
ESQUEMA DE CONTROLE ADAPTATIVO DE TRÁFEGO DE REDES BASEADO EM UM ALGORITMO DE PREDIÇÃO FUZZY

Flávio Henrique Teles Vieira*
flavio@eeec.ufg.br

Lígia Maria Carvalho Sousa †
ligia@decom.fee.unicamp.br

Lee Luan Ling†
lee@decom.fee.unicamp.br

*Escola de Engenharia Elétrica e de Computação, Universidade Federal de Goiás
Av. Universitária, 1488 - Quadra 86 - Bloco A - 74605-010
Setor Leste Universitário - Goiânia - Goiás - Brasil

†Departamento de Comunicações - DECOM - FEEC - UNICAMP
Caixa Postal 6101 CEP 13.083-970
Campinas, SP, Brasil

ABSTRACT

In this work, we propose an adaptive predictive flow control scheme based on the TSK fuzzy model for congestion control in broadband networks. The proposed control scheme intends to avoid congestion by applying a TSK type model to predict the buffer queue length. For this end, we developed an adaptive training algorithm for the TSK model that was incorporated into the proposed control scheme, achieving low loss rate network performance. Our developed fuzzy predictor consists of a two-step algorithm: an adaptive training stage with covariance resetting and a gradiente-based learning algorithm for refining the previous part of the prediction procedure. An evaluation of the proposed predictor is carried out by using real network traffic traces. As essential part of the proposed congestion control scheme, we derive an analytical expression involving the fuzzy model parameters for the control of the flow rates that minimizes the queueing length variance. Further, a network environment is considered and the congestion control is applied to each node of the analyzed

scenery. Simulations are made to compare the performance of the proposed adaptive control algorithm to those of some existing control approaches in literature.

KEYWORDS: Flow Control, Adaptive Prediction, Fuzzy Modeling, Network Traffic.

RESUMO

Neste trabalho, propomos um esquema de controle adaptativo de fluxos de tráfego de redes de computadores baseado no modelo fuzzy Takagi-Sugeno (TSK). O esquema de controle proposto evita a ocorrência de congestionamento utilizando o modelo fuzzy TSK para prever adaptativamente o tamanho da fila no buffer nos enlaces. Com o objetivo de ajustar dinamicamente os parâmetros do modelo fuzzy TSK, desenvolvemos um algoritmo de treinamento adaptativo para o mesmo. Os parâmetros das partes de premissa e das partes consequentes do modelo são atualizados na primeira etapa do algoritmo de treinamento desenvolvido. Já a segunda etapa consiste de um algoritmo de refinamento dos parâmetros do modelo baseado em gradiente descendente. A eficiência do preditor proposto é avaliada com o uso de traços de tráfego reais em comparação a outros preditores. A partir dos parâmetros do modelo fuzzy TSK, deduzimos uma expressão

Artigo submetido em 25/05/2007

1a. Revisão em 17/11/2007

2a. Revisão em 24/02/2008

3a. Revisão em 15/06/2008

Aceito sob recomendação do Editor Associado

Prof. Ivan Nunes Da Silva

para a taxa da fonte de tráfego de modo a minimizar a variância do tamanho de fila no buffer e que é parte essencial do nosso esquema de controle. O controle de congestionamento proposto é então aplicado em um cenário de rede composto de vários nós alimentados com fluxos de tráfego real. Comparações realizadas com outros métodos de controle de congestionamento demonstram que o controle de congestionamento proposto provê menores taxas de perdas, assim como a manutenção da taxa de ocupação no buffer abaixo do valor desejado.

PALAVRAS-CHAVE: Controle de Fluxos, Predição Adaptativa, Modelagem Fuzzy, Tráfego de Redes.

1 INTRODUÇÃO

Aplicações que exigem garantias de qualidade de serviço (QoS) têm sido cada vez mais encontradas na Internet, tais como voz sobre IP e vídeo conferência (Ditze and Jahnich, 2005; Cherry, 2005; Baldi and Ofek, 2000). Devido ao comportamento imprevisível e de rajadas dos fluxos em redes multimídia, congestionamentos podem ocorrer causando perdas de dados (bytes) e degradação dos parâmetros de QoS (Hatano et al., 2007). Desta forma, se faz necessário um mecanismo de controle de congestionamento eficiente para superar esses problemas e garantir a qualidade de serviço desejada.

