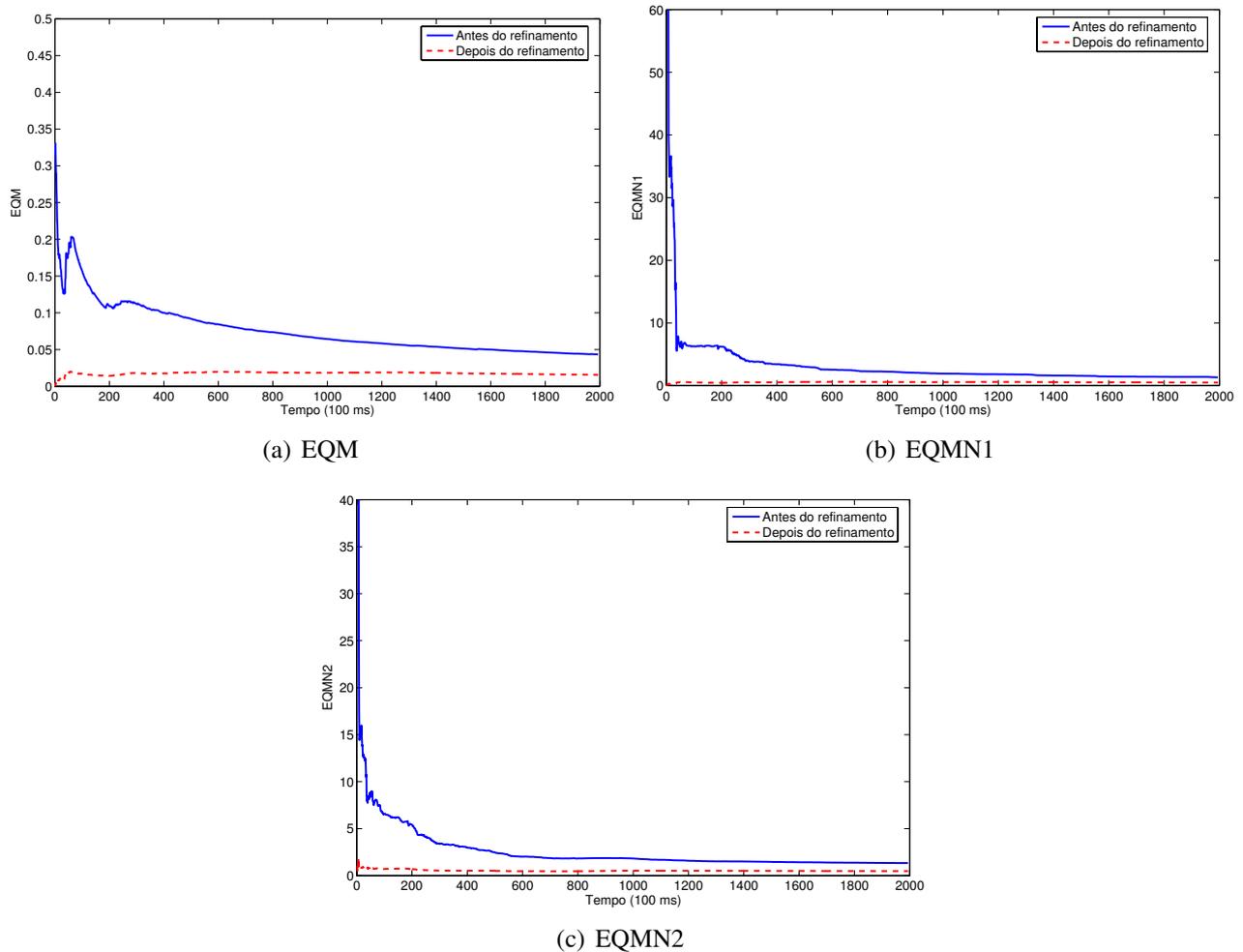


Fig. 5.9: Comparação das Medidas de Erro de Predição Antes e Depois da Aplicação do Algoritmo de Refinamento para o Traço *OctInt*.

### 5.5.3 Análise do Erro Quadrático Médio

Outro parâmetro útil para a verificação da eficiência de um preditor é a velocidade de convergência do seu algoritmo de predição, ou seja, analisar a rapidez com que as medidas de desempenho EQM, EQMN1 e EQMN2 diminuem e se estabilizam em um valor próximo a zero. As Figuras 5.10, 5.11 e 5.12 comparam as medidas de desempenho de predição (EQM, EQMN1, EQMN2) entre o preditor proposto e os outros preditores utilizados como comparação para os traços *dec-pkt-1*, *dec-pkt-2* e *OctInt*.

As Figuras 5.10, 5.11 e 5.12 mostram que o preditor proposto apresentou os menores valores de EQM, EQMN1 e EQMN2, em comparação aos outros preditores analisados. Com o uso de poucas amostras iniciais, o erro quadrático do preditor proposto decai e se mantém próximo a zero. Pode-se notar que os outros preditores analisados requerem um maior intervalo de tempo para atingir o mesmo

resultado.

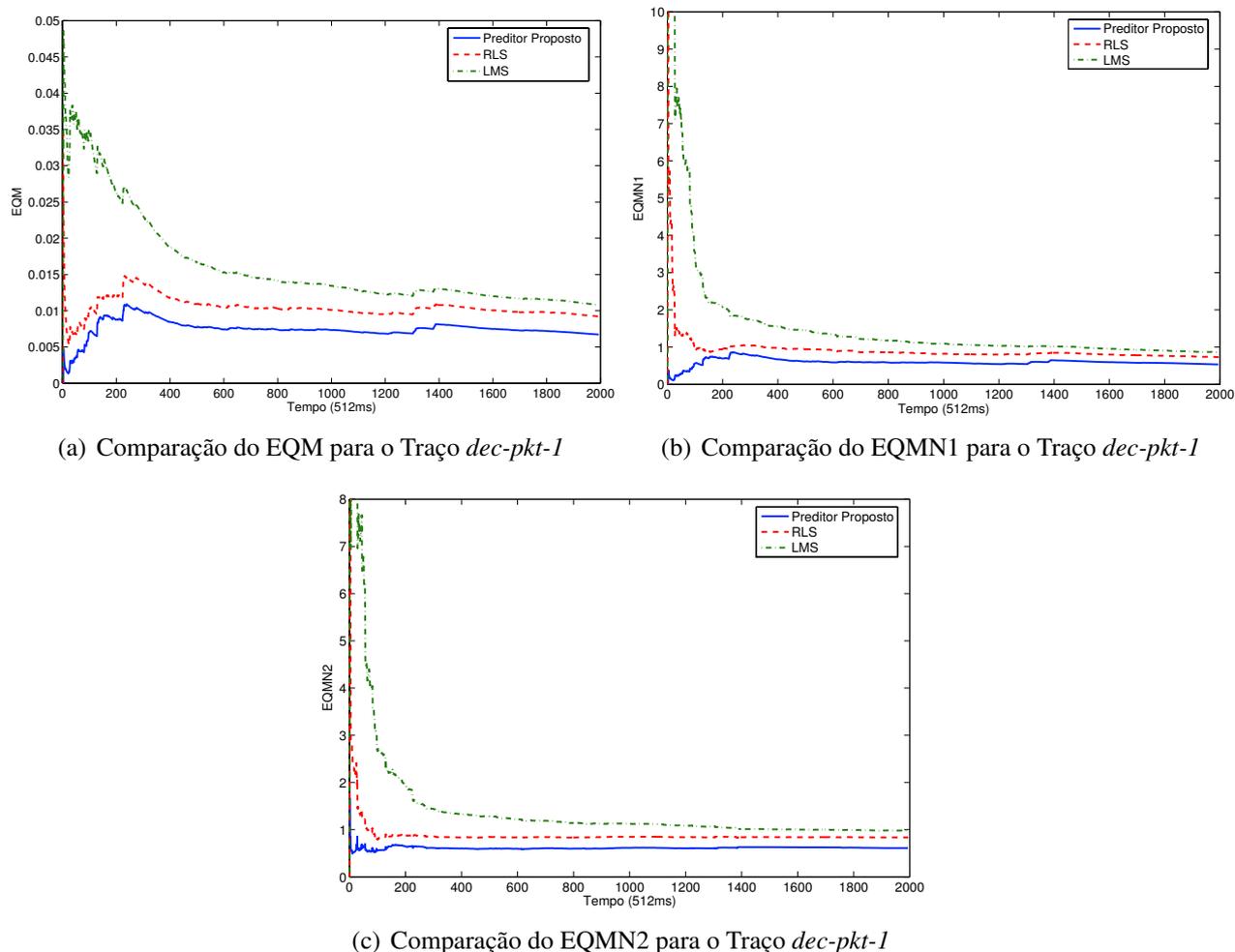


Fig. 5.10: Comparação das Medidas de Erro entre o Preditor Proposto e os Utilizados na Análise.

## 5.6 Considerações Finais

Neste capítulo, analisamos o desempenho de predição do modelo *fuzzy* TSK treinado com o algoritmo proposto ARFARC. Inicialmente, uma teoria de predição linear foi apresentada e os preditores utilizados para comparação foram detalhados. Em seguida, as medidas de desempenho utilizadas na análise da predição foram definidas. Para avaliar as características multifractais dos traços de tráfego utilizados na simulação, aplicou-se o diagrama de multiescala linear. Por fim, uma análise do preditor foi feita utilizando os traços de tráfego considerados monofractais e multifractais. Os resultados mostraram que o preditor *fuzzy* adaptativo proposto, quando comparado aos outros dois preditores,

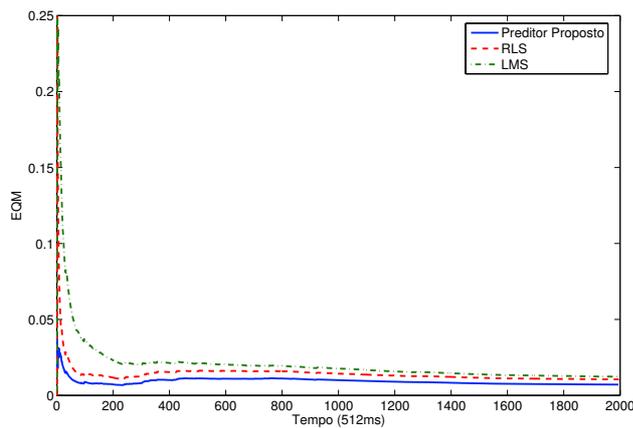
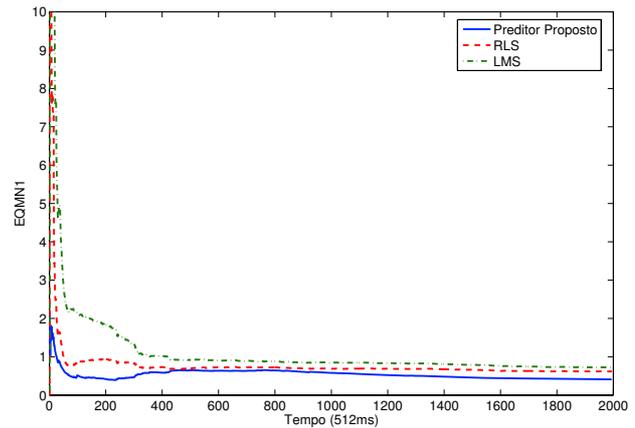
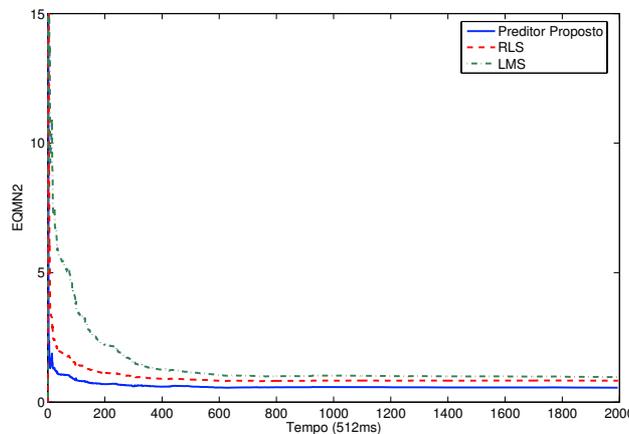
(a) Comparação do EQM para o Traço *dec-pkt-2*(b) Comparação do EQMN1 para o Traço *dec-pkt-2*(c) Comparação do EQMN2 para o Traço *dec-pkt-2*

Fig. 5.11: Comparação das Medidas de Erro entre o Preditor Proposto e os Utilizados na Análise.

é capaz de proporcionar ganhos de desempenho de predição e resultados expressivos, tanto para os traços de tráfego monofractais quanto para os traços de tráfego multifractais. Através da análise das medidas de desempenho de predição (EQM, EQMN1 e EQMN2) antes e depois da aplicação do algoritmo de refinamento, foi possível comprovar a alta eficiência desse algoritmo em reduzir as medidas de desempenho e proporcionar um grande incremento na precisão da modelagem. Além da comparação das medidas de desempenho de predição entre o preditor proposto e os outros preditores, foi comparado também a convergência do erro quadrático médio entre cada um dos preditores, onde foi possível observar que o preditor proposto obtém um decaimento mais íngreme do erro quadrático médio quando comparado aos outros preditores. Por fim, dentre todas as vantagens citadas, é conveniente frisar que o preditor proposto possui características adaptativas, ou seja, o ajuste dos parâmetros de modelagem é feito a cada nova amostra de tráfego disponibilizada. Essa característica permite que o preditor seja usado em aplicações de tempo real.

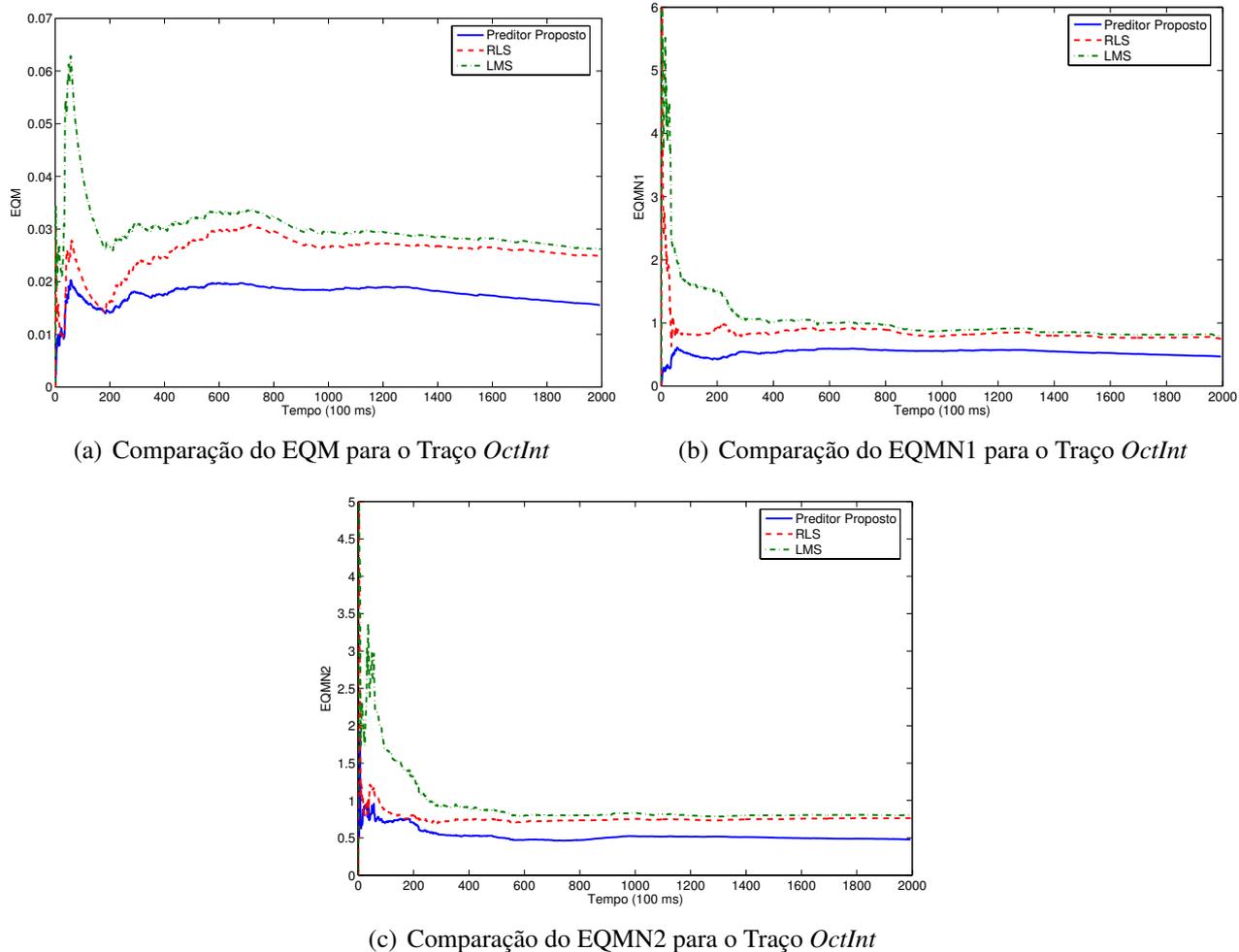


Fig. 5.12: Comparação das Medidas de Erro entre o Preditor Proposto e os Utilizados na Análise.



## Capítulo 6

# Análise do Esquema de Controle de Congestionamento Proposto

### 6.1 Introdução

Nesse capítulo, o esquema de controle de congestionamento proposto é avaliado e comparado a outros dois métodos de controle de congestionamento. Inicialmente, apresentamos os métodos de controle utilizados como comparação e as medidas de desempenho utilizadas. Em seguida, analisamos o esquema de controle de congestionamento proposto através de simulações na linguagem MatLab. A análise do esquema de controle proposto consiste de 4 etapas: na primeira etapa, considera-se o caso de um nó com duas fontes de tráfego: uma fonte com taxa controlável e outra não-controlável. Para tal cenário são analisados dois casos, um com a taxa média da fonte controlável maior do que a taxa média da fonte não-controlável, e o outro caso com a situação inversa. Na segunda etapa da análise, um cenário de rede composto por 3 nós e com 5 fontes de tráfego utilizando os traços *dec-pkt-2* e *dec-pkt-3* na escala de  $512ms$  é considerado. Na terceira etapa, avalia-se o caso de um nó com duas fontes de tráfego. Para especificar tais fontes são utilizadas as séries de tráfego monofractais fGn (*fractional Gaussian noise*). E na quarta etapa da análise do esquema de controle proposto, um cenário com 3 nós e 5 fontes de tráfego similar ao cenário da segunda etapa é avaliado. Os traços utilizados nesse cenário são os traços de tráfego multifractais da Petrobrás *3-7-I-1*, *4-7-I-1*, *4-7-I-9*, *4-7-I-5* e *4-7-I-2* com escala de agregação de  $100ms$ .

Uma comparação das medidas de desempenho da predição do tamanho da fila no *buffer* entre o preditor *fuzzy* proposto e os preditores RLS e LMS é realizada. E por fim, é verificado o desempenho do esquema de controle de congestionamento proposto variando-se o parâmetro de projeto  $\lambda$  no cálculo da taxa ótima para a fonte controlável.

## 6.2 Esquemas de Controle de Congestionamento Considerados como Comparação

Para avaliarmos o desempenho do esquema de controle de congestionamento proposto, comparamos algumas medidas de desempenho com outros métodos de controle de congestionamento conhecidos como método de Realimentação Binária e método de Controle Proporcional. Tais métodos foram escolhidos por serem bastante utilizados na literatura e por possuírem características semelhantes ao método proposto, isto é, esses métodos analisam o comportamento do tamanho da fila no *buffer* e estimam uma nova taxa para as fontes controláveis a partir de condições pré-estabelecidas particulares a cada método.

### 6.2.1 Realimentação Binária (*Binary Feedback*)

Nesse método de controle de congestionamento, a taxa da fonte de tráfego é controlada através do monitoramento do tamanho da fila no *buffer* [28]. Dois limiares são configurados:  $T_l$  e  $T_h$ . Quando o tamanho da fila no *buffer* excede  $T_h$ , o congestionamento é detectado e a taxa da fonte é reduzida por um fator de 0.98. Por outro lado, quando o tamanho da fila no *buffer* se torna menor do que  $T_l$ , considera-se que não há mais congestionamento e a taxa da fonte é incrementada por um fator correspondente a 1% da capacidade do enlace de saída do nó. Caso contrário, a taxa da fonte permanece inalterada.

Nas simulações, os dois limiares  $T_l$  e  $T_h$  são configurados como  $0.95b^\tau$  e  $1.05b^\tau$ , respectivamente, onde  $b^\tau$  é o nível desejado para o tamanho da fila no *buffer*. A taxa inicial da fonte é igual a capacidade do enlace menos a média da taxa da fonte não-controlável. São aplicados também níveis de saturação para a taxa da fonte, ou seja, um limite inferior igual a 0 e um limite superior igual a capacidade do enlace.

### 6.2.2 Controle de Congestionamento Proporcional (*Proportional Congestion Control*)

Esse método de controle de congestionamento monitora o tamanho da fila no *buffer* e usa esse valor monitorado de uma forma proporcional através de uma variável de controle  $c(k)$  para regular a taxa da fonte controlável [65].

Seja  $b^\tau$  o valor de referência de controle do tamanho da fila no *buffer* em um enlace. O sinal de controle da taxa de tráfego  $c(k)$  é gerado através do seguinte algoritmo de controle de congestionamento

mento:

$$c(k) = \begin{cases} 1.002 & \text{se } b(k) < 0.998b^\tau \\ 1.002 - \frac{b(k) - 0.998b^\tau}{b^\tau} & \text{se } 0.998b^\tau < b(k) < 1.002b^\tau \\ 0.998 & \text{se } b(k) > 1.002b^\tau \end{cases}$$

onde 0.998 pode ser visto como um limiar inferior e 1.002 como um limiar superior para o sinal de controle. A Figura 6.1 mostra a variação do sinal de controle  $c(k)$ , utilizado nesse método, em função do tamanho da fila no *buffer*.

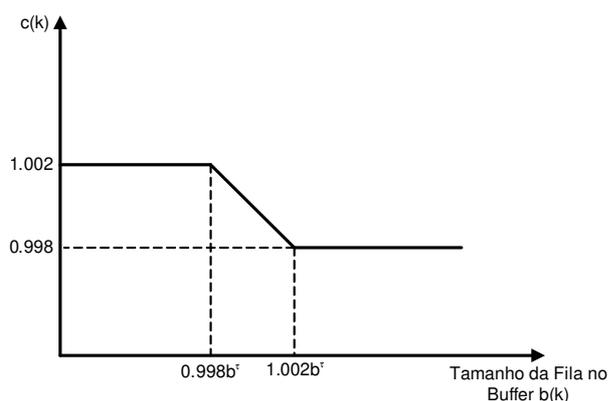


Fig. 6.1: Sinal de Controle  $c(k)$  do Controle de Congestionamento Proporcional.

O sinal de controle da taxa  $c(k)$  é inversamente proporcional a ocupação do *buffer* quando  $0.998b^\tau < b(k) < 1.002b^\tau$ . Quando  $b(k)$  é maior do que  $b^\tau$ , o sinal de controle  $c(k)$  é enviado para fonte para que esta diminua sua taxa. A situação inversa também ocorre, ou seja, quando  $b(k)$  é menor do que  $b^\tau$ , o sinal de controle  $c(k)$  é enviado para que a fonte aumente sua taxa. Sendo  $\mu(k)$  a taxa da fonte controlável no instante  $k$ , então a taxa da fonte para o instante seguinte será dada por  $\mu(k+1) = c(k)\mu(k)$ .

Outro método de controle de congestionamento que têm sido utilizado na literatura é o RED (*Random Early Detection*). Tal método tem comportamento pró-ativo, ou seja, visa evitar o congestionamento gerenciando o tamanho da fila no *buffer* ativamente [20]. O RED se baseia especificamente nos mecanismos de controle de congestionamento TCP/IP. Ou seja, para se avaliar o desempenho do RED, o ideal é reproduzir mais precisamente possível os mecanismos de controle do protocolo TCP/IP. Já, para se implementar o esquema de controle de fluxos proposto em uma rede, é necessário apenas que a rede tenha mecanismos de realimentação para controlar a taxa dos fluxos das fontes controláveis. Portanto, o desempenho do esquema proposto pode ser avaliado em simulações com filas sob a disciplina FIFO que não envolvam especificamente os mecanismos de controle do TCP/IP. Por esse motivo, o método de controle RED não é considerado um método que pode ser comparado ao esquema de controle de congestionamento proposto.

### 6.3 Medidas de Desempenho

Para avaliar o desempenho do controle de congestionamento proposto, três medidas de desempenho são utilizadas: TPB (Taxa de Perda de Bytes), utilização do enlace de saída e a vazão da taxa controlável.

A TPB é a razão entre o número de bytes perdidos e o número total de bytes transmitidos durante o período de simulação. Ou seja,

$$TPB(\%) = \frac{\text{No de Bytes Perdidos}}{\text{No Total de Bytes Transmitidos}} \times 100\% \quad (6.1)$$

A utilização do enlace é a proporção da capacidade do enlace de saída que é utilizada na transmissão dos fluxos de tráfego. É dada pela seguinte equação:

$$\text{Utilização}(\%) = \frac{\text{Tráfego Total Transmitido}}{\text{Capacidade do Enlace}} \times 100\% \quad (6.2)$$

A vazão do tráfego controlável é a quantidade de dados [bytes], oriundo das fontes controláveis, que são processados em um determinado período de tempo.

$$\text{Vazão}(\text{bytes}/\text{seg}) = \frac{\text{No de Bytes Transmitidos Vindos da Fonte Controlável}}{\text{Intervalo de Tempo de Observação}} \quad (6.3)$$

O objetivo do controle proposto é controlar a taxa dos fluxos controláveis de modo que se obtenha baixas perdas e alta utilização do enlace. No entanto, esses dois requisitos são conflitante no sentido que priorizando o desempenho de um desses parâmetros (ex.: perdas nula), pode-se degradar o outro (redução na utilização). Desse modo, deve-se procurar equilibrar esses dois fatores para obter um melhor desempenho da rede.

### 6.4 Etapa 1 da Análise: Avaliação em 1 Nó

Nessa primeira etapa da análise serão considerados dois casos, um com a taxa média da fonte controlável maior do que a taxa média da fonte não-controlável, e o outro caso com a situação inversa.

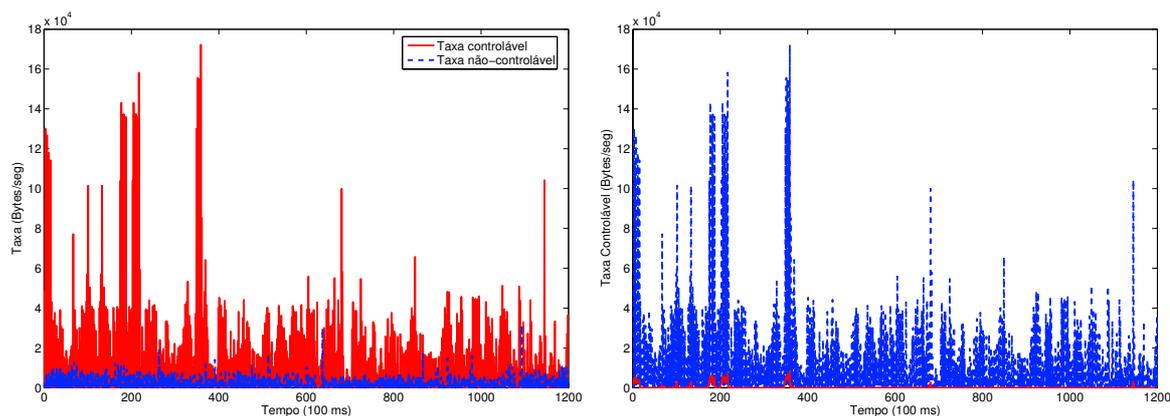
#### 6.4.1 Caso 1 - Taxa Controlável > Taxa Não-Controlável

Os traços da Petrobrás 4-7-I-5 e 3-7-I-1 foram utilizados para especificar, respectivamente, o tráfego controlável e não-controlável. Com o intuito de analisar o esquema de controle proposto em um ambiente com perdas, a partir das taxas média e de pico dos traços de tráfego utilizados

(Tabela 5.1), algumas características do sistema são especificadas, tais como: o tamanho do *buffer* (capacidade máxima)  $B_{max}$  foi fixado em  $6 \times 10^4$  bytes, a capacidade do enlace  $\eta$  foi escolhida como 13 kbytes/seg e o nível desejado do tamanho da fila no *buffer* é estabelecido como  $b^r = 30\%B_{max}$ , ou seja,  $b^r = 1.8 \times 10^4$  bytes. O parâmetro  $\lambda$  do cálculo da taxa ótima de controle de fluxo é fixado em 1. Esse valor do parâmetro  $\lambda$  é escolhido de tal forma a proporcionar melhores resultados de desempenho no sistema. Uma análise da influência da variação desse parâmetro nas medidas de desempenho é realizada no fim deste capítulo.

Nesse caso 1, como a taxa média dos fluxos de tráfego controlável é maior do que a taxa média dos fluxos de tráfego não-controlável, o enlace é em maior parte ocupado pelos fluxos de tráfego controlável. Após a aplicação do controle de congestionamento proposto, a taxa controlável será reduzida e a quantidade de dados que entram no nó diminuirão consideravelmente, logo o *buffer* ficará ocioso. Deste modo, o comportamento esperado para o cenário considerado após a aplicação do controle de congestionamento proposto, é que a utilização do enlace diminua devido à redução da taxa controlável.

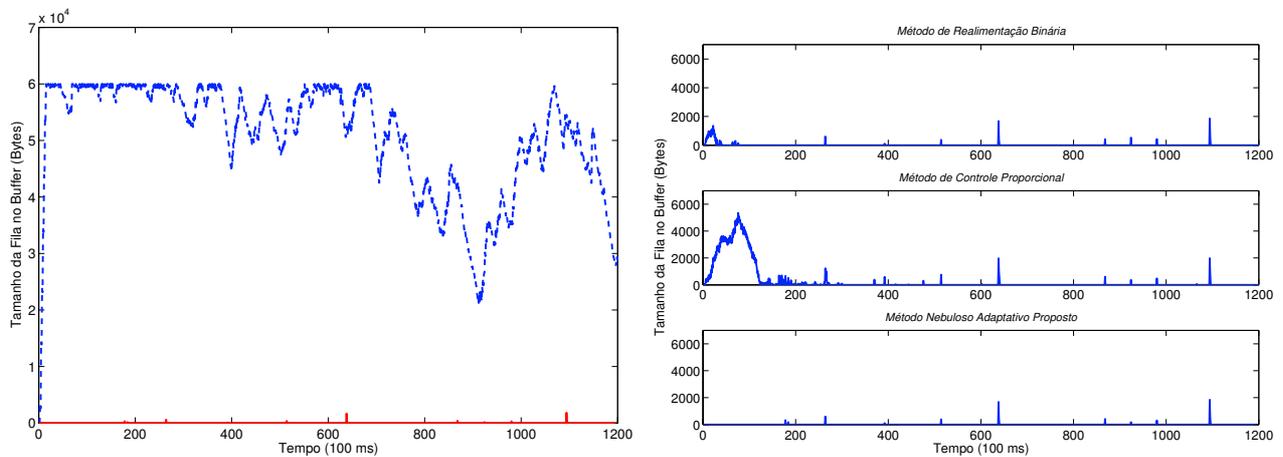
Na Figura 6.2.a, são mostradas as taxas dos fluxos de tráfego controlável e não-controlável que entram no nó considerado. E na Figura 6.2.b é mostrada a taxa do fluxo controlável antes e depois da aplicação do controle proposto. Como se pode verificar na Figura 6.2.a, a taxa do fluxo controlável antes da aplicação do controle é, em média, maior do que a taxa do fluxo não-controlável. Pode-se notar também que, após o controle, a taxa controlável é consideravelmente reduzida, e em alguns instantes até levada a um valor nulo, em favor da taxa não-controlável, visto que os fluxos de tráfego das fontes não-controláveis são prioritários.



(a) Taxas dos Fluxos Controlável e Não-controlável (b) Taxa do Fluxo Controlável Antes (Linha Tracejada) e Depois (Linha Cheia) da Aplicação do Controle Proposto.

Fig. 6.2: Taxas dos Fluxos Controlável e Não-controlável .

A Figura 6.3.a mostra o tamanho da fila no *buffer* antes e depois da aplicação do controle proposto. Nota-se que o tamanho da fila foi bastante reduzido e, em quase todos os instantes, o *buffer* está vazio confirmando a análise feita inicialmente. Uma comparação do tamanho da fila no *buffer* resultante da aplicação do controle proposto e dos outros métodos comparativos é mostrada na Figura 6.3.b, onde é possível observar que o *buffer* com fila mais comportada é o que foi obtido pelo método proposto. Na Tabela 6.1, os resultados de desempenho do método proposto e dos métodos comparativos são resumidos, onde podemos observar que todos os métodos alcançaram taxa de perda zero, porém o método de controle proposto foi o que melhor reduziu o tamanho da fila no *buffer*, o qual pode ser observado pela menor variância do tamanho da fila. A utilização do enlace, no entanto, diminuiu um pouco no método proposto, como havia sido suposto. Nos outros métodos a utilização permaneceu inalterada.



(a) Tamanho da Fila no *Buffer* Antes (Linha Tracejada) e Depois (Linha Cheia) da Aplicação do Controle  
(b) Comparação do Tamanho da Fila no *Buffer* Resultante do Controle de Congestionamento Proposto e dos Outros Métodos Comparativos.

Fig. 6.3: Análise do Tamanho da Fila no *Buffer*.

Tab. 6.1: Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento para o Caso 1.

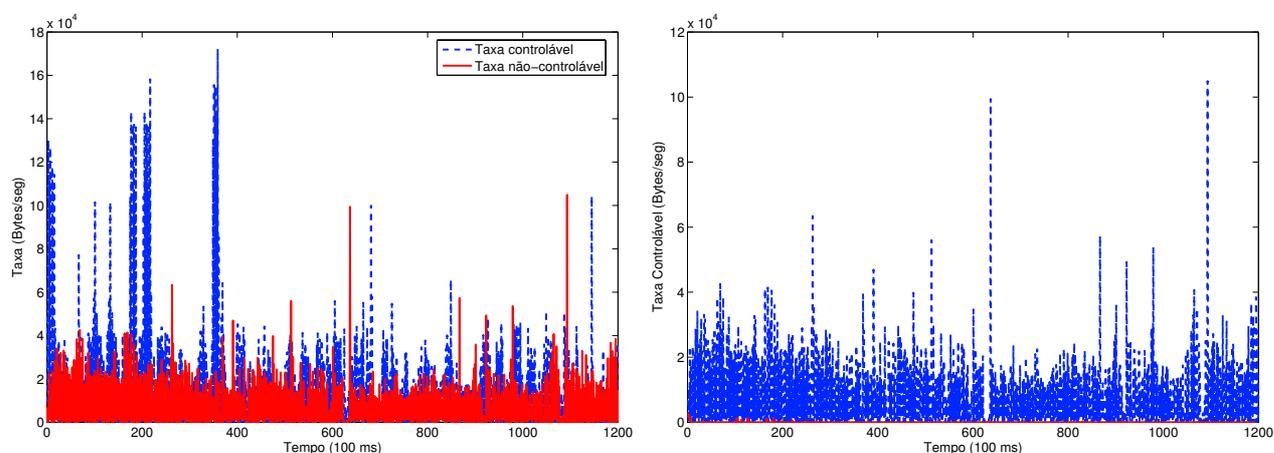
Parâmetro	Sem Controle	Realimentação Binária	Controle Proporcional	Método Proposto
Média da Fila	$5.06 \times 10^4$	21.54	291.73	5.83
Variância da Fila	$9.94 \times 10^7$	$1.85 \times 10^4$	$8.64 \times 10^5$	$6.38 \times 10^3$
TPB	22.49%	0.00%	0.00%	0.00%
Utilização	99.91%	99.91%	99.91%	89.68%
Vazão	$6.03 \times 10^3$	720.17	$4.06 \times 10^3$	230.10

### 6.4.2 Caso 2 - Taxa Controlável < Taxa Não-Controlável

Os traços da Petrobrás 3-7-I-2 e 4-7-I-5 foram utilizados para especificar, respectivamente, o tráfego controlável e não-controlável. Para gerar um ambiente com perdas, o tamanho do *buffer*  $B_{max}$  foi fixado em  $8 \times 10^4$  bytes, a capacidade do enlace  $\eta$  foi escolhida como 17 kbytes/seg o nível desejado do tamanho da fila no *buffer* é estabelecido como  $b^r = 30\%B_{max}$ , ou seja,  $b^r = 2.4 \times 10^4$  bytes. O parâmetro  $\lambda$  do cálculo da taxa de controle de fluxo ótima é fixado em 1.

Nesse caso, como a média da taxa não-controlável é maior do que a média da taxa controlável, após a aplicação do controle de congestionamento proposto a taxa controlável será reduzida. Contudo, como o enlace é em maior parte ocupado pelos fluxos de taxa não-controlável, o tamanho da fila no *buffer* continuará alto e perdas poderão ocorrer. Logo, após a aplicação do controle, é esperado que os dados que entram no nó diminuam ligeiramente e um alto valor para o tamanho da fila no *buffer* seja verificado.

Na Figura 6.4.a são mostradas as taxas dos fluxos controlável e não controlável antes da aplicação do controle proposto. Como pode-se verificar, nesse caso, a média da taxa não-controlável é maior do que a média da taxa controlável. Na mesma Figura 6.4.b, é mostrada a taxa do fluxo controlável antes e depois da aplicação do controle proposto, e é possível observar que a taxa é reduzida e em muitos instante levada a um valor nulo.



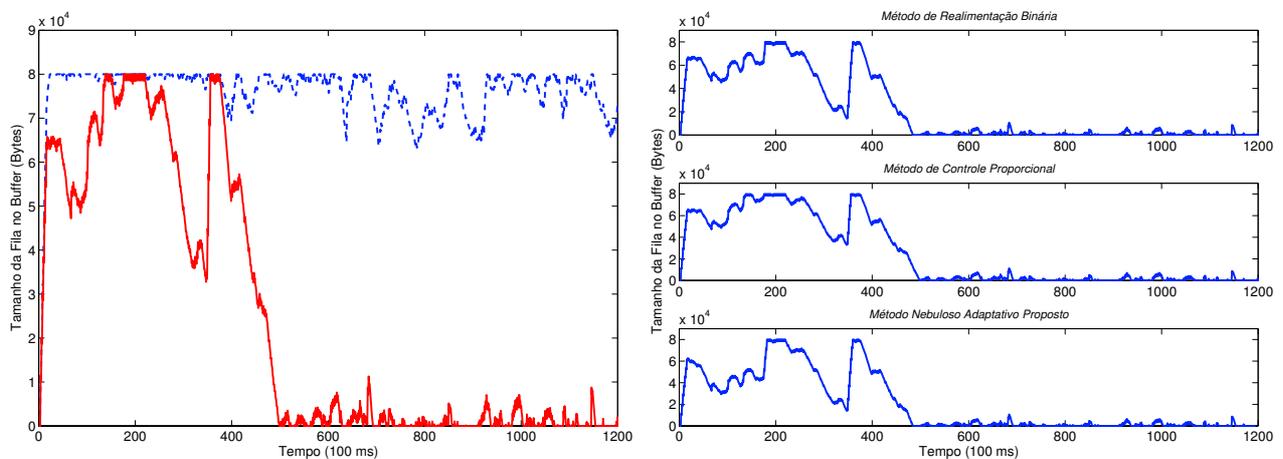
(a) Taxas dos Fluxos Controlável e Não-controlável

(b) Taxa do Fluxo Controlável Antes (Linha Tracejada) e Depois (Linha Cheia) da Aplicação do Controle Proposto.

Fig. 6.4: Taxas dos Fluxos Controlável e Não-controlável.

A Figura 6.5.a mostra o comportamento do tamanho da fila no *buffer* antes e depois da aplicação do controle proposto. Nota-se que o tamanho da fila foi reduzido, mas em alguns instantes ele ainda ultrapassa o tamanho do *buffer*  $B_{max} = 8 \times 10^4$  bytes. Esse comportamento confirma o que havia sido suposto inicialmente. Mesmo controlando e reduzindo a taxa do fluxo controlável, os

dados que entram não diminuem o suficiente para que o tamanho da fila no *buffer* fique abaixo do tamanho do *buffer*  $B_{max}$ , deste modo, perdas ainda acontecem. Na Figura 6.5.b temos uma comparação do tamanho da fila no *buffer* resultante da aplicação do método proposto e dos outros métodos comparativos. Podemos observar da Figura 6.5.b que os comportamentos das filas no *buffer* ficaram semelhantes, porém o método proposto ainda levou vantagem em alguns aspectos como se pode verificar pela Tabela 6.2. Dentre todos os métodos, o método proposto foi o que conseguiu menor taxa de perda de bytes. A variância do tamanho da fila no *buffer* aumentou em todos os métodos, porém o método proposto obteve o menor aumento. Da Tabela, é possível perceber também que a utilização pouco foi reduzida após a aplicação do método proposto.