Duas estratégias distintas de controle e gerenciamento de congestionamento são comumente encontradas na literatura: recuperação e prevenção (Durresti et al., 2006; Jacobson, 1995). A primeira tem como objetivo atuar na rede após a ocorrência de congestionamento para resolver tal problema. A segunda trata de detectar possíveis condições que levem a situações de congestionamentos e de executar procedimentos para impedir suas ocorrências. Esta estratégia de prevenção, que se baseia em antecipar situações de congestionamento, normalmente inclui modelos para análise do comportamento do tráfego, sendo este frequentemente complexo e não-linear (Lee and Fapojuwo, 2005).

A solução adotada neste trabalho está relacionada com a aplicação da modelagem fuzzy visando a predição do comportamento de fila no *buffer* e o controle da taxa de entrada dos fluxos de tráfego. Neste tipo de abordagem, um fator importante para o sucesso do controle de congestionamento é a aplicação de uma modelagem de tráfego adequada. Quanto mais precisa for a estimação do comportamento dos fluxos da rede, mais apropriado será o serviço oferecido ao usuário. Por outro lado, se o modelo não for capaz de representar precisamente o tráfego real, o desempenho real da rede pode ser subestimado ou superestimado. Assim, um modelo de tráfego eficiente deve capturar fielmente as características do tráfego de redes. Muitos estudos mostram que modelos fuzzy

possuem vantagens sobre os modelos lineares em descrever o comportamento não-linear e variante no tempo de processos reais desconhecidos, como é o caso dos fluxos de tráfego de redes (Ouyang et al., 2005; Chen et al., 2000). De fato, a modelagem fuzzy é capaz de representar um sistema complexo não-linear através da combinação de vários modelos locais lineares invariantes no tempo (Chen et al., 2007; Sugeno and Yasukawa, 1993).

Várias técnicas de controle de congestionamento em redes de computadores têm sido propostas na literatura (Wang et al., 2007; Durresti et al., 2006; Nejad et al., 2006; Karnik and Kumar, 2005). Dentre as propostas de controle de congestionamento utilizando lógica nebulosa, algumas utilizam modelos fuzzy sem adaptação dos parâmetros como em Hu and Petr (2000) e outras são baseadas em protocolos ou tecnologias de rede específicos (Nejad et al., 2006; Chen et al., 2003). No primeiro caso, muitos dos esquemas não são suficientemente precisos em prever o comportamento variante do tráfego gerado por aplicações em tempo real devido a não adaptação de seus parâmetros (Pitsillides and Lambert, 1997). Por exemplo, em Hu and Petr (2000), os autores propõem um algoritmo de controle preditivo das taxas dos fluxos de tráfego. Entretanto, nesta abordagem 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. No segundo caso, podemos citar propostas de esquemas de controle que são dedicados aos protocolos de redes ATM (*Asynchronous Transfer Mode*) (Kharraajoo, 2004; Chen et al., 2003) ou baseados nos mecanismos de controle de congestionamento do protocolo TCP/IP (Transmission Control Protocol/Internet Protocol), como em (Wang et al., 2007; Wang et al., 2006; Basar and Srikant, 2005). Entre as propostas de controle de congestionamento que não dependem de mecanismos específicos de rede, destacamos o método de Realimentação Binária (Chen et al., 1996) e o método de Controle Proporcional (Habib and Saadawi, 1992). Tais métodos podem ser utilizados para controle de aplicações de tempo real e são também eficazes para outros problemas de controle (Karnik and Kumar, 2005; Zhang et al., 2000). Assim, usaremos estes métodos nas análises comparativas deste trabalho.

Neste trabalho, objetivamos estabelecer um estratégia de controle de congestionamento cujos parâmetros sejam atualizados adaptativamente e que possa ser aplicada em diferentes tecnologias de rede. Para isso, propomos um algoritmo de treinamento adaptativo para um preditor fuzzy do tipo Takagi-Sugeno (TSK), o qual é aplicado na modelagem e predição do tamanho da fila no *buffer* com o intuito de controlar as taxas das fontes em um enlace. O algoritmo de treinamento proposto (ARFARC - *Agrupamento Regressivo Fuzzy Adaptativo com Reset da Covariância*) consiste

de duas partes principais: um procedimento de treinamento adaptativo baseado em um método de agrupamento (*clustering*) que divide os dados de tráfego em grupos, onde cada grupo é descrito por um modelo fuzzy TSK. Na segunda parte, um algoritmo de aprendizagem baseado na técnica de gradiente descendente é usado para refinar os parâmetros do modelo obtido, melhorando a precisão da modelagem. Por fim, uma taxa ótima de regulação da fonte é calculada através dos parâmetros obtidos na predição.