(a) Tamanho da Fila no *Buffer* Antes (Linha Tracejada) e (b) Comparação do Tamanho da Fila no *Buffer* Resultante do Controle de Congestionamento Proposto e dos Outros Métodos Comparativos.

Fig. 6.5: Análise do Tamanho da Fila no *Buffer*.

Tab. 6.2: Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento para o Caso 2.

Parâmetro	Sem Controle	Realimentação Binária	Controle Proporcional	Método Proposto
Média da Fila	$7.61 \times 10^4$	$2.16 \times 10^4$	$2.43 \times 10^4$	$1.97 \times 10^4$
Variância da Fila	$5.11 \times 10^7$	$8.10 \times 10^8$	$9.23 \times 10^8$	$6.85 \times 10^8$
TPB	24.22%	8.38%	10.50%	7.44%
Utilização	99.86%	99.91%	99.91%	99.84%
Vazão	$7.10 \times 10^3$	172.71	985.96	17.57

## 6.5 Etapa 2 da Análise: Avaliação em 3 Nós - Traços de Tráfego *dec-pkt-2* e *dec-pkt-3*

### 6.5.1 Cenário de Simulação

Para avaliar o esquema de controle proposto, considera-se um cenário de rede composto por 3 nós e vários tráfegos de entrada (Figura 6.6). As Fontes 1 e 3 são fontes de tráfego não-controlável, ou seja, de maior prioridade. As Fontes 2, 4 e 5 são fontes de tráfego controlável, ou seja, que devem adaptar a sua taxa de acordo com a capacidade disponível no enlace de saída.

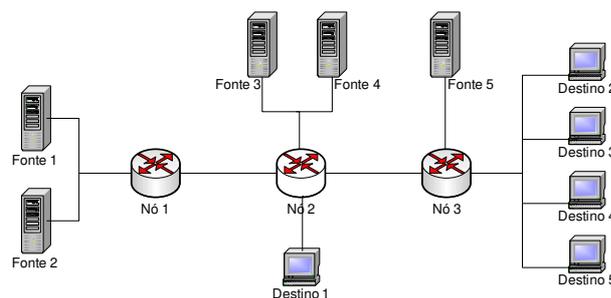


Fig. 6.6: Topologia de Rede Proposta para Análise.

Para especificar os fluxos de tráfego de entrada, são considerados os traços TCP/IP *dec-pkt-2* e *dec-pkt-3* com escala de agregação de  $512ms$ . Nesta escala, conforme comentado na seção 5.4.4, as séries *dec-pkt-2* e *dec-pkt-3* apresentam, respectivamente comportamento monofractal e multifractal. As especificações detalhadas dos componentes do cenário da Figura 6.6 são dadas a seguir:

**Fonte 1:** Tráfego não-controlável *dec-pkt-2* (amostras 501-2501).

**Fonte 2:** Tráfego controlável *dec-pkt-3* (amostras 501-2501).

**Nó 1:** Capacidade do Enlace:  $210k$  bytes/seg,  $d = 1$  intervalo de tempo ( $512ms$ ).

**Fonte 3:** Tráfego não-controlável *dec-pkt-2* (amostras 3001-5001).

**Fonte 4:** Tráfego controlável *dec-pkt-3* (amostras 1-2001).

**Destino 1:** Saída do Tráfego da Fonte 1.

**Nó 2:** Capacidade do Enlace:  $200k$  bytes/seg,  $d = 2$  intervalos de tempo ( $2 \times 512ms$ ).

**Fonte 5:** Tráfego Controlável *dec-pkt-3* (amostras 3001-5001).

**Nó 3:** Capacidade do Enlace:  $200k$  bytes/seg,  $d = 1$  intervalo de tempo ( $512ms$ ).

Destinos 2,3,4 e 5: Saída do Tráfego das Fontes 2, 3, 4 e 5 respectivamente.

## 6.5.2 Desempenho do Esquema de Controle de Congestionamento Proposto

Para a simulação do esquema de controle de congestionamento proposto neste cenário, devemos estipular uma configuração para a rede considerada de tal forma a obtermos um ambiente com perdas. O tamanho do *buffer* dos roteadores é fixado em  $B_{max} = 1.5 \times 10^6$  bytes e o nível desejado do tamanho da fila no *buffer* é estabelecido como  $b^r = 30\% B_{max}$ , ou seja,  $b^r = 4.5 \times 10^5$  bytes. O parâmetro  $\lambda$  do cálculo da taxa ótima de controle de fluxo é fixado em 1 e o algoritmo de controle é inserido em cada nó do cenário considerado.

Na Figura 6.7 as taxas dos fluxos controlável e não-controlável são mostradas para cada nó do cenário considerado. Podemos perceber que as duas taxas estão, em média, na mesma ordem com exceção do nó 3, onde em tal nó a média da taxa não-controlável ultrapassa ligeiramente a média da taxa controlável. Logo, para este nó é esperado um comportamento semelhante ao caso 2 da etapa 1 da análise do esquema de controle proposto.

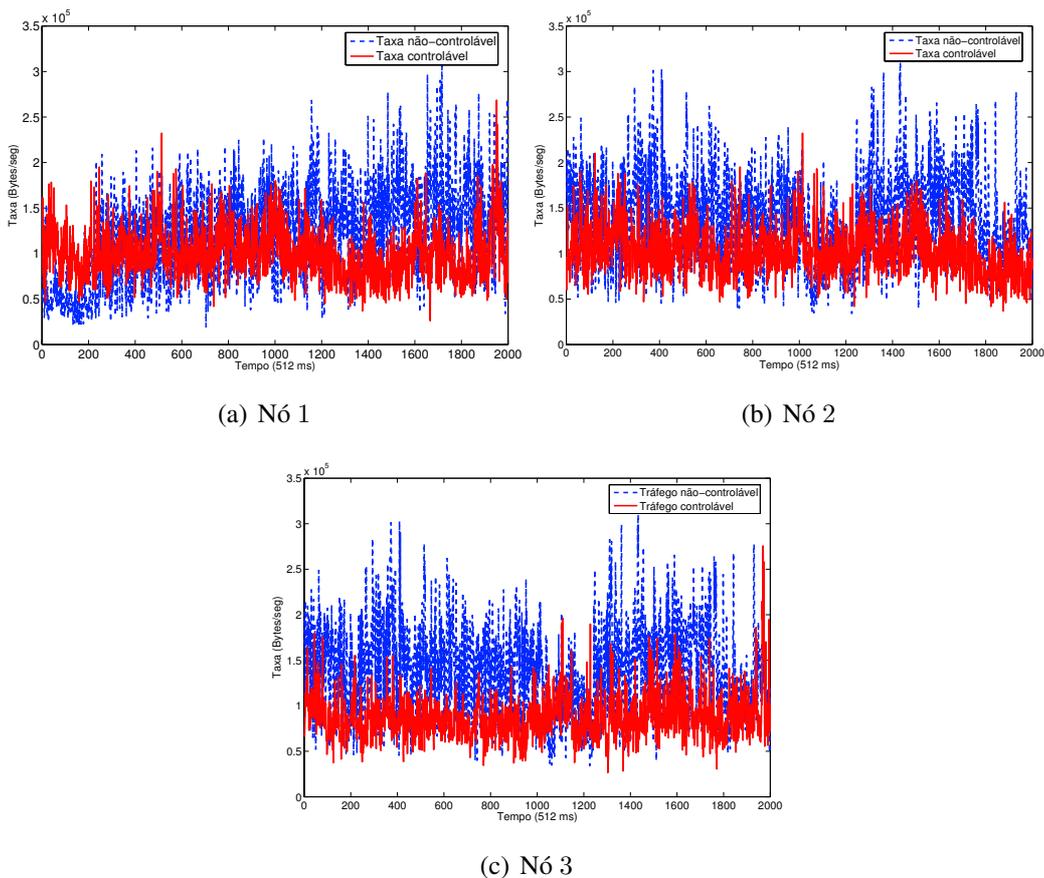


Fig. 6.7: Taxas dos Fluxos Controlável e Não-Controlável para Cada Nó do Cenário.

Na Figura 6.8, é mostrado o comportamento da fila no *buffer* nos Nós 1, 2 e 3 antes e depois da aplicação do esquema de controle de congestionamento. Como se pode observar, a fila no *buffer* se tornou mais comportada e abaixo do nível desejado  $b^T$ , o qual é igual a  $4.5 \times 10^5$  bytes, exceto para o nó 3, no qual o tamanho da fila no *buffer* num determinado instante ultrapassa o nível desejado, de acordo com o que já havia sido suposto. Da Figura 6.8 é possível perceber que em nenhum instante o tamanho da fila no *buffer*, após o controle, se tornou maior do que o tamanho do *buffer*  $B_{max}$ , logo a taxa de perda de bytes (TPB) é igual a 0% para todos os nós.

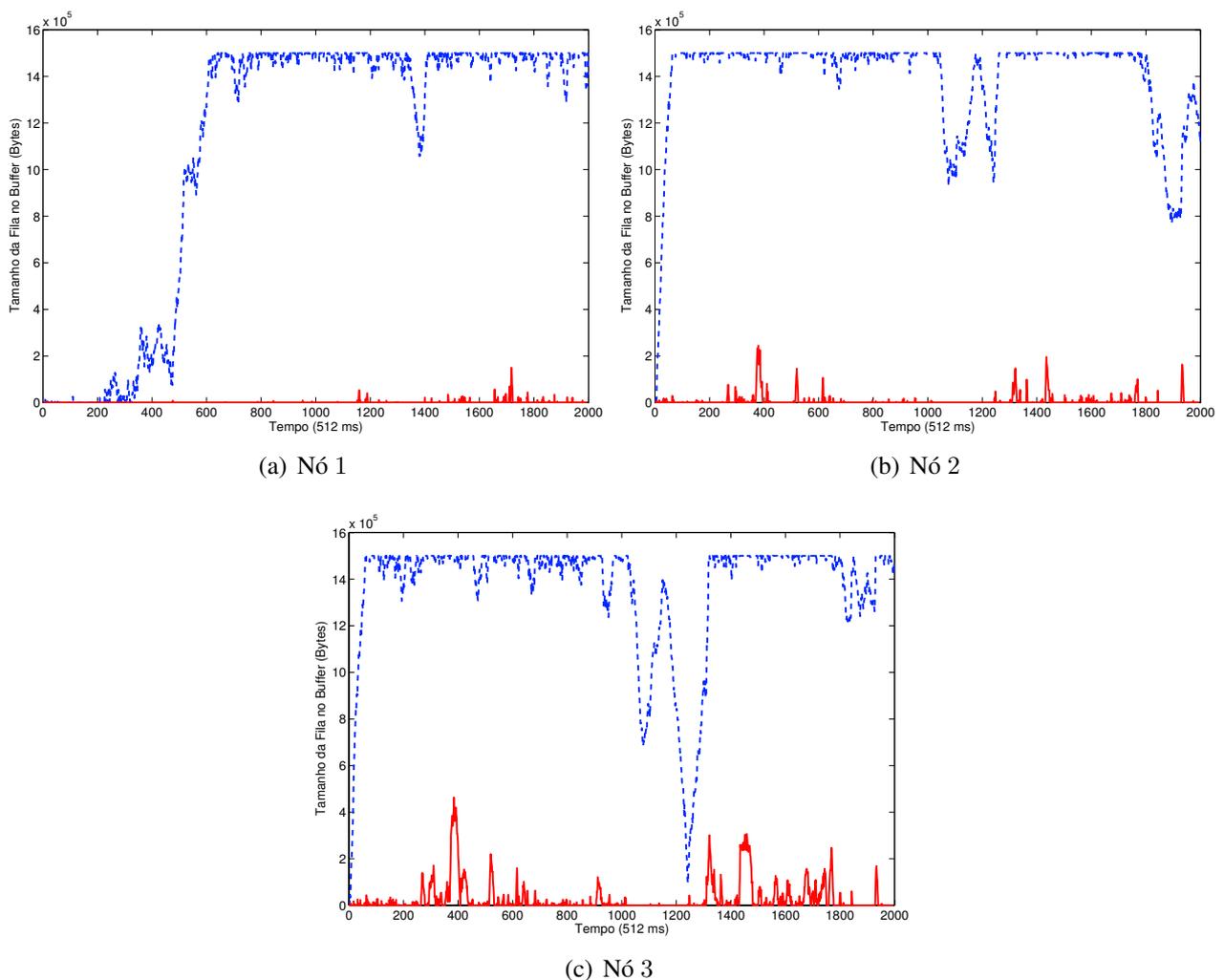


Fig. 6.8: Tamanho da Fila no *Buffer* Antes e Depois da Aplicação do Controle de Congestionamento Proposto para Cada Nó do Cenário. Antes (Linha Tracejada), Depois (Linha Cheia).

A Figura 6.9 apresenta uma comparação da taxa da fonte controlável antes e depois da aplicação do controle de congestionamento proposto. Através da Figura 6.9, é possível observar que a taxa controlável foi reduzida e algumas vezes até mesmo levada a um valor nulo. Como foi já dito ante-

riormente, a fonte controlável usa a banda que está disponível, ou seja, a taxa de tal fonte é igual a taxa correspondente a banda que não está sendo usada pelas fontes prioritárias. Logo, é possível que em determinados instantes, o tráfego dessas fontes controláveis seja todo vetado em favor das fontes não-controláveis.

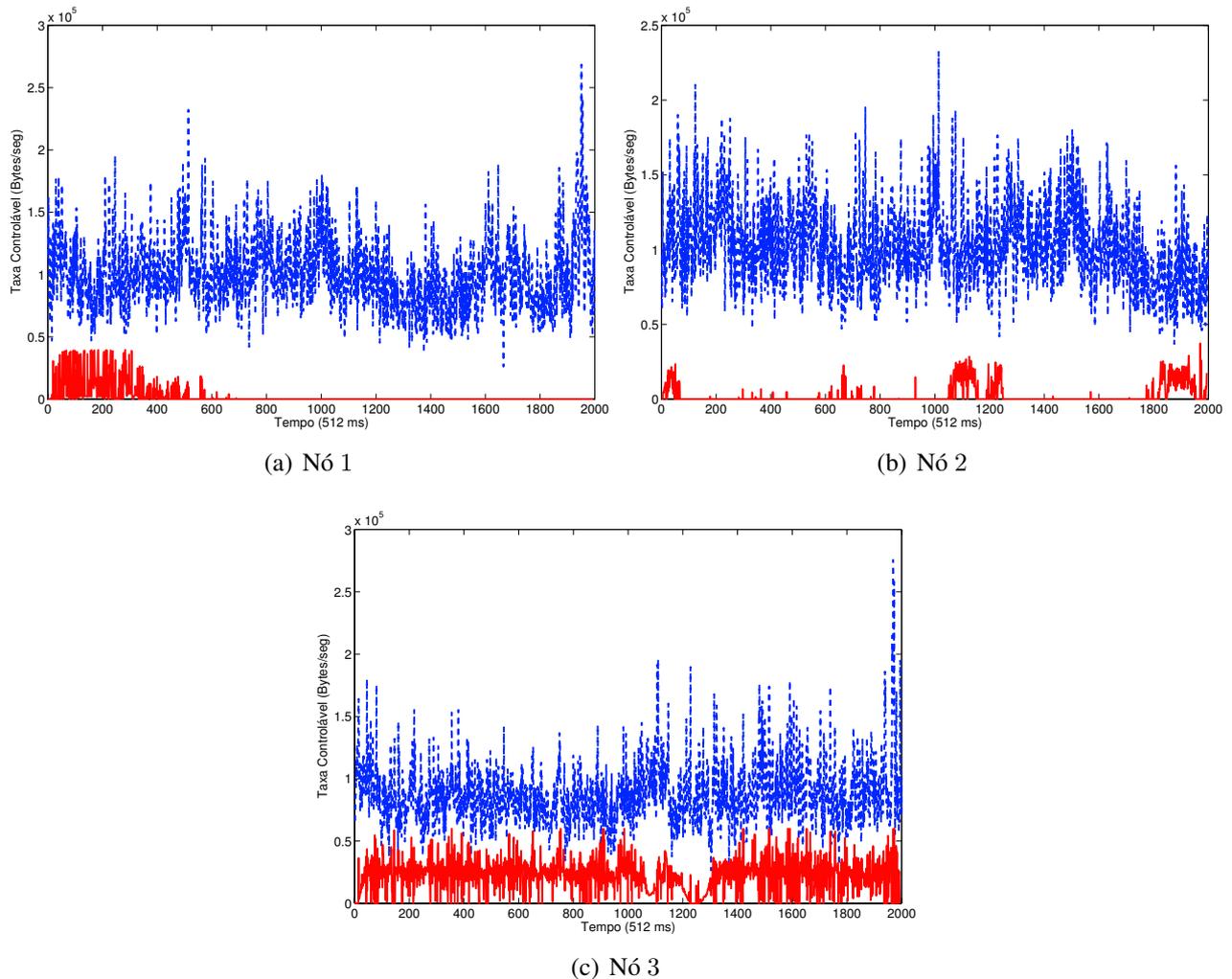


Fig. 6.9: Taxa da Fonte Controlável Antes e Depois da Aplicação do Controle de Congestionamento Proposto para Cada Nó do Cenário. Antes (Linha Tracejada), Depois (Linha Cheia).

Comparamos o esquema de controle de congestionamento proposto com os métodos mencionados na seção 6.2. A Figura 6.10 mostra o tamanho da fila no *buffer*, onde para cada figura referente a cada nó, temos respectivamente o método de Realimentação Binária, o método de Controle Proporcional e, por fim, o método proposto. Através da Figura 6.10, podemos observar que o método proposto obtém um processo de fila no *buffer* bem mais comportado e abaixo do nível desejado.

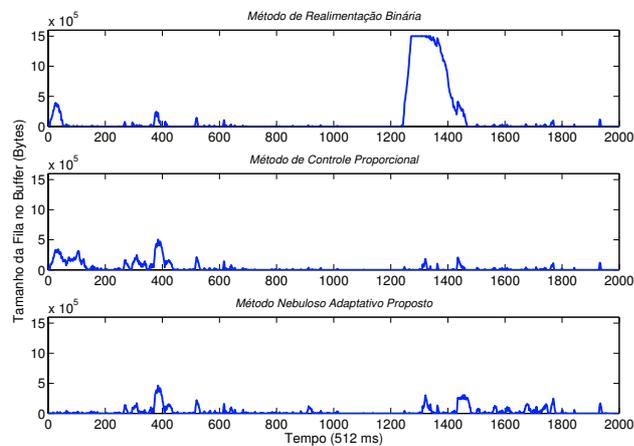
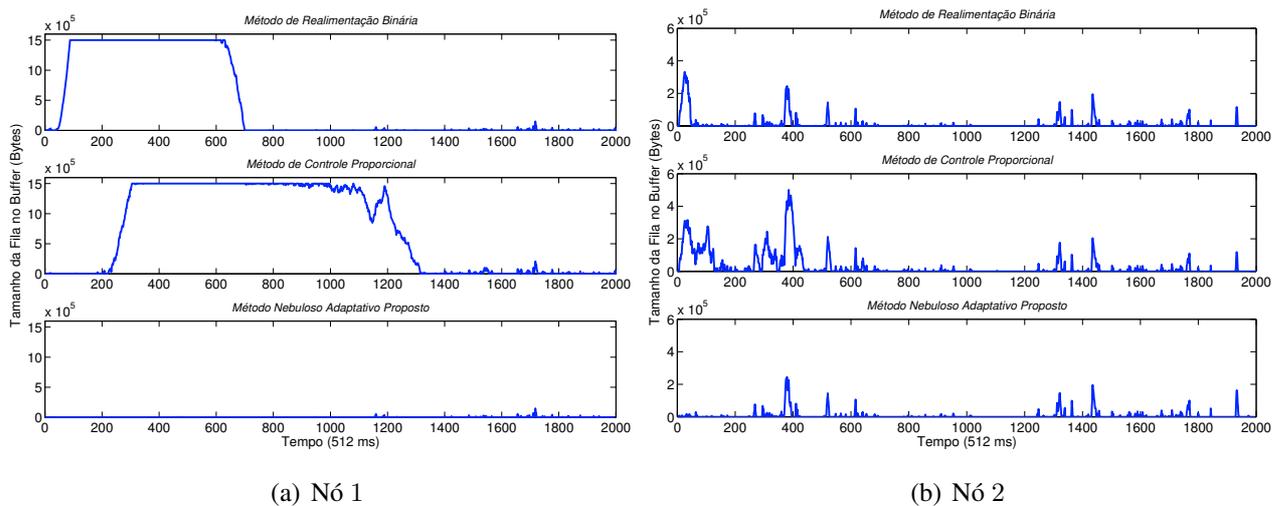


Fig. 6.10: Comparação do Tamanho da Fila no *Buffer* Resultante do Controle de Congestionamento Proposto e dos Outros Métodos Comparativos para Cada Nó do Cenário.

No nó 1, os tamanhos da fila no *buffer* obtidos com a aplicação dos dois métodos de comparação tiveram que ser limitados no tamanho do *buffer*  $B_{max} = 1.5 \times 10^6$  bytes, indicando a ocorrência de perdas no *buffer*. Essas perdas acontecem pois os métodos de comparação respondem, aumentando ou diminuindo sua taxa, de acordo com o tamanho da fila no *buffer*. Como se pode perceber, o tamanho da fila no Nó1 (Figura 6.8.a) inicia nulo e assim permanece até as 200 primeiras amostras. Assim, como mostra a Figura 6.11, os métodos de comparação, que aumentam a taxa quando o tamanho da fila no *buffer* está pequeno e vice-versa, incrementam a taxa controlável e só começam a diminuí-la quando o tamanho da fila chega a um valor razoável (aproximadamente no instante 400). As perdas no *buffer* ocorrem devido à lentidão desses algoritmos em diminuir a taxa controlável e, conseqüentemente, o tamanho da fila no *buffer*. Essa lentidão ocorre principalmente no método de

Controle Proporcional, conforme se pode observar na Figura 6.11, e tal método é o que obteve perdas durante um intervalo maior de tempo. Esse comportamento não ocorre com o método proposto, como podemos ver pela Figura 6.9, a taxa é controlada dinamicamente e assim não ocorrem perdas, ou seja, temos TPB igual a 0%. Na Tabela 6.3, os resultados de desempenho obtidos são apresentados, onde podemos observar que nos método de comparação, a variância da fila no *buffer* aumentou, o que não ocorre com o método proposto. Entretanto, a utilização do enlace diminuiu um pouco após a aplicação do método proposto, fato este explicado pela grande capacidade do algoritmo em controlar a taxa controlável, eliminando as perdas em consequência de um pequeno decremento na utilização do enlace.

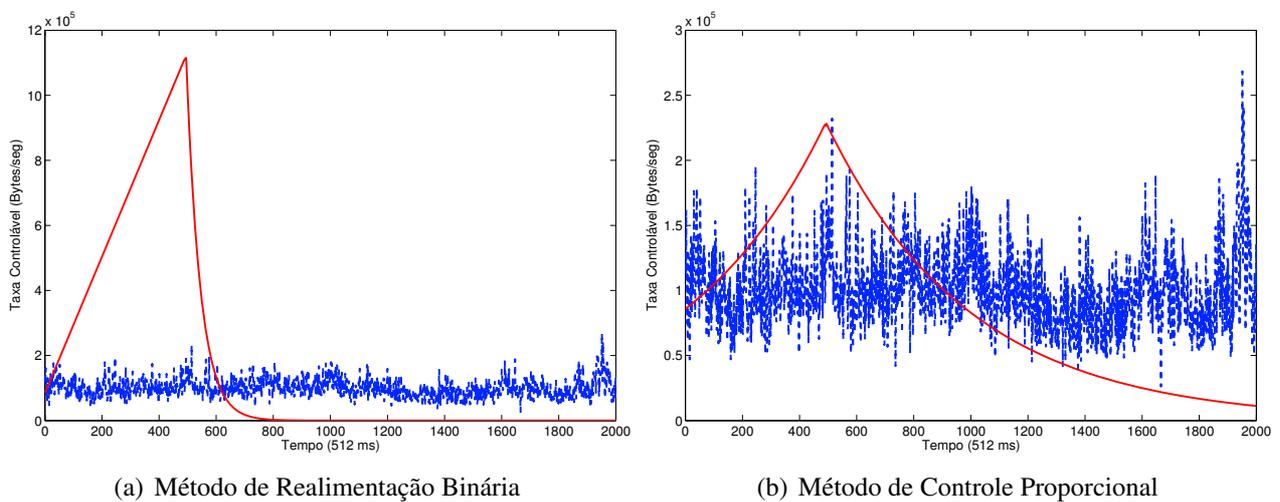


Fig. 6.11: Taxa da Fonte Controlável Antes e Depois da Aplicação do Controle de Realimentação Binária e do Controle Proporcional para o Nó 1 do Cenário. Antes (Linha Tracejada), Depois (Linha Cheia).

No nó 2, todos os métodos obtiveram resultados sem perdas, porém, o método proposto foi o que melhor conseguiu reduzir o tamanho da fila no *buffer* obtendo média e variância menores, conforme pode ser visto na Tabela 6.4. Nesse nó, a utilização do enlace após a aplicação do método proposto pouco foi reduzida, enquanto que nos métodos de comparação a utilização não modificou.

E por fim, no nó 3, apenas o método de Realimentação Binária resultou em um *buffer* com perdas, fato já explicado anteriormente, e mais uma vez, uma menor variância do processo de fila no *buffer* foi obtida utilizando o método proposto. Os resultados foram resumidos na Tabela 6.5.

Tab. 6.3: Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento no Nó 1 - Cenário 1.

Parâmetro	Realimentação Binária		Controle Cong. Proporcional		Controle Cong. Proposto	
	Antes	Depois	Antes	Depois	Antes	Depois
Média da Fila	$1.10 \times 10^6$	$4.50 \times 10^5$	$1.10 \times 10^6$	$7.12 \times 10^5$	$1.10 \times 10^6$	$1.12 \times 10^3$
Variância da Fila	$3.62 \times 10^{11}$	$4.51 \times 10^{11}$	$3.62 \times 10^{11}$	$4.92 \times 10^{11}$	$3.62 \times 10^{11}$	$5.67 \times 10^7$
TPB	9.03%	45.33%	9.03%	11.09%	9.03%	0.00%
Utilização	99.90%	99.95%	99.90%	99.95%	99.90%	88.01%
Vazão	$1.00 \times 10^5$	$6.39 \times 10^4$	$1.00 \times 10^5$	$8.98 \times 10^4$	$1.00 \times 10^5$	$2.34 \times 10^3$

Tab. 6.4: Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento no Nó 2 - Cenário 1.

Parâmetro	Realimentação Binária		Controle Cong. Proporcional		Controle Cong. Proposto	
	Antes	Depois	Antes	Depois	Antes	Depois
Média da Fila	$1.41 \times 10^6$	$1.02 \times 10^4$	$1.48 \times 10^6$	$2.85 \times 10^4$	$1.40 \times 10^6$	$6.35 \times 10^3$
Variância da Fila	$3.60 \times 10^{10}$	$1.53 \times 10^9$	$1.11 \times 10^{10}$	$4.88 \times 10^9$	$4.57 \times 10^{10}$	$6.44 \times 10^8$
TPB	51.78%	0.00%	39.05%	0.00%	17.04%	0.00%
Utilização	99.90%	99.90%	99.90%	99.90%	99.90%	99.85%
Vazão	$1.32 \times 10^5$	$1.97 \times 10^3$	$1.65 \times 10^5$	$1.50 \times 10^4$	$1.02 \times 10^5$	$2.32 \times 10^3$

Tab. 6.5: Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento no Nó 3 - Cenário 1.

Parâmetro	Realimentação Binária		Controle Cong. Proporcional		Controle Cong. Proposto	
	Antes	Depois	Antes	Depois	Antes	Depois
Média da Fila	$1.30 \times 10^6$	$1.21 \times 10^5$	$1.35 \times 10^6$	$3.11 \times 10^4$	$1.36 \times 10^6$	$2.84 \times 10^4$
Variância da Fila	$1.44 \times 10^{11}$	$1.28 \times 10^{11}$	$9.10 \times 10^{10}$	$5.54 \times 10^9$	$7.58 \times 10^{10}$	$4.28 \times 10^9$
TPB	11.73%	1.71%	16.23%	0.00%	11.49%	0.00%
Utilização	99.95%	99.95%	99.95%	99.95%	99.95%	99.949%
Vazão	$8.69 \times 10^4$	$1.15 \times 10^4$	$9.99 \times 10^4$	$1.53 \times 10^4$	$8.73 \times 10^4$	$2.29 \times 10^4$

Através das Tabelas 6.3, 6.4 e 6.5, pode-se notar que a média e a variância do tamanho da fila no *buffer* diminuem consideravelmente em todos os nós do cenário considerado após a aplicação do método proposto. Conforme já comentado anteriormente, a TPB é reduzido a zero, oferecendo melhorias de desempenho em termos de taxa de perdas e do comportamento da fila no *buffer*. A

utilização do enlace, após o controle proposto, no entanto, é um pouco reduzida em todos os nós (em no máximo 11,9% no caso do Nó 1) devido a grande habilidade do controle em limitar a taxa da fonte controlável em favor da fonte não-controlável. Logo, se a vazão do tráfego controlável é reduzida, a utilização do enlace acaba sendo um pouco reduzida também.

Assim, podemos concluir que o método de controle de congestionamento proposto obtém melhores resultados e controla, em todos os nós, a taxa da fonte de modo que o tamanho da fila no *buffer* fique abaixo do tamanho do *buffer*  $B_{max}$ .

## 6.6 Etapa 3 da Análise: Avaliação em 1 nó - Traços monofractais fGn

Nessa seção é analisado o controle de congestionamento proposto utilizando as séries obtidas com o modelo fGn (*fractional Gaussian noise*) já comentadas no Capítulo anterior na seção 5.4.4. Através da análise do diagrama multiescala linear dessas séries, constatou-se que as mesmas possuem características monofractais. O intuito da análise do esquema de controle proposto com essas séries é verificar como o controle se comporta com séries monofractais. São utilizadas as séries fGn com  $H = 0.5$  e fGn com  $H = 0.8$  para especificar, respectivamente, o tráfego não-controlável e controlável que entram em um nó. O tamanho do *buffer*  $B_{max}$  foi fixado em 10, o nível desejado do tamanho da fila no *buffer* é estabelecido como  $b^\tau = 30\%B_{max}$ , ou seja,  $b^\tau = 3$  e o parâmetro  $\lambda$  do cálculo da taxa de controle de fluxo ótima é fixado em 1. Essas séries fGn são normalizadas, logo os parâmetros do sistema não possuem unidade de medida.

Na Figura 6.12.a, é mostrada a taxa dos fluxos controlável e não-controlável que entra no nó considerado. E na Figura 6.12.b é mostrada a taxa do fluxo controlável antes e depois da aplicação do controle proposto. Como se pode perceber da Figura 6.12.b, a taxa controlável é consideravelmente reduzida, e em alguns instantes até levada a um valor nulo, em favor da taxa não-controlável.

A Figura 6.13.a mostra o comportamento da fila no *buffer* antes e depois da aplicação do controle proposto. É possível perceber que o tamanho da fila foi bastante reduzido e ficou em todos os instantes abaixo do nível desejado  $b^\tau = 3$ . Uma comparação do tamanho da fila no *buffer* resultante do controle proposto e dos outros métodos comparativos é mostrada na Figura 6.13.b, onde é possível observar que o processo de fila no *buffer* mais comportado é o que foi obtido pelo método proposto e que, nos métodos comparativos, houve perdas. A explicação para essas perdas deve-se à lentidão dos métodos de comparação para diminuir sua taxa controlável em resposta a um aumento no tamanho da fila no *buffer*.

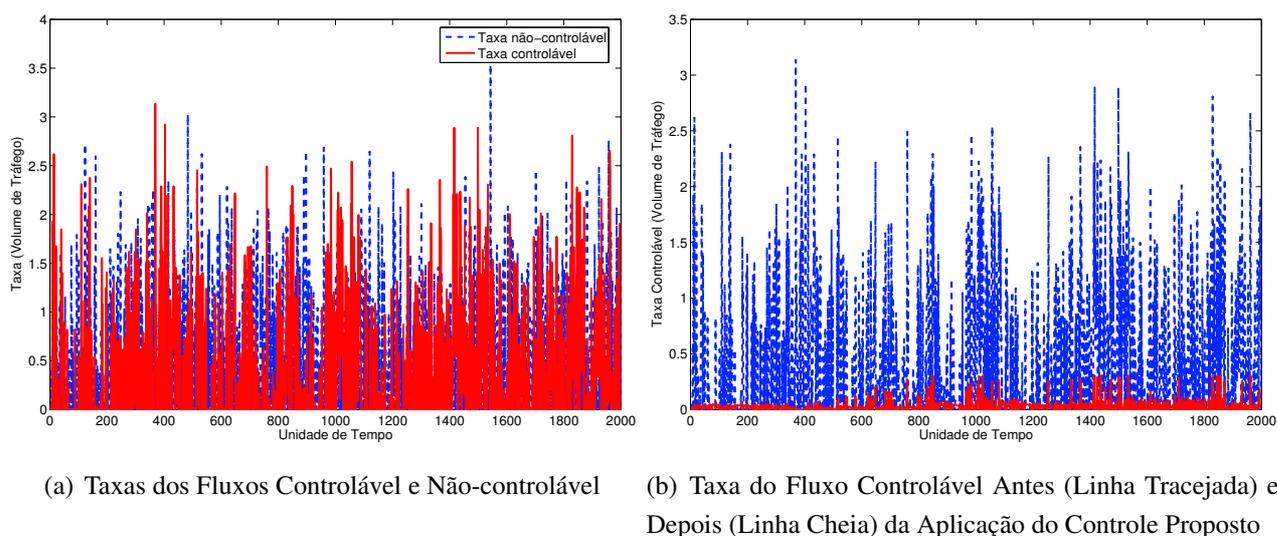
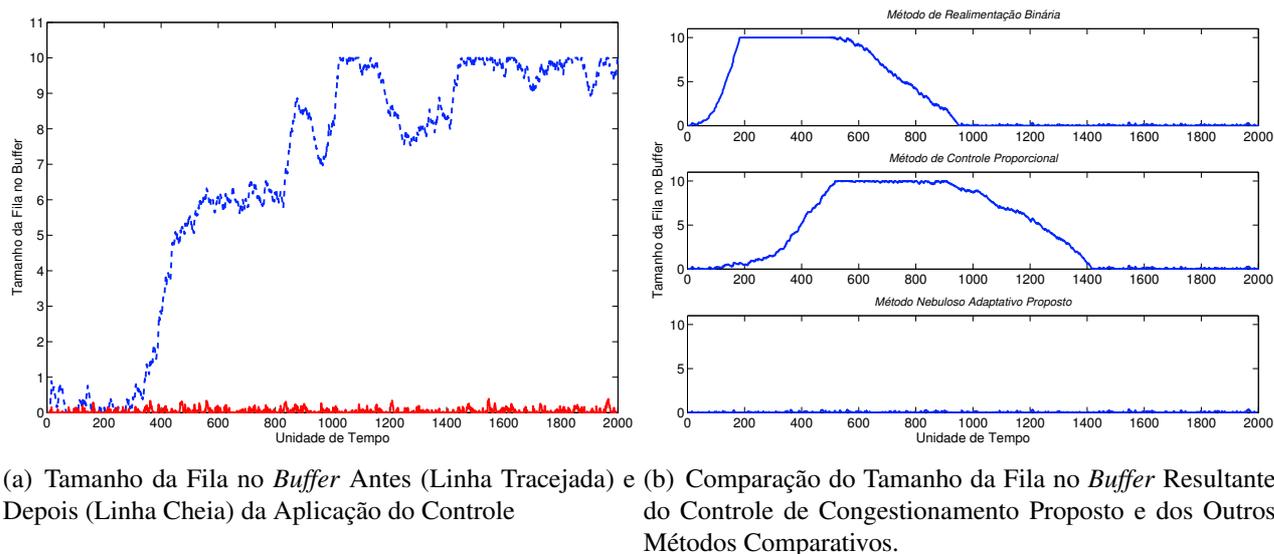


Fig. 6.12: Análise das Taxas Controlável e Não-controlável .

Fig. 6.13: Análise do Tamanho da Fila no *Buffer*.

Como podemos observar na Figura 6.13.a, o tamanho da fila no *buffer* é pequeno nas primeiras 400 amostras. Em resposta a esse pequeno tamanho da fila, os métodos de comparação aumentam a taxa controlável nesse período, conforme pode ser observado na Figura 6.14. No entanto, quando o tamanho da fila aumenta, a taxa deve ser diminuída mas devido à lentidão desses métodos, perdas acontecem como pode ser observado na Figura 6.13.b. Na Tabela 6.6, os resultados de desempenho do método proposto e dos métodos comparativos são resumidos, onde podemos observar que, apenas no método de controle proposto, a variância da fila no *buffer* diminuiu e a taxa de perda de bytes obtida

foi nula. Entretanto, dentre todos os métodos, o método proposto foi o que mais controlou a taxa controlável, podendo ser observado pela menor vazão. Desse modo, temos que o método proposto é um método de controle mais ativo que visa variância do tamanho da fila mínima e perdas nulas em função de uma menor vazão da taxa controlável quando comparado aos demais métodos.

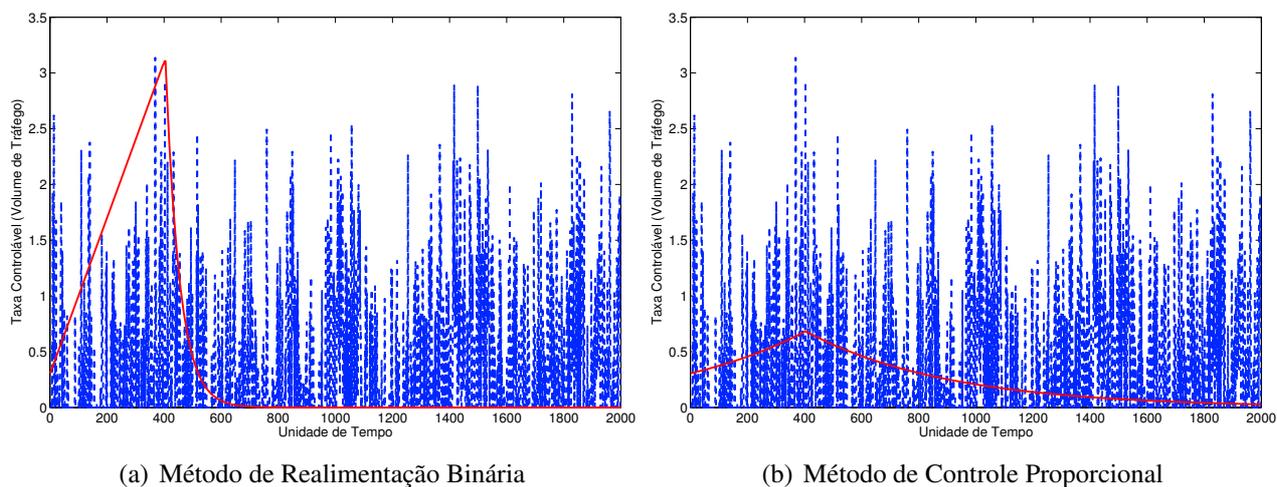


Fig. 6.14: Taxa da Fonte Controlável Antes e Depois da Aplicação do Controle de Realimentação Binária e do Controle Proporcional. Antes (Linha Tracejada), Depois (Linha Cheia).

Tab. 6.6: Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento.

Parâmetro	Sem Controle	Realimentação Binária	Controle Proporcional	Método Proposto
Média da Fila	6.65	3.18	4.13	0.03
Variância da Fila	12.16	17.03	16.96	0.004
TPB	8.24%	34.61%	4.01%	0.00%
Utilização	99.61%	99.95%	99.95%	99.62%
Vazão	0.26	.018	0.26	0.013

Dos resultados obtidos, pode-se concluir que o método de controle proposto obtêm bons resultados de desempenho quando aplicado em traços de tráfego monofractais. É também observado que a utilização aumentou ligeiramente, comportamento não verificado nas etapas anteriores.

## 6.7 Etapa 4 da Análise: Avaliação com 3 Nós - Traços de Tráfego 3-7-I-1, 4-7-I-1, 4-7-I-9, 4-7-I-5 e 4-7-I-2

### 6.7.1 Cenário de Simulação (Séries Multifractais)

No cenário 2, uma rede também composta por 3 nós e vários tráfegos de entrada é considerada. A Figura 6.6 também representa o cenário proposto para a Parte 2, ou seja, as Fontes 1 e 3 são fontes de tráfego não-controláveis e as Fontes 2, 4 e 5 são fontes de tráfego controlável.

Para representar os fluxos de tráfego de entrada, foram usados os traços da Petrobrás 3-7-I-1, 4-7-I-1, 4-7-I-9, 4-7-I-5 e 4-7-I-2 com escala de agregação de  $100ms$ . Tais traços de tráfego, conforme analisado no Capítulo 5 seção 5.4.4, possuem características multifractais. O objetivo nessa etapa é avaliar um cenário de rede apenas com traços de comportamento multifractal, visto que na etapa 2 da análise o cenário foi simulado com traços monofractais e multifractais. As especificações detalhadas dos componentes do cenário são descritas a seguir:

**Fonte 1:** Tráfego não-controlável 4-7-I-1 (amostras 1-1201).

**Fonte 2:** Tráfego controlável 3-7-I-1 (amostras 1-1201).

**Nó 1:** Capacidade do Enlace:  $18k$  bytes/seg,  $d = 1$  intervalo de tempo ( $100ms$ ).

**Fonte 3:** Tráfego não-controlável 4-7-I-5 (amostras 1-1201).

**Fonte 4:** Tráfego controlável 4-7-I-9 (amostras 1-1201).

**Destino 1:** Saída do Tráfego da Fonte 1.

**Nó 2:** Capacidade do Enlace:  $20k$  bytes/seg,  $d = 2$  intervalos de tempo ( $2 \times 100ms$ ).

**Fonte 5:** Tráfego Controlável 4-7-I-2 (amostras 1-1201).

**Nó 3:** Capacidade do Enlace:  $22k$  bytes/seg,  $d = 1$  intervalo de tempo ( $100ms$ ).

**Destinos 2,3,4 e 5:** Saída do Tráfego das Fontes 2, 3, 4 e 5 respectivamente.

### 6.7.2 Desempenho do Esquema de Controle de Congestionamento

Para a simulação do controle de congestionamento proposto aplicado ao cenário em questão, devemos estipular uma configuração para a rede considerada de tal forma a obtermos um ambiente com perdas. A partir das informações da taxa média e da taxa de pico dos traços utilizados conforme

mostra a Tabela 5.1, o tamanho do *buffer* dos roteadores é fixado em  $B_{max} = 2 \times 10^5$  bytes, o nível desejado do tamanho da fila no *buffer* é estabelecido como  $b^r = 30\%B_{max}$ , ou seja,  $b^r = 6 \times 10^4$  bytes. O parâmetro  $\lambda$  do cálculo da taxa ótima de controle de fluxo é fixado em 1 e o esquema de controle proposto é aplicado em cada nó do cenário considerado.

Na Figura 6.15 as taxas dos fluxos controláveis e não-controláveis são mostradas para cada nó do cenário considerado. Podemos perceber que as duas taxas estão, em média, na mesma ordem.

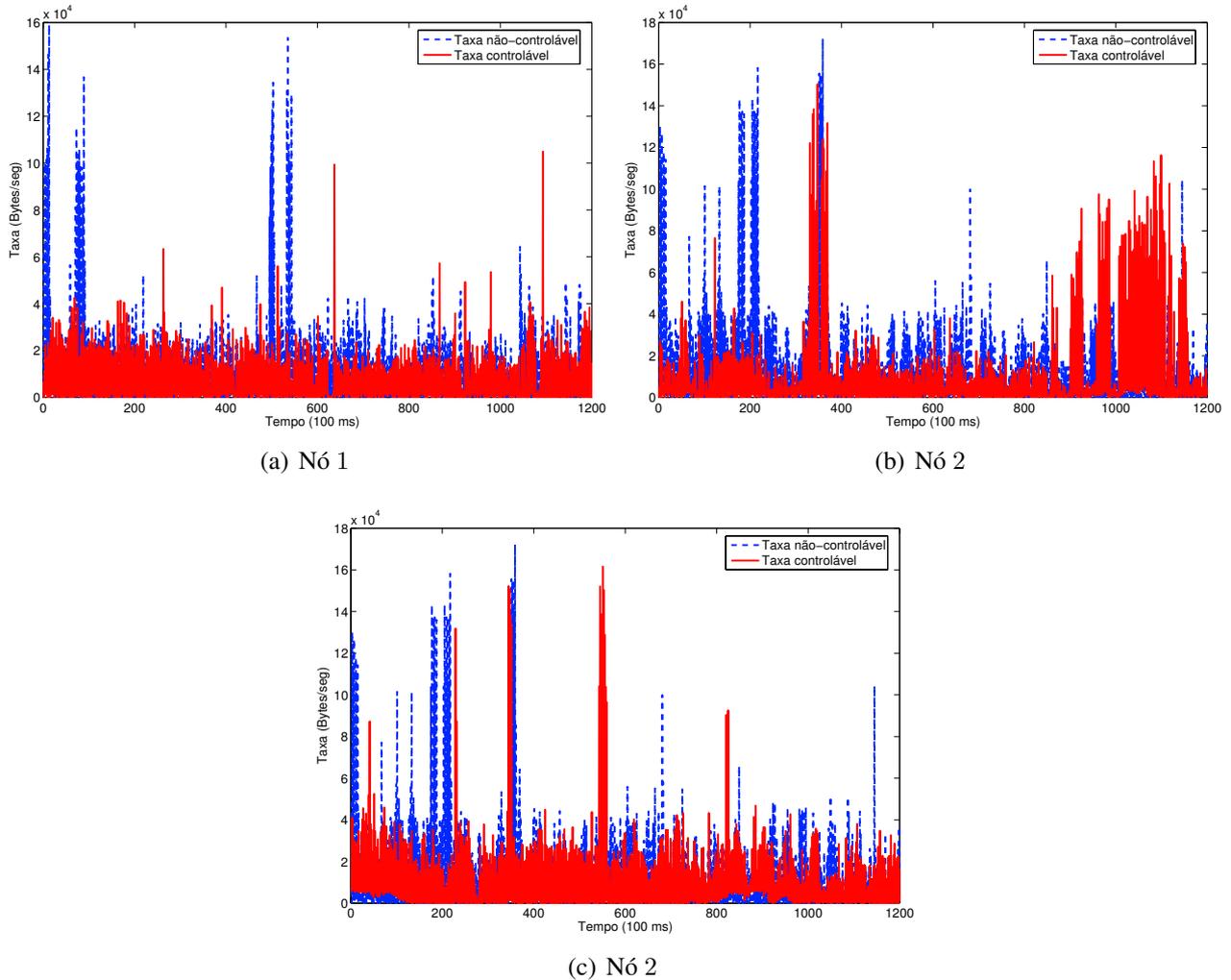


Fig. 6.15: Taxas dos Fluxos Controlável e Não-Controlável para Cada Nó do Cenário.

A Figura 6.16 mostra o comportamento da fila no *buffer* nos Nós 1, 2 e 3 antes e depois da aplicação do esquema de controle de congestionamento proposto. Como se pode observar, após a aplicação do controle, o tamanho da fila no *buffer* em nenhum instante se tornou maior do que o tamanho do *buffer*  $B_{max}$ , logo, a taxa de perda de bytes (TPB) é igual a 0% em todos os nós.

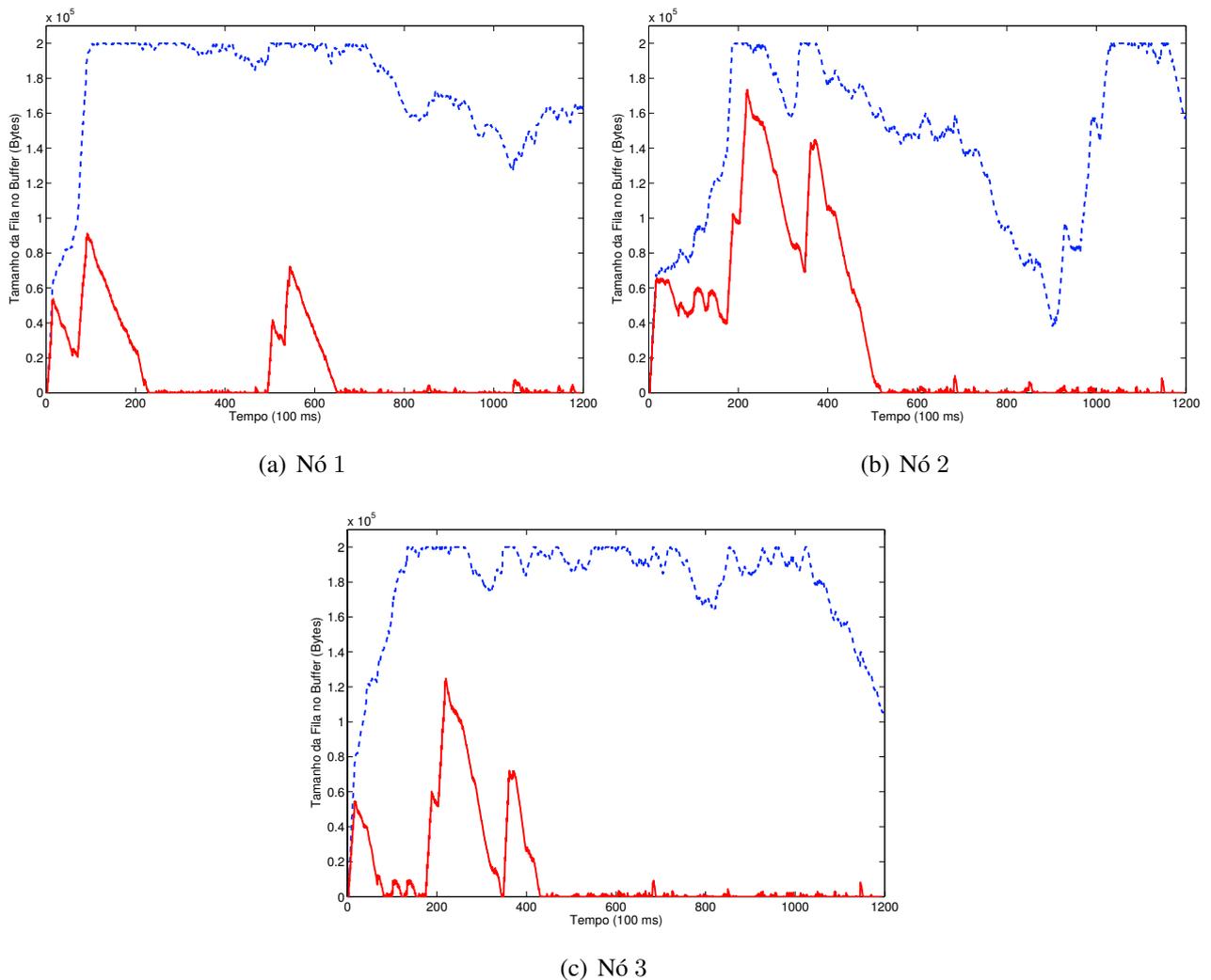


Fig. 6.16: Tamanho da Fila no *Buffer* Antes e Depois da Aplicação do Controle de Congestionamento Proposto para Cada Nó do Cenário. Antes (Linha Tracejada), Depois (Linha Cheia).

A Figura 6.17 compara a taxa do fluxo controlável antes e depois da aplicação do controle de congestionamento proposto. Através da Figura 6.17, é possível observar que a taxa do fluxo controlável foi reduzida e algumas vezes até mesmo levada a um valor nulo, fato já explicado anteriormente.

Comparamos o esquema de controle de congestionamento proposto com os métodos mencionados na seção 6.2. A Figura 6.18 mostra o tamanho da fila no *buffer*, onde para cada figura referente a cada nó, temos respectivamente o método de Realimentação Binária, em seguida o método de Controle Proporcional e, por fim, o método proposto. Através da Figura 6.18, podemos observar que o método proposto obtém um processo de fila no *buffer* mais comportado dentre os outros métodos comparativos.

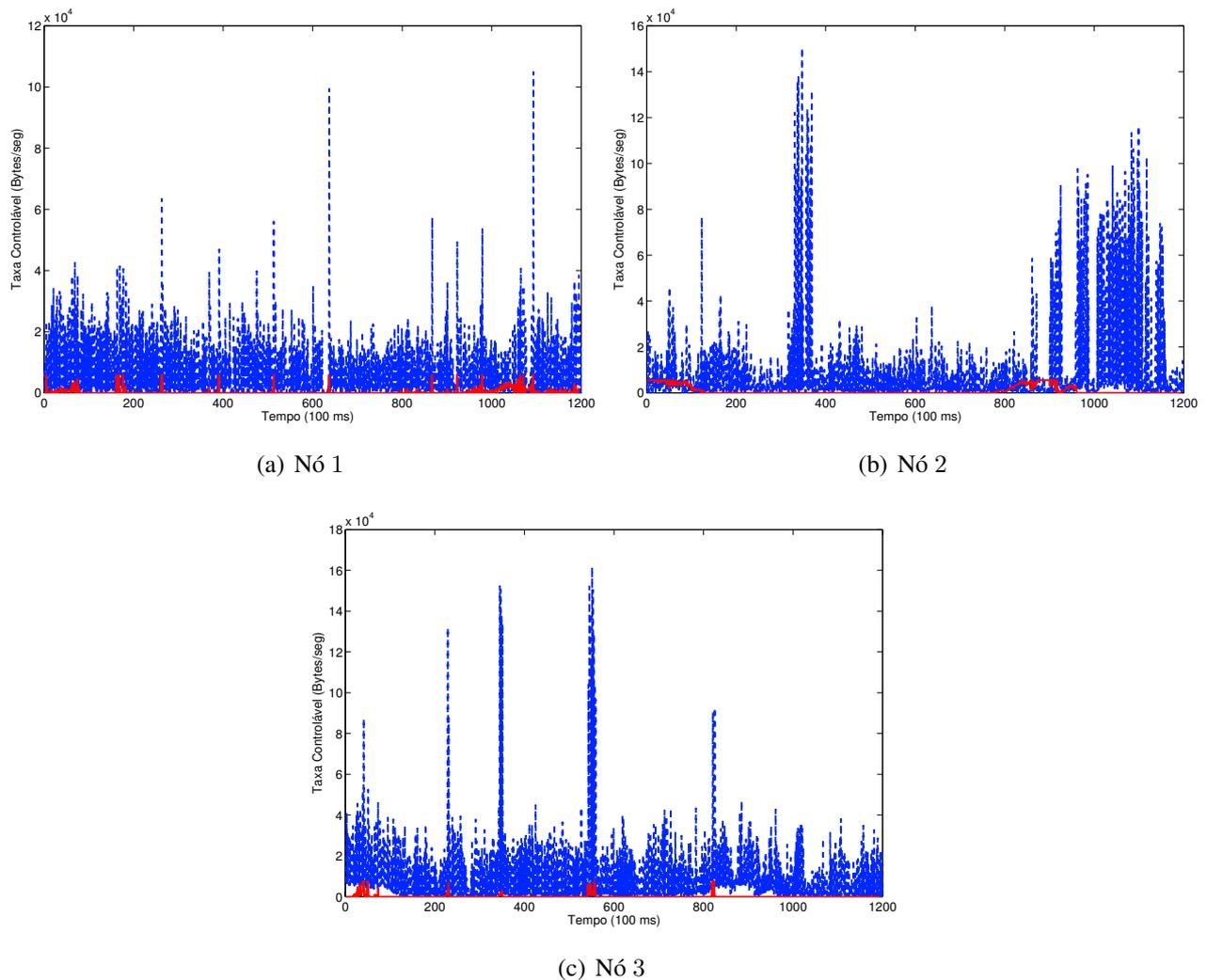


Fig. 6.17: Taxa da Fonte Controlável Antes e Depois da Aplicação do Controle de Congestionamento Proposto para Cada Nó do Cenário. Antes (Linha Tracejada), Depois (Linha Cheia).

No nó 1, apenas no caso do método de Realimentação Binária houve perdas, pois o tamanho da fila no *buffer* teve que ser limitado no tamanho do *buffer*  $B_{max} = 2 \times 10^5$  bytes. Conforme pode ser observado na Tabela 6.7, a variância do tamanho da fila diminuiu apenas no método proposto. Em contrapartida, a utilização de enlace do nó 1 após a aplicação do método proposto diminuiu ligeiramente.

No nó 2, apenas no método proposto não foi constatado perdas e, em todos os métodos, a variância do tamanho da fila no *buffer* aumentou, porém com o método proposto essa variância obteve menor aumento. Entretanto, analisando a vazão da taxa controlável, podemos também concluir que o método proposto foi o que mais reduziu a sua vazão. Fato este explicado pela eficiência do algoritmo em controlar a fonte controlável visando obter perdas nulas e variância do tamanho da fila mínima.

Os resultados são resumidos na Tabela 6.8.

Finalmente, analisando o nó 3, foram obtidas perdas apenas com o método de Controle Proporcional. E apenas o método proposto conseguiu reduzir a variância do tamanho da fila. Os resultados são apresentados na Tabela 6.9.

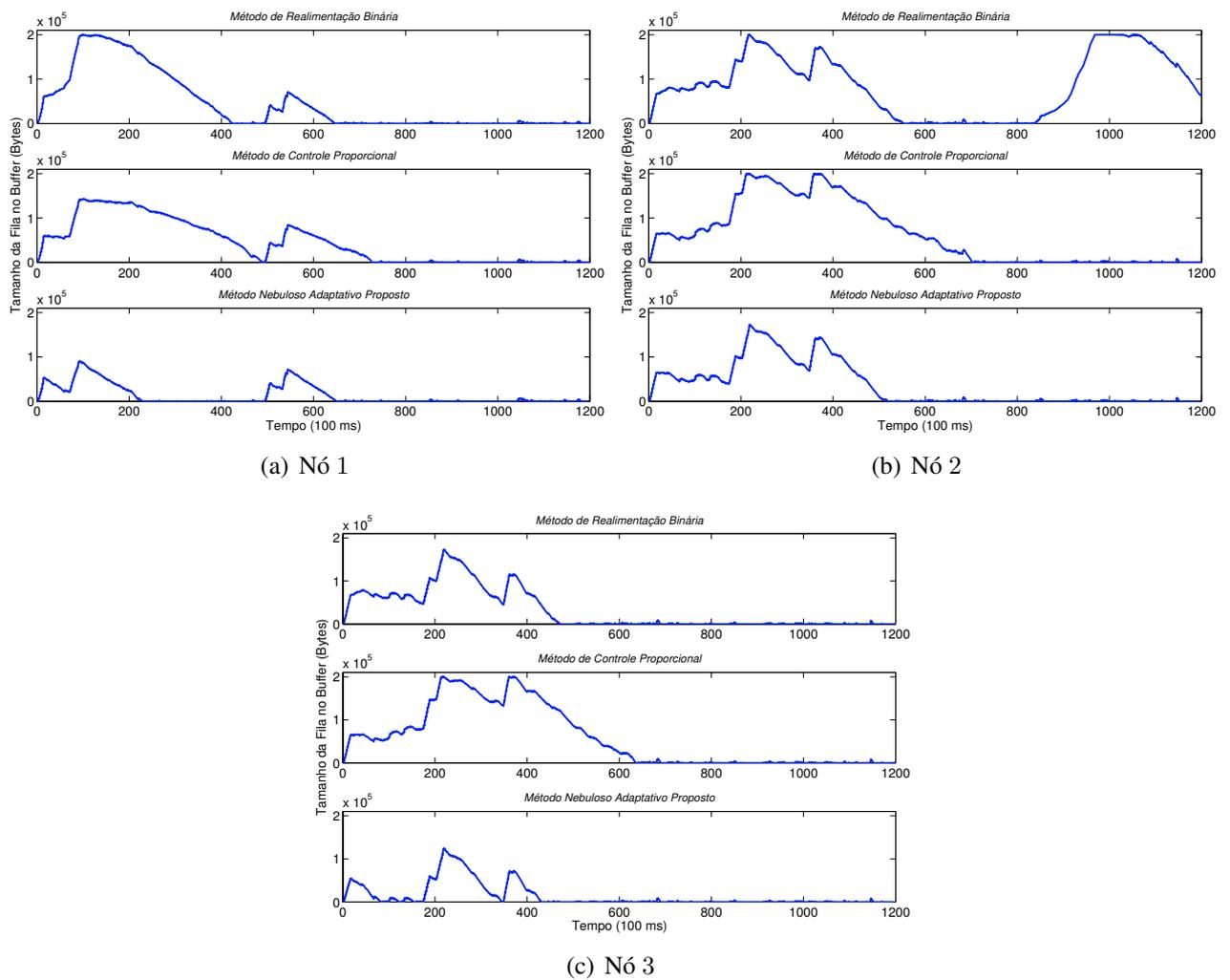


Fig. 6.18: Comparação do Tamanho da Fila no *Buffer* Resultante do Controle de Congestionamento Proposto e dos Outros Métodos Comparativos para Cada Nó do Cenário.

Tab. 6.7: Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento no Nó 1 - Cenário 2.

Parâmetro	Realimentação Binária		Controle Cong. Proporcional		Controle Cong. Proposto	
	Antes	Depois	Antes	Depois	Antes	Depois
Média da Fila	$1.74 \times 10^5$	$4.47 \times 10^4$	$1.74 \times 10^5$	$4.47 \times 10^4$	$1.74 \times 10^5$	$1.34 \times 10^4$
Variância da Fila	$1.16 \times 10^9$	$4.10 \times 10^9$	$1.16 \times 10^9$	$2.33 \times 10^9$	$1.16 \times 10^9$	$5.21 \times 10^8$
TPB	7.38%	0.17%	7.38%	0.00%	7.38%	0.00%
Utilização	99.87%	99.91%	99.87%	99.91%	99.87%	99.84%
Vazão	$7.30 \times 10^3$	$1.41 \times 10^3$	$7.30 \times 10^3$	$2.88 \times 10^3$	$7.30 \times 10^3$	190.34

Tab. 6.8: Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento no Nó 2 - Cenário 2.

Parâmetro	Realimentação Binária		Controle Cong. Proporcional		Controle Cong. Proposto	
	Antes	Depois	Antes	Depois	Antes	Depois
Média da Fila	$1.51 \times 10^5$	$8.73 \times 10^4$	$1.77 \times 10^5$	$6.33 \times 10^4$	$1.43 \times 10^5$	$3.56 \times 10^4$
Variância da Fila	$2.21 \times 10^9$	$4.87 \times 10^9$	$1.12 \times 10^9$	$4.76 \times 10^9$	$2.29 \times 10^9$	$2.38 \times 10^9$
TPB	22.00%	5.48%	25.90%	3.39%	18.35%	0.00%
Utilização	99.88%	99.91%	99.88%	99.91%	99.85%	99.85%
Vazão	$7.47 \times 10^3$	$4.50 \times 10^3$	$8.79 \times 10^3$	$2.53 \times 10^3$	$6.48 \times 10^3$	874.60

Tab. 6.9: Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento no Nó 3 - Cenário 2.

Parâmetro	Realimentação Binária		Controle Cong. Proporcional		Controle Cong. Proposto	
	Antes	Depois	Antes	Depois	Antes	Depois
Média da Fila	$1.87 \times 10^5$	$3.13 \times 10^4$	$1.78 \times 10^5$	$5.70 \times 10^4$	$1.78 \times 10^5$	$1.39 \times 10^4$
Variância da Fila	$6.40 \times 10^8$	$2.05 \times 10^9$	$1.01 \times 10^9$	$4.60 \times 10^9$	$9.40 \times 10^8$	$7.56 \times 10^8$
TPB	26.09%	0.00%	20.12%	2.05%	15.09%	0.00%
Utilização	99.89%	99.91%	99.89%	99.91%	99.89%	99.83%
Vazão	$1.20 \times 10^4$	787.84	$1.07 \times 10^4$	$3.22 \times 10^3$	$9.43 \times 10^3$	152.97

Através das Tabelas 6.7, 6.8 e 6.9, pode-se notar que apenas o método proposto conseguiu reduzir a variância do tamanho da fila no *buffer* e obter TPB igual a zero, em todos os nós do cenário considerado, oferecendo melhorias no desempenho em termos de taxa de perdas e comportamento da fila no *buffer*. E, assim como nas etapas anteriores da análise, a utilização do enlace, após a aplicação

controle proposto, é ligeiramente reduzida devido a grande habilidade do controle em limitar a taxa da fonte controlável em favor da fonte não-controlável. De uma forma geral, podemos concluir que o método de controle de congestionamento proposto obtém melhores resultados e controla, em todos os nós, a taxa da fonte de modo que a variância do tamanho da fila no *buffer* seja mínima possível e que o tamanho da fila no *buffer* fique em torno do nível desejado  $b^r$  e assim nenhuma perda seja verificada. Contudo, tais melhorias de desempenho são conseguidas com um decremento na utilização do enlace e na vazão da taxa controlável.

## 6.8 Análise da Predição do Tamanho da Fila no *Buffer*

No capítulo anterior, verificamos o desempenho do preditor nebuloso proposto utilizando traços de tráfego monofractais e multifractais. A análise dos resultados nos mostra que o preditor proposto consegue melhores resultados de desempenho de predição (EQM, EQMN1 e EQMN2) em comparação a outros preditores adaptativos. Nessa seção, alguns resultados da predição do tamanho da fila no *buffer* serão mostrados. Consideramos nessa análise o cenário da etapa 4 da análise, que utiliza traços de tráfego multifractais da Petrobrás. Para a escolha do número de amostras passadas do tamanho da fila no *buffer* e da taxa controlável, uma análise foi realizada a fim de obtermos um equilíbrio entre a qualidade de predição e a complexidade do algoritmo, visto que se utilizamos poucas amostras (pequena ordem), temos um menor tempo de processamento de informações, mas por outro lado, o algoritmo de predição pode não ter informações suficientes como dados de entrada, resultando em uma queda na qualidade da predição. Analisando as medidas de desempenho de predição variando o número de amostras passadas, os resultados são mostrados na Tabela 6.10. A partir dessa análise, escolhemos duas amostras passadas para o tamanho da fila no *buffer* e para a taxa controlável, ou seja,  $p = 2$  e  $q = 2$ . As taxas de aprendizagem do algoritmo de refinamento foram estipuladas de tal forma a obter melhores resultados de desempenho, assim o valor encontrado foi  $\eta = 0.62$  e  $\zeta = 0.72$ . Os resultados de desempenho da predição do tamanho da fila no *buffer* nos Nós 1, 2 e 3 do cenário considerado são detalhados na Tabela 6.11, 6.12 e 6.13, respectivamente. A predição do tamanho da fila no *buffer* em todos os nós é mostrado na Figura 6.19.

Dos resultados obtidos e mostrados nas Tabelas 6.11, 6.12 e 6.13 pode-se afirmar que o preditor nebuloso proposto obteve melhores resultados em comparação aos preditores adaptativos RLS e LMS.

Tab. 6.10: Resultados de Desempenho do Preditor Proposto para Diferentes Ordens  $n$  Considerando o Nó 1.

<b>Preditor</b>	<b>EQM</b>	<b>EQMN1</b>	<b>EQMN2</b>
$n = 1$	0.0119	0.4258	226.8590
$n = 2$	0.0018	0.0643	34.3912
$n = 3$	0.0024	0.0896	46.9976
$n = 4$	0.0117	0.4469	236.0474

Tab. 6.11: Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Nó 1.

<b>Preditor</b>	<b>EQM</b>	<b>EQMN1</b>	<b>EQMN2</b>
Proposto	0.0018	0.0643	34.3912
RLS	0.0018	0.0663	35.4993
LMS	0.0049	0.1778	95.2074

Tab. 6.12: Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Nó 2.

<b>Preditor</b>	<b>EQM</b>	<b>EQMN1</b>	<b>EQMN2</b>
Proposto	0.0025	0.0448	28.8276
RLS	0.0028	0.0495	31.3992
LMS	0.0049	0.0869	55.0828

Tab. 6.13: Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Nó 3.

<b>Preditor</b>	<b>EQM</b>	<b>EQMN1</b>	<b>EQMN2</b>
Proposto	$5.1630 \times 10^{-4}$	0.0244	9.7249
RLS	$5.8918 \times 10^{-4}$	0.0279	11.0977
LMS	0.0048	0.2321	92.2063

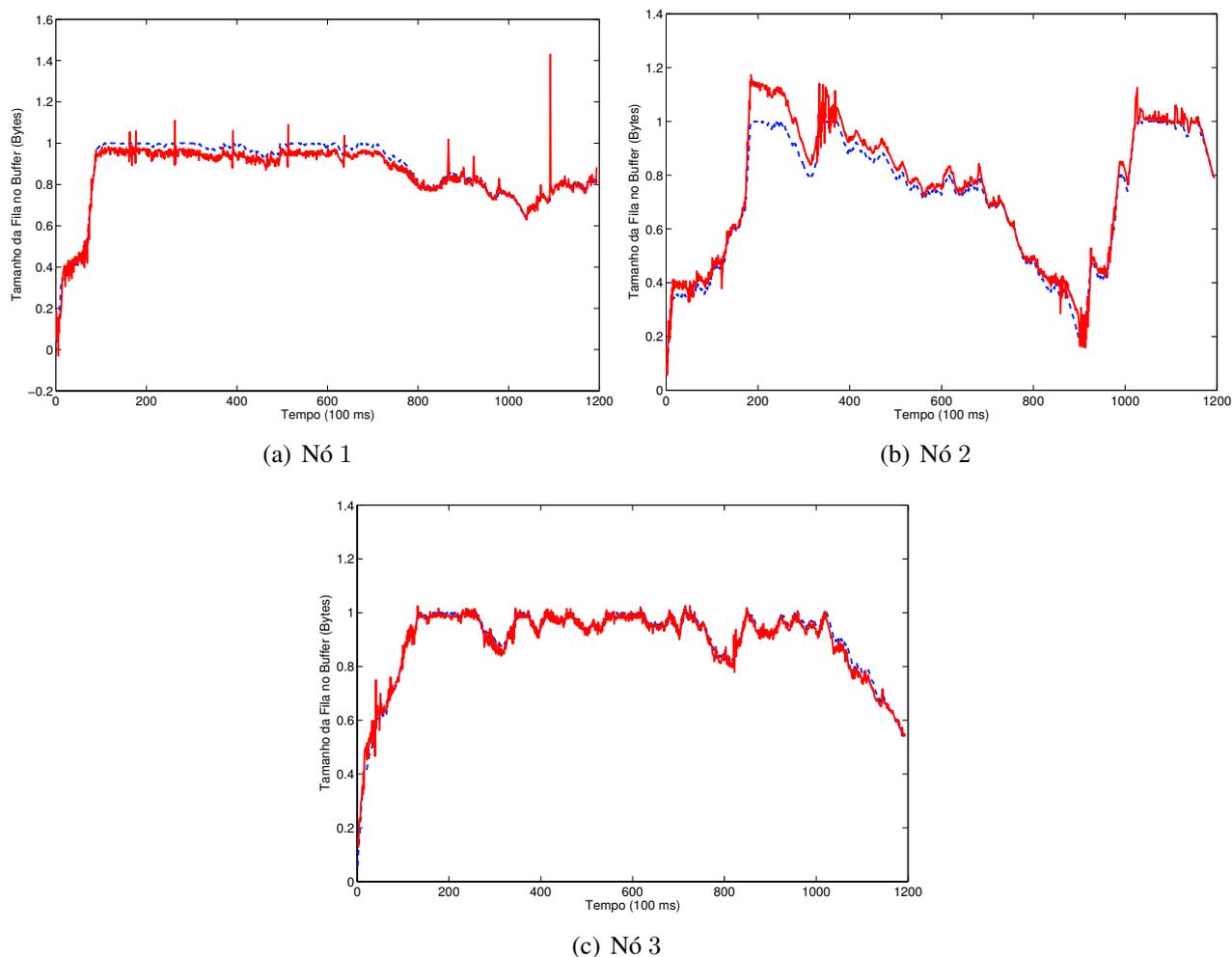


Fig. 6.19: Predição a um Passo do Tamanho da Fila no *Buffer*. Real (Linha Tracejada), Predito (Linha Cheia).

Com o objetivo de analisar a predição do tamanho da fila no *buffer* em um nó que utiliza traços monofractais, tal como o nó da seção 6.6, resumimos na Tabela 6.14 as medidas de desempenho de predição obtidas pelo preditor proposto e pelos preditores RLS e LMS para esse nó do cenário considerado na seção 6.6. A predição do tamanho da fila é mostrada na Figura 6.20. Dos resultados mostrados na Tabela 6.14, é possível observar que o preditor proposto obtém melhores resultados de desempenho quando comparado aos outros preditores. A Tabela 6.15 analisa os resultados para diferentes amostras passadas utilizadas, onde as medidas de desempenho se tornam melhores quando esse número de amostra aumenta até  $n = 2$  e a partir desse valor, as medidas começam a aumentar caracterizando uma queda na qualidade da predição.

Tab. 6.14: Resultados de Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Nó com Traços Monofractais fGn.

<b>Preditor</b>	<b>EQM</b>	<b>EQMN1</b>	<b>EQMN2</b>
Proposto	0.0038	0.0315	72.4058
RLS	0.0039	0.0319	66.2528
LMS	0.0045	0.0375	77.9536

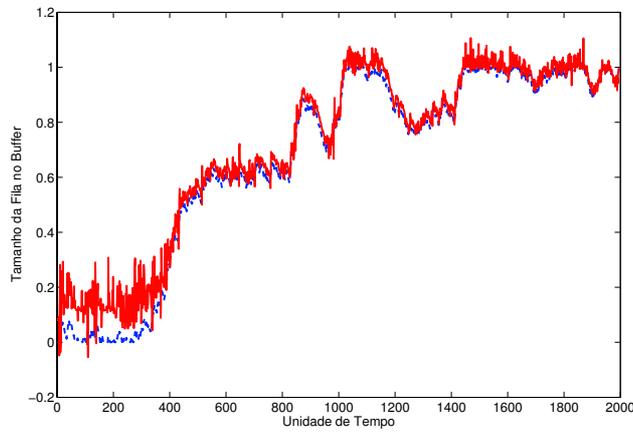


Fig. 6.20: Predição a um Passo do Tamanho da Fila no *Buffer*. Real (Linha Tracejada), Predito (Linha Cheia).

Tab. 6.15: Resultados de Desempenho do Preditor Proposto Para Diferentes Ordens  $n$  Considerando o Nó com Traços Monofractais fGn.

<b>Preditor</b>	<b>EQM</b>	<b>EQMN1</b>	<b>EQMN2</b>
$n = 1$	0.0310	0.2559	589.5114
$n = 2$	0.0038	0.0315	72.4058
$n = 3$	0.0444	0.3667	841.9693
$n = 4$	0.0691	0.5717	$1.3103 \times 10^3$

## 6.9 Análise do Efeito do Fator de Ponderação ( $\lambda$ ) no Desempenho do Sistema

Na seção 4.4, o parâmetro  $\lambda$  foi definido como um fator de ponderação para o cálculo da taxa ótima de controle. Esse parâmetro é considerado um parâmetro de projeto e, portanto, o seu valor deve ser escolhido apropriadamente. Nessa seção, será analisado o efeito desse fator de ponderação  $\lambda$  no desempenho do sistema variando  $\lambda$  em 0, 0.01, 0.1, 0.8, 0.9 e 0.95 para o caso do Nó 1 do cenário proposto na etapa 2 da análise do esquema de controle proposto. Os resultados obtidos nas simulações se encontram na Tabela 6.16.

Tab. 6.16: Resultados de Desempenho do Esquema de Controle de Congestionamento Proposto para Diferentes  $\lambda$ 's.

$\lambda$	Média da Fila	Variância da Fila	TPB	Utilização	Vazão
0	$1.7740 \times 10^3$	$7.1714 \times 10^7$	0%	99.5235%	$1.5478 \times 10^4$
0.01	$1.7231 \times 10^3$	$7.0840 \times 10^7$	0%	99.5235%	$1.4994 \times 10^4$
0.1	$1.3987 \times 10^3$	$6.3709 \times 10^7$	0%	93.8787%	$1.0317 \times 10^4$
0.8	$1.1311 \times 10^3$	$5.6731 \times 10^7$	0%	91.0922%	$2.9229 \times 10^3$
0.9	$1.1297 \times 10^3$	$5.6725 \times 10^7$	0%	88.1261%	$2.6043 \times 10^3$
0.95	$1.1296 \times 10^3$	$5.6723 \times 10^7$	0%	88.0678%	$2.4696 \times 10^3$

A partir do resultados da Tabela 6.16, pode-se observar que, quando  $\lambda$  aumenta, a média e a variância do tamanho da fila no *buffer* diminuem, caracterizando melhorias nas medidas de desempenho do comportamento da fila. Por outro lado, é possível observar que a vazão da taxa controlável e a utilização do enlace diminuem quando  $\lambda$  aumenta. Dessas observações feitas, é possível concluir que um valor adequado para o fator de ponderação  $\lambda$  deve ser escolhido, de tal modo a se obter uma média e variância do tamanho da fila no *buffer* e utilização do enlace satisfatórios.

## 6.10 Aplicações

O esquema de controle de congestionamento proposto foi detalhado em um contexto geral de redes de alta-velocidade. Nessa seção, algumas aplicações para o esquema de controle proposto são apresentadas.

### 6.10.1 ATM - *Asynchronous Transfer Mode*

O Modo de Transferência Assíncrono (ATM) é uma tecnologia que integra diferentes tipos de tráfegos em uma única rede. As redes ATM foram desenvolvidas para transportar voz, vídeo e dados, deste modo, várias classes de serviços foram definidas para suportar os diferentes tipos de tráfegos com diferentes requisitos de qualidade de serviço (QoS). Dentre as classes definidas, têm-se a Taxa de Bit Constante, ou CBR (*Constant Bit Rate*); Taxa de Bit Variável em Tempo Real, ou rt-VBR (*real-time Variable Bit Rate*); Taxa de Bit Variável em Tempo Não-Real, ou nrt-VBR (*nonreal time Variable Bit Rate*); Taxa de Bit Disponível, ou ABR (*Available Bit Rate*) e Taxa de Bits Não-especificado, ou UBR (*Unspecified Bit Rate*). O tráfego não-controlável  $\nu(k)$  é a união dos tráfegos CBR, rt-VBR e nrt-VBR e o tráfego controlável  $\mu(k)$  corresponde ao tráfego ABR, o qual utiliza a largura de banda disponível no enlace de saída. A taxa de transmissão desse tipo de conexão não pode ser determinada na fase de configuração da conexão, logo é necessário um mecanismo de gerenciamento para controlar adaptativamente a taxa de transmissão das fontes ABR através de um laço fechado [28].

Fazendo uma correspondência dos tipos de tráfegos das redes ATM com os tipos de tráfego definidos no esquema de controle de congestionamento proposto, temos que o tráfego ABR é similar ao tráfego TBD, e os tráfegos CBR e VBR correspondem aos tráfegos TBC e TBV, respectivamente. No controle de congestionamento de redes ATM fim-a-fim, a fonte ABR inicia enviando células com uma taxa inicial conhecida como ICR (*Initial Cell Rate*) e, nos instantes subsequentes, o tráfego é regulado através da informação de status da rede oriundo das células RM (*Resource Management*). Cada célula RM trafega no sentido direto (fonte-destino) e reverso (destino-fonte) do fluxo. No caminho reverso, essas células trazem a informação da largura de banda disponível e a fonte ABR ajusta sua taxa de acordo com essa informação até que novas células RM's sejam recebidas. A informação da largura de banda disponível para a fonte ABR é transportada no campo ER (*Explicit Rate*) da célula RM [66].

### 6.10.2 MPLS - *Multiprotocol Label Switching*

Em um ambiente de redes MPLS, os fluxos de tráfego que pertencem à mesma classe de serviço e compartilham os mesmos requisitos de transporte, recebendo o mesmo tratamento no trajeto para o destino, são mapeados através de rótulos (*labels*) em uma Classe Equivalente de Encaminhamento, ou FEC (*Forward Equivalent Class*). A designação de um pacote particular à uma FEC particular é feita apenas uma vez (quando o pacote entra no domínio MPLS) pelo roteador de ingresso do domínio da rede [67]. Esse procedimento de mapeamento e encaminhamento dos pacotes é realizado pelo Protocolo de Distribuição de Rótulos, ou LDP (*Label Distribution Protocol*), o qual distribui as associações Rótulo/FEC entre os equipamentos da rede MPLS chamados de Roteadores de Comutação

por Rótulos, ou LSR (*Label Switching Router*). Depois que os rótulos são distribuídos, o Caminho Comutado por Rótulo, ou LSP (*Label Switching Path*), é estabelecido como uma sequência ordenada de nós MPLS construída para encaminhar os pacotes encapsulados da FEC correspondente, visto que cada LSP é associado a uma FEC. A Figura 6.21 mostra os principais elementos que compõem uma rede MPLS.

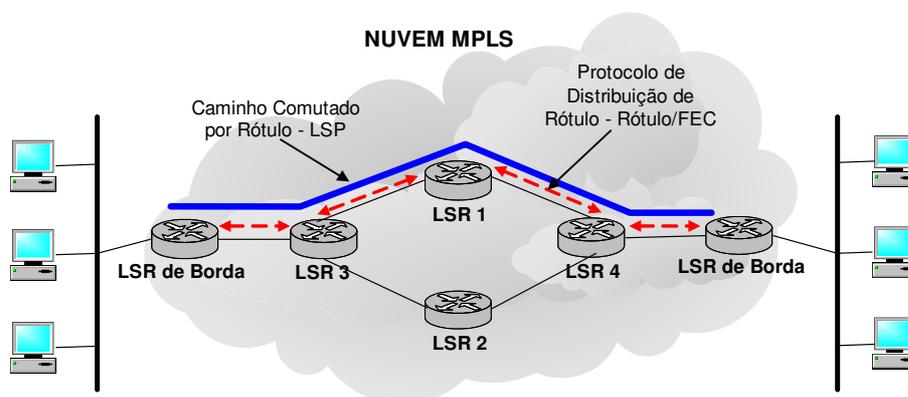


Fig. 6.21: Estrutura Básica de uma Rede MPLS.

Para a implantação de Engenharia de Tráfego em redes MPLS, através do compartilhamento da carga entre vários caminhos visando a garantia de QoS, é necessário criar Rotas Explícitas, nas quais o LSR de ingresso especifica todos os outros LRS's que compõem a rota do LSP. Isto é feito através do Roteamento Baseado em Restrições (CBR - *Constraint Based Routing*), o qual determina os caminhos para os LSP's levando em conta restrições impostas aos elementos de rede.

Essas restrições são baseadas nos atributos de tráfego, tais como, largura de banda requerida, afinidade, prioridade e outros; e nos atributos associados aos recursos da rede, tais como largura de banda disponível do enlace, afinidade do enlace e outros [68] [69]. Podemos, assim, considerar que os atributos de tráfego associados à prioridade dividem o tráfego em dois tipos: o tráfego não-controlável  $\nu(k)$  que tem maior prioridade e é sensível ao atraso, (usado para transmitir vídeo e áudio em tempo real ou não), e o tráfego controlável  $\mu(k)$  que tem menor prioridade e é insensível ao atraso e de tempo não-real. Para os atributos associados aos recursos da rede, é definido que a fonte de tráfego de maior prioridade terá sempre a largura de banda necessária para transmitir seus dados enquanto a de menor prioridade necessitará adaptar sua taxa ao que está disponível. Para o CBR ter acesso às informações necessárias sobre o estado da rede e determinar os caminhos LSP's sujeitos à restrições, deve haver algum mecanismo para disseminação destas informações pela rede. É, então, proposto a extensão do protocolo de roteamento IGP (*Internet Gateway Protocol*) baseado no estado dos enlaces (*link state*), na qual informações adicionais são incluídas, tais como, banda máxima reservável, atributo de classe do recurso e banda reservada. Exemplos de protocolos IGP's modificados são OSPF-TE e IS-IS-TE.

## 6.11 Considerações Finais

Nesse capítulo, o esquema de controle de congestionamento proposto foi avaliado através de algumas medidas de desempenho, tais como, vazão da taxa controlável, utilização do enlace e taxa de perda de bytes. Comparações são realizadas com outros métodos de congestionamento encontrados na literatura. O esquema de controle de congestionamento proposto prediz o comportamento do sistema, mais especificamente, o tamanho da fila no *buffer*, e através dos parâmetros obtidos na predição, calcula uma taxa de controle ótima para regular a fonte controlável  $\mu(k)$ . Para a análise do esquema de controle de congestionamento proposto, vários cenários de simulações foram especificados e analisados. Dessas análises, foi verificado que em todos os nós a taxa de perda de bytes (TPB) foi nula e a média e variância do tamanho da fila no *buffer* diminuíram consideravelmente. Através da comparação entre o método de controle proposto e os outros métodos de controle de congestionamento, constatou-se que o modelo proposto foi o que obteve melhores resultados de desempenho e que manteve o tamanho da fila no *buffer* em torno no nível desejado  $b^T$ .

Da análise do controle de congestionamento proposto com o uso traços de tráfego monofractais e multifractais, pode-se concluir que o controle proposto obteve resultados expressivos ao utilizar os dois tipos de traços de tráfego. A verificação foi realizada com a aplicação do controle proposto em um nó utilizando traços multifractais (Etapa 1 da Análise), com a aplicação do controle proposto em um cenário com 3 nós que utiliza traços monofractais e multifractais (Etapa 2 da Análise), com a aplicação do controle em um nó que utiliza apenas traços monofractais (Etapa 3 da Análise) e em um cenário com 3 nós que utiliza apenas traços multifractais (Etapa 4 da Análise). Em todas as etapas, com exceção da etapa que utilizou traços de tráfego monofractais (Etapa 3), o controle proposto obteve melhorias de desempenho (TPB nula e variância da fila mínima) em função de um pequeno decremento na utilização do enlace e da vazão da taxa controlável. Nessa etapa 3, foi verificado um controle mais ativo e um pequeno aumento na utilização do enlace.

A predição do tamanho da fila no *buffer* foi analisada em dois casos: utilizando apenas traços multifractais e, outro, utilizando apenas traços monofractais. Dos resultados foi possível observar que o preditor proposto obteve melhores resultados de desempenho de predição, em ambos os casos, quando comparado aos preditores RLS e LMS.

Uma análise do fator de ponderação  $\lambda$  no desempenho do sistema foi feita e dos resultados pode-se concluir que a variância do tamanho da fila no *buffer* diminui a medida que tal parâmetro é incrementado. Uma consequência disso é a obtenção de uma taxa ótima de controle mais lenta e suave, comprovando a importância do parâmetro  $\lambda$  e do correto ajuste de seu valor para se ter uma melhor resposta do sistema de controle.

Por fim, algumas aplicações reais do controle de congestionamento proposto em redes de alta-velocidade, tais como ATM e MPLS foram detalhadas e alguns procedimentos foram comentados.

# Capítulo 7

## Conclusões

A integração dos vários tipos de serviços, tais como voz, áudio e vídeo, nas redes de comunicações, traz consigo a necessidade de mecanismos de gerenciamento e controle de congestionamento do tráfego devido às rígidas exigências desses serviços em termos de vazão, perda e atraso. O uso de mecanismos dinâmicos e preventivos requer que as características dos fluxos de tráfego sejam previstas. Contudo, tal previsão encontra como obstáculo o fato de que os fluxos de tráfego possuem características altamente irregulares e complexas. O presente trabalho apresentou um esquema de controle de fluxos adaptativo baseado na modelagem *fuzzy* TSK, a qual tem como vantagem a capacidade de manipular informações vagas (*fuzzy*) obtidas em um ambiente de incerteza e imprecisão. A modelagem TSK foi utilizada em sua versão preditiva com o intuito de prever o comportamento dos fluxos de tráfego, mais especificamente do tamanho da fila no *buffer*.

São duas as principais contribuições desse trabalho: a primeira é a proposta de um preditor adaptativo baseado na modelagem *fuzzy* TSK, cujo algoritmo de treinamento é dividido em dois estágios. No primeiro estágio, os parâmetros das partes antecedentes e consequentes das regras nebulosas são obtidos de forma adaptativa. Além disso, o cálculo dos parâmetros antecedentes apresenta característica recursiva, ou seja, é utilizado a amostra anterior para o cálculo da amostra atual. No segundo estágio da previsão, um algoritmo de refinamento baseado no gradiente descendente é aplicado para ajustar os parâmetros iniciais do modelo obtido no estágio anterior. A segunda contribuição encontra-se no cálculo de uma taxa ótima para regular a taxa das fontes controláveis. Tal cálculo utiliza os parâmetros do modelo *fuzzy* TSK preditivo e um valor desejado para o tamanho da fila no *buffer*.

Análises comparativas entre o preditor *fuzzy* TSK proposto e outros preditores adaptativos mostraram que a aplicação da lógica *fuzzy* na modelagem de fluxos de tráfego proporcionou ganhos de desempenho de previsão e resultados expressivos das medidas de erro analisadas (EQM, EQMN1 e EQMN2). Os traços de tráfego utilizados foram caracterizados de acordo com o seu comportamento fractal utilizando o diagrama multiescala linear e, nas análises da previsão, foram utilizados traços

considerados monofractais e multifractais. O esquema de controle de congestionamento proposto foi avaliado sob vários cenários de simulação, variando-se o número de nós na rede e os traços de tráfego utilizados quanto ao seu comportamento fractal. Mostramos, portanto, que o esquema de controle proposto é válido tanto para tráfego monofractal quanto para tráfego multifractal. Comparações entre o método de controle de fluxos proposto e outros métodos utilizados na literatura mostraram que o método proposto apresentou, em todos os cenários considerados, melhores resultados de desempenho, tais como, taxa de perda de byte nula e uma menor variância do tamanho da fila no *buffer* em função de um pequeno decremento da utilização do enlace e da vazão da taxa controlável, possibilitando ganhos nos parâmetros de qualidade de serviço das redes. Desse modo, temos que o controle de fluxos proposto visa controlar a taxa da fonte de tal forma que a variância do tamanho da fila no *buffer* se torne mínima possível e o tamanho da fila fique em torno do nível desejado  $b^r$  obtendo assim perdas de bytes nula.

Uma das possíveis extensões para esse trabalho é a análise do modelo *fuzzy* TSK preditivo proposto quanto ao retardo dos pacotes na rede dos cenários considerados e quanto a complexidade computacional exigida. Outra possível extensão seria a utilização de mais classes de tráfego no modelo de controle de congestionamento proposto, visto que no modelo atual utilizamos duas classes: a classe de tráfego controlável e a de tráfego não-controlável. Também como trabalho futuro, seria interessante verificar a influência da variação dos parâmetros multifractais ( $\zeta_q$  e  $h_q$ ) no desempenho do controlador.

# Referências Bibliográficas

- [1] C. Q. Yang, A. V. S. Reddy. A Taxonomy for Congestion Control Algorithms in Packet Switching Networks. *IEEE Network Magazine*, 9:34–45, Jul.-Aug. 1995.
- [2] D. O. Awduche. MPLS and Traffic Engineering in IP Networks. *IEEE Communications Magazine*, pages 42–47, Dec. 1999.
- [3] R. Jain. Congestion Control in Computer Networks: Issues and Trends. *IEEE Network Magazine*, pages 24–30, May 1990.
- [4] V. Jacobson. Congestion Avoidance and Control. *ACM SIGCOMM Computer Communication Review*, 25:157–187, 1995.
- [5] R. Jain. Myths about Congestion Management in High Speed Networks. *Internetworking: Research and Experience*, 3:101–113, 1992.
- [6] C. Lefelhocz, B. Lyles, S. Shenker, L. Zhang. Congestion Control for Best-Effort Service: Why We Need a New Paradigm. *IEEE Network*, 10:40–48, Jan. 1996.
- [7] B. S. Chen, S. C. Peng K. C. Wang. Traffic Modeling, Prediction and Congestion Control for High-Speed Networks: A Fuzzy AR Approach. *IEEE Trans. on Fuzzy Systems*, 8(5):491–508, Oct. 2000.
- [8] T. Takagi, M. Sugeno. Fuzzy Identification of Systems and Its Applications to Modeling and Control. *IEEE Trans. Syst. Man and Cybernetics*, SMC-15:116–132, Jan.-Feb. 1985.
- [9] M. Sugeno, T. Yasukawa. A Fuzzy-Logic-Based Approach to Qualitative Modeling. *IEEE Trans. Fuzzy Systems*, 1:7–31, Feb. 1993.
- [10] C. C. Lee. Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part II. *IEEE Trans. on Systems, Man and Cybernetics*, 20(2):419–435, Mar. 1990.

- [11] L. Baochun, K.Nahrstedt. A Control-Based Middleware Framework for Quality of Service Adaptations. *IEEE Journal on Selected Areas in Communications*, 17(9):1632–1650, Sep. 1999.
- [12] A. Vasilakos, K. Anagnostakis. Evolutionary-Fuzzy Prediction for Strategic Inter-Domain Routing: Architecture and Mechanisms. in *IEEE World Conference on Computational Intelligence (WCCI 98)*, May 1998.
- [13] R.Q.Hu , D.W.Petr. A Predictive Self-tuning Fuzzy-logic Feedback Rate Controller. *IEEE/ACM Transactions on Networking*, 8(6):697–709, Dec. 2000.
- [14] ITU-T Rec. I.371. Traffic Control and Congestion Control in B-ISDN. Perth, U.K., Nov. 1995.
- [15] J. Widmer, R. Denda, M. Mauve. A Survey on TCP-Friendly Congestion Control. *IEEE Network*, 15:28–37, May 2001.
- [16] R. Rejaie, M. Handley, D. Estrin. RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet. *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies.*, 3:1337–1345, Mar 1999.
- [17] I. Rhee, V. Ozdemir, Y. Yi. TEAR: TCP Emulation at Receivers - Flow Control for Multimedia Streaming. *NSF CAREER ANI-9875651*, April 2000.
- [18] J. Padhye J. Widmer M. Handley, S. Floyd. TCP Friendly Rate Control (TFRC): Protocol Specification. *RFC 3448*, 2003.
- [19] M. Handley J. Widmer. TCP-Friendly Multicast Congestion Control (TFMCC): Protocol Specification. *IETF RFC 4654*, Aug. 2006.
- [20] P. Gevros, J. Crowcroft, P. Kirstein, S. Bhatti. Congestion Control Mechanisms and the Best Effort Service Model. *IEEE Network*, 15:16–26, May 2001.
- [21] K. Ramakrishnan, S. Floyd. A Proposal to Add Explicit Congestion Notification (ECN) to IP. *IETF RFC 2481*, Jan. 1999.
- [22] J. Postel. Internet Control Message Protocol. *IETF RFC 792*, Sept. 1981.
- [23] G. Huston. *Internet Performance Survival Guide*. John Wiley e Sons, Feb 2000.
- [24] A. Croll, E. Packman. *Managing Bandwidth: Deploying QoS in Enterprise Networks*. Prentice Hall, 1st edition, July 1999.

- [25] B. Braden et al. Recommendations on Queue Management and Congestion Avoidance in the Internet. *IETF RFC 2309*, April 1998.
- [26] S. Floyd, V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.
- [27] ATM Forum. ATM User-Network Interface Specification Version 3.1. *Prentice Hall*, 1994.
- [28] T. M. Chen, S. S. Liu, V. K. Salamam. The Available Bit Rate Service for Data in ATM Networks. *IEEE Communications Magazine*, 34(5):56–58, 63–71, May 1996.
- [29] ATM Forum. Traffic Management Specification Version 4.0. *ATM Forum/95-0013R10*, Feb. 1995.
- [30] D. Dubois, H. Prade. Fuzzy Sets in Approximate Reasoning, Part 1: Inference with Possibility Distributions. *Fuzzy Sets and Systems*, 100:73–132, 1999.
- [31] L. A. Zadeh. Fuzzy Sets. *Information and Control* 8, pages 338–353, 1965.
- [32] H. J. Zimmermann. *Fuzzy Set Theory and Its Applications*. Kluwer Boston, 4th edition, 2001.
- [33] I. B. Turksen. Measurement of Membership Functions and their Acquisitions. *Fuzzy Sets and Systems*, 40:5–38, 1991.
- [34] D. Dubois. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York, 1980.
- [35] D. Dubois, H. Prade. A Class of Fuzzy Measures Based on Triangular Norms. *International Journal of General Systems* 8, pages 43–61, 1982.
- [36] H. Prade D. Dubois. Review of fuzzy set aggregation connectives. *Information Science*, 36(1-2):85–121, 1985.
- [37] L. X. Wang. *Adaptive Fuzzy Systems and Control*. Prentice Hall, 1994.
- [38] E. H. Mamdani. Applications of Fuzzy Algorithms for Simple Dynamic Plant. *Proc. IEEE*, 121(12):1585–1588, 1974.
- [39] E. H. Mamdani P. J. King. The Application of Fuzzy Control Systems to Industrial Processes. *Automática*, 13:235–242, 1977.
- [40] A. Adas. Traffic Models in Broadband Networks. *IEEE Communications Magazine*, pages 82–89, July 1997.

- [41] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, 4th edition, Sept. 2001.
- [42] R. J. Hathaway, J. C. Bezdek. Switching Regression Models and Fuzzy Clustering. *IEEE Trans. Fuzzy Systems*, 1(3):195–204, Aug. 1993.
- [43] H. J. Bortolossi. *Cálculo Diferencial a Várias Variáveis - Uma Introdução A Teoria de Otimização*. Loyola, 2002.
- [44] A. Papoulis, S. Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, New York, fourth edition, 2002.
- [45] B. B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman and Company, 1982.
- [46] W.E. Leland, M.S. Taqqu, W. Willinger, D.V. Wilson. On The Self-similar Nature of Ethernet Traffic (Extended Version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, Feb. 1994.
- [47] J. Beran, R. Sherman, M.S. Taqqu, W. Willinger. Long-Range Dependence in Variable-Bit-Rate Video Traffic. *IEEE Transactions on Communications*, 43(234):1566–1579, Feb./Mar./Apr. 1995.
- [48] V. Paxson, S. Floyd. Wide Area Traffic: The Failure of Poisson Modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, Jun 1995.
- [49] M.E. Crovella, A. Bestavros. Self-similarity in World Wide Web Traffic: Evidence and Possible Causes. pages 160–169, 1996.
- [50] I. Norros. A Storage Model with Self-Similar Input. *Queueing Systems*, 16(3-4):387–396, Sept. 1994.
- [51] I. Norros. On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks. *IEEE Journal of Selected Areas in Communications*, 13(6):953–962, Aug. 1995.
- [52] R. Ried, J. Levy-Vehel. TCP Traffic is Multifractal: A Numerical Study. *INRIA Research Report*, 3129, Oct. 1997.
- [53] A. Feldmann, A.C. Gilbert, W. Willinger, T.G. Kurtz. The Changing Nature of Network Traffic: Scaling Phenomena. *ACM SIGCOMM Computer Communication Review*, 28(2):5–29, April 1998.
- [54] A. Feldmann, A. Gilbert, W. Willinger. Data Networks as Cascades: Explaining the Multifractal Nature of Internet WAN Traffic. *Computer Communication Review*, Sept. 1998.

- [55] Z.L. Zhang, V.J.Ribeiro, S.Moon, C.Diot. Small-Time Scaling Behaviors of Internet Backbone Traffic: An Empirical Study. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies.*, 3:1826–1836, Apr. 2003.
- [56] Multiscaling Page. Disponível em: [http://www.cubinlab.ee.unimelb.edu.au/darryl/MS\\_code.html](http://www.cubinlab.ee.unimelb.edu.au/darryl/MS_code.html). *Multiscaling Page*, Algoritmo de Análise Multifractal.
- [57] J. Mogul. *Traços de Tráfego WAN entre a Digital Equipment Corporation e o Resto do Mundo*. Internet Traffic Archive, disponíveis para download em <http://ita.ee.lbl.gov/html/contrib/DEC-PKT.html>.
- [58] A. Erramilli, O.Narayan, A.Neidhardt, I.Saniee. Performance Impacts of Multi-Scaling in Wide Area TCP/IP Traffic. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings.*, 1:352–359, 2000.
- [59] W. Leland, D. Wilson. *Traço de Tráfego LAN entre a Bellcore e a Internet*. Internet Traffic Archive, disponíveis para download em <http://ita.ee.lbl.gov/html/contrib/BC.html>.
- [60] P. S. Barreto, R. P. Lemos . ATM Traffic Prediction Using Artificial Neural Networks and Wavelet Transforms . *Lecture Notes in Computer Science - LNCS*, 2094/2001, 2001.
- [61] F. H. T. Vieira, L. L. Lee. Alocação Dinâmica de Taxa de Transmissão em Redes de Pacotes Utilizando Redes Neurais Recorrentes Treinadas com Algoritmos em Tempo Real. *IEEE Latin America*, 1, 2003.
- [62] G. R. Bianchi, F. H. T. Vieira, L. L. Lee. A Novel Network Traffic Predictor Based on Multifractal Characteristic. *IEEE Communications Society: Globecom*, 2004.
- [63] T. D. Dangy, S. Molnár, I. Maricza. Queueing Performance Estimation for General Multifractal Traffic. *International Journal of Communication Systems*, 16:117–136, Feb. 2003.
- [64] G. Banerjee, R. Rosenberry, D. Sidhu. Performance Analysis of IP Switching and Tag Switching . *Lecture Notes in Computer Science - LNCS*, 1815/2000:752–762, 2000.
- [65] I. W. Habib, T. N. Saadawi. Access Flow Control Algorithms in Broadband Networks. *Military Communications Conference*, 1:252–256, Oct. 1992.
- [66] ATM Forum. ATM Forum Technical Committee and Traffic Management Working Group. *ATM Forum Traffic Management Specification/96-0056.000*, Apr 1996.

- [67] E. Rosen, A. Viswanathan, R. Callon. Multi-Protocol Label Switching Architecture. *RFC 3031*, Jan. 2001.
- [68] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManus. Requirements for Traffic Engineering over MPLS. *RFC 2702*, Sept. 1999.
- [69] X. Xiao, A. Hannan, B. Bailey, L.M. Ni. Traffic Engineering with MPLS in the Internet. *IEEE Network*, 14(2):28–33, Mar/Apr 2000.

# Apêndice A

## Cálculo Recursivo do Vetor de Parâmetros Consequentes $\vec{a}^i$

Sabe-se que, o vetor de parâmetros  $\vec{a}^i$  das partes consequentes da  $i$ -ésima regra para o  $k$ -ésimo padrão de treinamento é dado por:

$$\vec{a}^i(k) = P_i(k)X^T(k)Q_i(k)Y(k) \quad (\text{A.1})$$

Logo, o vetor dos parâmetros consequentes  $\vec{a}^i$  para o  $(k + 1)$ -ésimo padrão de treinamento é:

$$\vec{a}^i(k + 1) = P_i(k + 1)X^T(k + 1)Q_i(k + 1)Y(k + 1) \quad (\text{A.2})$$

Note que o vetor  $\vec{a}^i(k + 1)$  pode ser escrito como:

$$\vec{a}^i(k + 1) = P_i(k + 1) \begin{pmatrix} X(k) \\ x(k + 1) \end{pmatrix}^T \begin{bmatrix} Q_i(k) & 0 \\ 0 & \omega(k + 1) \end{bmatrix} \begin{pmatrix} Y(k) \\ y(k + 1) \end{pmatrix}, \quad (\text{A.3})$$

$$\vec{a}^i(k + 1) = P_i(k + 1)[X^T(k)Q_i(k) + x^T(k + 1)\omega(k + 1)] \begin{pmatrix} Y(k) \\ y(k + 1) \end{pmatrix}, \quad (\text{A.4})$$

$$\vec{a}^i(k + 1) = P_i(k + 1)[X^T(k)Q_i(k)Y(k) + x^T(k + 1)\omega(k + 1)y(k + 1)]. \quad (\text{A.5})$$

Sabe-se que, da equação (A.1),  $X^T(k)Q_i(k)Y(k) = P_i^{-1}(k)\vec{a}^i(k)$ . Então, substituindo este resultado em (A.5), temos que:

$$\vec{a}^i(k + 1) = P_i(k + 1)[P_i^{-1}(k)\vec{a}^i(k) + x^T(k + 1)\omega(k + 1)y(k + 1)]. \quad (\text{A.6})$$

Adicionando e subtraindo  $\bar{a}^i(k)$  em ambos os lados da equação (A.6), temos:

$$\bar{a}^i(k+1) = \bar{a}^i(k) + [P_i(k+1)P_i^{-1}(k)\bar{a}^i(k) + P_i(k+1)x^T(k+1)\omega(k+1)y(k+1)] - \bar{a}^i(k), \quad (\text{A.7})$$

$$\bar{a}^i(k+1) = \bar{a}^i(k) + P_i(k+1)P_i^{-1}(k)\bar{a}^i(k) - \bar{a}^i(k) + P_i(k+1)x^T(k+1)\omega(k+1)y(k+1), \quad (\text{A.8})$$

$$\bar{a}^i(k+1) = \bar{a}^i(k) + (P_i(k+1)P_i^{-1}(k) - I)\bar{a}^i(k) + P_i(k+1)x^T(k+1)\omega(k+1)y(k+1). \quad (\text{A.9})$$

Sabe-se que  $P_i(k) = (X^T(k)Q_i(k)X(k))^{-1}$ , então a matriz de covariância  $P_i(k+1)$  é dada por:

$$P_i(k+1) = (X^T(k+1)Q_i(k+1)X(k+1))^{-1} \quad (\text{A.10})$$

O vetor  $P_i(k+1)$  pode ser escrito como:

$$P_i(k+1) = \left( \begin{pmatrix} X(k) \\ x(k+1) \end{pmatrix}^T \begin{bmatrix} Q_i(k) & 0 \\ 0 & \omega(k+1) \end{bmatrix} \begin{pmatrix} X(k) \\ x(k+1) \end{pmatrix} \right)^{-1} \quad (\text{A.11})$$

$$P_i(k+1) = \left( [X^T(k)Q_i(k) + x^T(k+1)\omega(k+1)] \begin{pmatrix} X(k) \\ x(k+1) \end{pmatrix} \right)^{-1} \quad (\text{A.12})$$

$$P_i(k+1) = (X^T(k)Q_i(k)X(k) + x^T(k+1)\omega(k+1)x(k+1))^{-1} \quad (\text{A.13})$$

Substituindo  $X^T(k)Q_i(k)X(k) = P_i^{-1}(k)$  na equação A.13 temos:

$$P_i(k+1) = (P_i^{-1}(k) + x^T(k+1)\omega(k+1)x(k+1))^{-1} \quad (\text{A.14})$$

Fazendo a inversão dos termos em ambos os lados da equação (A.14), obtém-se:

$$P_i^{-1}(k+1) = P_i^{-1}(k) + x^T(k+1)\omega(k+1)x(k+1) \quad (\text{A.15})$$

$$P_i^{-1}(k) = P_i^{-1}(k+1) - x^T(k+1)\omega(k+1)x(k+1) \quad (\text{A.16})$$

Substituindo (A.16) em (A.9) temos:

$$\begin{aligned} \bar{a}^i(k+1) &= \bar{a}^i(k) + (P_i(k+1) (P_i^{-1}(k+1) - x^T(k+1)\omega(k+1)x(k+1)) - I) \bar{a}^i(k) + \\ &+ P_i(k+1)x^T(k+1)\omega(k+1)y(k+1) \end{aligned} \quad (\text{A.17})$$

$$\bar{a}^i(k+1) = \bar{a}^i(k) + (I - P_i(k+1)x^T(k+1)\omega(k+1)x(k+1) - I) \bar{a}^i(k) +$$

$$+P_i(k+1)x^T(k+1)\omega(k+1)y(k+1) \quad (\text{A.18})$$

$$\vec{a}^i(k+1) = \vec{a}^i(k) - P_i(k+1)x^T(k+1)\omega(k+1)x(k+1)\vec{a}^i(k) + P_i(k+1)x^T(k+1)\omega(k+1)y(k+1) \quad (\text{A.19})$$

Finalmente, obtemos o vetor dos parâmetros consequentes recursivo, dado por:

$$\vec{a}^i(k+1) = \vec{a}^i(k) + P_i(k+1)x^T(k+1)\omega(k+1)(y(k+1) - x(k+1)\vec{a}^i(k)) \quad (\text{A.20})$$

O vetor  $\vec{a}^i(k+1)$  é calculado utilizando  $P_i(k+1)$ , o qual é dado pela equação (A.14). Entretanto, conforme citado anteriormente, a inversão matricial de  $P_i(k)$  adiciona um custo computacional muito alto. Para solucionar esse problema, usamos o teorema da inversão matricial [41] cujo desenvolvimento matemático está detalhado no Apêndice B. Assim, a equação (A.14) pode ser reescrita como:

$$P_i(k+1) = P_i(k) - P_i(k)x^T(k+1)(1/\omega(k+1) + x(k+1)P_i(k)x^T(k+1))^{-1}x(k+1)P_i(k) \quad (\text{A.21})$$

$$P_i(k+1) = P_i(k) - P_i(k)x^T(k+1)\left(\frac{1 + \omega(k+1)x(k+1)P_i(k)x^T(k+1)}{\omega(k+1)}\right)^{-1}x(k+1)P_i(k) \quad (\text{A.22})$$

Logo,  $P_i(k+1)$  é dado por:

$$P_i(k+1) = P_i(k) - \left(\frac{P_i(k)x^T(k+1)\omega(k+1)x(k+1)P_i(k)}{1 + \omega(k+1)x(k+1)P_i(k)x^T(k+1)}\right) \quad (\text{A.23})$$

em que  $x(k+1)$  é a  $(k+1)$ -ésima linha da matriz  $X(k)$  e  $w(k+1)$  é o  $(k+1)$ -ésimo elemento da matriz diagonal  $Q_i(k+1)$ .



# Apêndice B

## Teorema da Inversão Matricial

Sejam  $A$  e  $B$  duas matrizes positivas de tamanho  $M \times M$  relacionadas da seguinte forma:

$$A = B^{-1} + CD^{-1}C^H \quad (\text{B.1})$$

em que  $D$  é uma matriz positiva  $N \times M$  e  $C$  é uma matriz  $M \times N$ . De acordo com o teorema da inversão matricial, é possível expressar a inversa de  $A$  da seguinte forma:

$$A^{-1} = B - BC(D + C^H BC)^{-1}C^H B \quad (\text{B.2})$$

A prova deste teorema é estabelecida multiplicando as equações (B.1) e (B.2) e reconhecendo que o produto de uma matriz quadrada pela sua inversa é igual a matriz identidade.

### B.1 Desenvolvimento Matemático da Equação da Matriz de Covariância $P_i(k+1)$ Equação (A.14)

A equação (A.14) é dada por:

$$P_i(k+1) = (P_i^{-1}(k) + x^T(k+1)\omega(k+1)x(k+1))^{-1} \quad (\text{B.3})$$

Fazendo a inversa em ambos os lados da equação (B.3), tem-se que:

$$P_i^{-1}(k+1) = P_i^{-1}(k) + x^T(k+1)\omega(k+1)x(k+1) \quad (\text{B.4})$$

Aplicando o teorema da Inversão Matricial e fazendo  $A = P_i^{-1}(k+1)$ ,  $B = P_i(k)$ ,  $C = x^T(k+1)$  e  $D = w^{-1}(k+1)$ , temos que, pela equação (B.2):

$$P_i(k+1) = P_i(k) - P_i(k)x^T(k+1) \left(1/\omega(k+1) + x(k+1)P_i(k)x^T(k+1)\right)^{-1} x(k+1)P_i(k) \quad (\text{B.5})$$

## B.2 Desenvolvimento Matemático da Equação da Matriz de Correlação Cruzada $\Phi(k)$ Equação (5.37)

A equação (5.37) é escrita como:

$$\Phi(k) = \lambda\Phi(k-1) + U(k)U^H(k) \quad (\text{B.6})$$

Fazendo  $A = \Phi(k)$ ,  $B = (\lambda\Phi(k-1))^{-1}$ ,  $C = U(k)$  e  $D = 1$  e aplicando o teorema da Inversão Matricial temos que pela equação (B.2):

$$\Phi^{-1}(k) = \lambda^{-1}\Phi^{-1}(k-1) - \lambda^{-1}\Phi^{-1}(k-1)U(k)(1+U^H(k)\lambda^{-1}\Phi^{-1}(k-1)U(k))^{-1}U^H(k)\lambda^{-1}\Phi^{-1}(k-1) \quad (\text{B.7})$$

$$\Phi^{-1}(k) = \lambda^{-1}\Phi^{-1}(k-1) - \frac{\lambda^{-1}\Phi^{-1}(k-1)U(k)U^H(k)\lambda^{-1}\Phi^{-1}(k-1)}{1+U^H(k)\lambda^{-1}\Phi^{-1}(k-1)U(k)} \quad (\text{B.8})$$

$$\Phi^{-1}(k) = \lambda^{-1}\Phi^{-1}(k-1) - \frac{\lambda^{-2}\Phi^{-1}(k-1)U(k)U^H(k)\Phi^{-1}(k-1)}{1+\lambda^{-1}U^H(k)\Phi^{-1}(k-1)U(k)} \quad (\text{B.9})$$

## Apêndice C

# Cálculo das Equações de Ajuste dos Parâmetros das Partes Antecedentes e Consequentes

### Equação I (Ajuste da Parte Antecedente)

Os parâmetros das partes premissas do modelo nebuloso  $\theta_{jl}^i$  com  $l = 1$  e  $2$ , podem ser ajustados precisamente através da equação abaixo:

$$\Delta\theta_{jl}^i(k) = \eta(y(k) - \hat{y}(k))(y^i(k) - \hat{y}(k)) \frac{1}{\sum_{i=1}^C w^i(k)} \frac{\partial w^i(k)}{\partial \theta_{jl}^i(k)} \quad (\text{C.1})$$

em que  $\eta$  é uma taxa de de aprendizagem,  $y(k)$  é a saída desejada,  $\hat{y}(k)$  é a saída do modelo nebuloso TSK e  $y^i(k)$  é a saída da  $i$ -ésima regra do modelo TSK.

*Prova:* Para um dado conjunto de observações  $[\vec{x}(k), y(k)]$ , definimos como erro  $e$ , a diferença entre a saída desejada  $y(k)$  e a saída do modelo TSK  $\hat{y}(k)$ :

$$e = y(k) - \hat{y}(k) = y(k) - \frac{\sum_{i=1}^C y^i(k)w^i(k)}{\sum_{i=1}^C w^i(k)} \quad (\text{C.2})$$

As partes premissas devem ser atualizadas de tal forma a minimizar o erro quadrático  $e^2$ . Assim, do algoritmo gradiente descendente [41], temos:

$$\Delta\theta_{jl}^i(k) = -\eta \frac{\partial}{\partial \theta_{jl}^i(k)} \left( \frac{e^2}{2} \right)$$

$$\Delta\theta_{jl}^i(k) = -\eta e \frac{\partial e}{\partial \theta_{jl}^i(k)}$$

$$\Delta\theta_{jl}^i(k) = -\eta(y(k) - \hat{y}(k)) \frac{\partial(y(k) - \hat{y}(k))}{\partial \theta_{jl}^i(k)}$$

$$\Delta\theta_{jl}^i(k) = -\eta(y(k) - \hat{y}(k)) \frac{\partial}{\partial \theta_{jl}^i(k)} \left( y(k) - \frac{\sum_{i=1}^C y^i(k) w^i(k)}{\sum_{i=1}^C w^i(k)} \right)$$

$$\begin{aligned} \Delta\theta_{jl}^i(k) = & -\eta(y(k) - \hat{y}(k))(-1) \left( \frac{\partial}{\partial \theta_{jl}^i(k)} \left( \sum_{i=1}^C y^i(k) w^i(k) \right) \times \left( \sum_{i=1}^C w^i(k) \right) - \left( \sum_{i=1}^C y^i(k) w^i(k) \right) \times \right. \\ & \left. \times \frac{\partial}{\partial \theta_{jl}^i(k)} \left( \sum_{i=1}^C w^i(k) \right) \right) / \left( \sum_{i=1}^C w^i(k) \right)^2 \end{aligned}$$

$$\Delta\theta_{jl}^i(k) = \eta(y(k) - \hat{y}(k)) \frac{\frac{\partial w^i(k)}{\partial \theta_{jl}^i(k)} y^i(k) \times \sum_{i=1}^C w^i(k) - \sum_{i=1}^C y^i(k) w^i(k) \times \frac{\partial w^i(k)}{\partial \theta_{jl}^i(k)}}{\left( \sum_{i=1}^C w^i(k) \right)^2}$$

$$\Delta\theta_{jl}^i(k) = \eta(y(k) - \hat{y}(k)) \frac{\partial w^i(k)}{\partial \theta_{jl}^i(k)} \left( \frac{y^i}{\sum_{i=1}^C w^i(k)} - \frac{\sum_{i=1}^C y^i(k) w^i(k)}{\sum_{i=1}^C w^i(k)} \frac{1}{\sum_{i=1}^C w^i(k)} \right)$$

$$\Delta\theta_{jl}^i(k) = \eta(y(k) - \hat{y}(k)) \frac{\partial w^i(k)}{\partial \theta_{jl}^i(k)} \frac{1}{\sum_{i=1}^C w^i(k)} (y^i(k) - \hat{y}(k))$$

$$\Delta\theta_{jl}^i(k) = \eta(y(k) - \hat{y}(k)) (y^i(k) - \hat{y}(k)) \frac{1}{\sum_{i=1}^C w^i(k)} \frac{\partial w^i(k)}{\partial \theta_{jl}^i(k)}$$

Calculando  $\frac{\partial w^i(k)}{\partial \theta_{jl}^i(k)}$ , temos que  $w^i(k)$  é definido como  $\min_{j=1,2,\dots,n} A_j^i(x_j(k))$  e, sendo  $j^*$  igual ao índice  $j$  que define a minimização em  $w^i(k)$ , isto é,  $j^* = \arg \min_j A_j^i(x_j(k))$ .<sup>1</sup> Então, quando  $j = j^*$  teremos, para  $l = 1$ :

$$\frac{\partial w^i(k)}{\partial \theta_{j1}^i(k)} = \frac{\partial A_j^i}{\partial \theta_{j1}^i(k)}$$

$$\frac{\partial w^i(k)}{\partial \theta_{j1}^i(k)} = \frac{\partial}{\partial \theta_{j1}^i(k)} \left( \exp \left\{ - \left( \frac{x_j(k) - \theta_{j1}^i(k)}{\theta_{j2}^i(k)} \right)^2 \right\} \right)$$

$$\frac{\partial w^i(k)}{\partial \theta_{j1}^i(k)} = \exp \left\{ - \left( \frac{x_j(k) - \theta_{j1}^i(k)}{\theta_{j2}^i(k)} \right)^2 \right\} \times \frac{\partial}{\partial \theta_{j1}^i(k)} \left( - \left( \frac{x_j(k) - \theta_{j1}^i(k)}{\theta_{j2}^i(k)} \right)^2 \right)$$

<sup>1</sup>Outra definição seria,  $j$  tal que  $w^i = \min_{j=1,2,\dots,n} A_j^i(x_j(k)) = A_j^i(x_j(k))$ .

$$\frac{\partial w^i(k)}{\partial \theta_{j_1}^i(k)} = \exp \left\{ - \left( \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \right)^2 \right\} \times (-2) \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \times -\frac{1}{\theta_{j_2}^i(k)}$$

$$\frac{\partial w^i(k)}{\partial \theta_{j_1}^i(k)} = \frac{2}{\theta_{j_2}^i(k)} \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \exp \left\{ - \left( \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \right)^2 \right\}$$

E para  $l = 2$ , temos:

$$\frac{\partial w^i(k)}{\partial \theta_{j_2}^i(k)} = \frac{\partial A_j^i}{\partial \theta_{j_2}^i(k)}$$

$$\frac{\partial w^i(k)}{\partial \theta_{j_2}^i(k)} = \exp \left\{ - \left( \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \right)^2 \right\} \times \frac{\partial}{\partial \theta_{j_2}^i(k)} \left( - \left( \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \right)^2 \right)$$

$$\frac{\partial w^i(k)}{\partial \theta_{j_2}^i(k)} = \exp \left\{ - \left( \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \right)^2 \right\} \times (-2) \left( \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \right) \left( \frac{-(x_j(k) - \theta_{j_1}^i(k))}{(\theta_{j_2}^i(k))^2} \right)$$

$$\frac{\partial w^i(k)}{\partial \theta_{j_2}^i(k)} = \frac{2}{\theta_{j_2}^i(k)} \left( \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \right)^2 \exp \left\{ - \left( \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \right)^2 \right\}$$

$$\frac{\partial w^i(k)}{\partial \theta_{j_2}^i(k)} = \frac{x_j(k) - \theta_{j_1}^i(k)}{\theta_{j_2}^i(k)} \frac{\partial w^i(k)}{\partial \theta_{j_1}^i(k)}$$

Quando  $j \neq j^*$ ,  $\frac{\partial w^i(k)}{\partial \theta_{j_2}^i(k)} = \frac{\partial w^i(k)}{\partial \theta_{j_1}^i(k)} = 0$ , o que significa que nenhum ajuste será feito para esses casos.

## Equação II (Ajuste da Parte Consequente)

Os parâmetros das partes consequentes do modelo nebuloso definido em (4.3) podem ser ajustados através da equação:

$$\Delta a_j^i(k) = \zeta (y(k) - \hat{y}(k)) \frac{w^i(k) x_j(k)}{\sum_{i=1}^C w^i(k)} \quad (C.3)$$

em que  $\zeta$  é outra taxa de aprendizagem,  $y(k)$  é a saída desejada e  $\hat{y}(k)$  é a saída do modelo nebuloso TSK.

*Prova:* Da mesma forma que os parâmetros das partes premissas, os parâmetros consequentes são ajustados de tal forma a minimizar o erro quadrático  $e^2$ , onde  $e(k) = y(k) - \hat{y}(k)$ , através do algoritmo do gradiente descendente. Assim, temos:

$$\Delta a_j^i(k) = -\zeta \frac{\partial}{\partial a_j^i(k)} \left( \frac{e^2}{2} \right)$$

$$\Delta a_j^i(k) = -\zeta e \frac{\partial e}{\partial a_j^i(k)}$$

$$\Delta a_j^i(k) = -\zeta (y(k) - \hat{y}(k)) \times \left( -\frac{\hat{y}(k)}{a_j^i(k)} \right)$$

$$\Delta a_j^i(k) = \zeta (y(k) - \hat{y}(k)) \frac{w^i(k)x_j(k)}{\sum_{i=1}^C w^i(k)}$$