Na seção 2, descrevemos o sistema onde é aplicado o esquema de controle de congestionamento proposto. Na seção 3, apresentamos o modelo TSK e, em seguida, o algoritmo de treinamento adaptativo proposto. Na seção 4, derivamos uma expressão para a taxa ótima de controle de fluxo em função dos parâmetros do modelo obtido na predição. Na seção 5, o preditor é analisado e comparado a alguns preditores lineares adaptativos existentes na literatura. Na seção 6, o método de controle de congestionamento proposto é comparado com outros esquemas de controle de congestionamento e avaliado segundo algumas medidas de desempenho. Por fim, apresentamos as conclusões obtidas na seção 7.

2 DESCRIÇÃO DO SISTEMA DE CONTROLE DE CONGESTIONAMENTO

Nessa seção, detalhamos o sistema de controle de congestionamento implementado. Este sistema de controle visa prever adaptativamente o comportamento da fila no *buffer* e, a partir dos parâmetros do modelo de predição de tráfego, controlar a taxa da fonte de tal forma que o tamanho da fila no *buffer* seja igual ou inferior ao nível de referência desejado.

A estrutura básica do sistema de controle de congestionamento considerado é mostrada na Figura 1, na qual estão identificados todos os elementos do sistema:

- Fluxo de tráfego controlável $\mu(k)$;
- Fluxo de tráfego não-controlável $\nu(k)$;
- *Buffer*;
- Atraso de realimentação (*round-trip time*) d ;
- Tamanho da fila no *buffer* $b(k)$ no instante k ;
- Capacidade η do enlace de saída do sistema;
- Nível desejado para o tamanho da fila no *buffer* b^* ;
- Bloco funcional que representa o esquema de controle de congestionamento proposto, o qual atua no tráfego controlável regulando sua taxa.

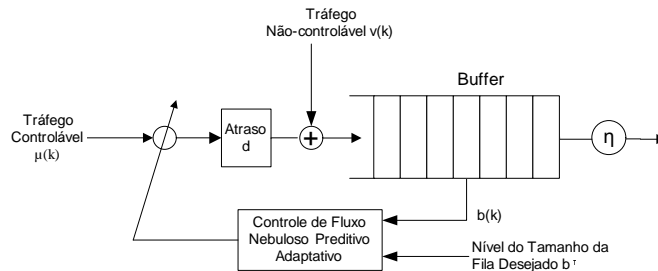


Figura 1: Modelo do Sistema de Controle de Congestionamento.

O intervalo de amostragem do esquema de controle proposto é de T segundos, ou seja, as taxas dos fluxos de tráfego e o tamanho da fila no *buffer* $b(k)$ se modificam a cada T segundos.

O tráfego que chega ao *buffer* vem de duas classes distintas. Uma está relacionada ao tráfego controlável $\mu(k)$ que adapta sua taxa às condições da rede, chamado de tráfego TBD (Taxa de Bits Disponível). A taxa de transmissão do tráfego TBD é regulada pelo controlador fuzzy adaptativo proposto implementado no *switch*. A outra classe é a do tráfego não-controlável $\nu(k)$, no qual se enquadram os serviços do tipo TBC (Taxa de Bits Constante) e TBV (Taxa de Bits Variável). Essa classe de tráfego é sensível ao atraso e tem maior prioridade de uso do enlace. Deste modo, as fontes controláveis podem usufruir apenas a banda restante não usada pelas fontes não-controláveis.

Os serviços de taxa de bits variável (TBV) e constante (TBC) são apropriados para um grande número de aplicações (Lee and Veciana, 2000; Chen et al., 1996). O serviço TBV, cujos fluxos possuem características 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 transmitidos a taxas arbitrárias e entregues com atraso e Taxa de Perda de Bytes (TPB) limitados de acordo com os valores requeridos pela aplicação. O serviço TBC pode ser considerado um caso especial do serviço TBV, onde a taxa de pico é igual a taxa média de serviço. Esses tipos de serviços são usados para aplicações em tempo-real com largura de banda fixa, tais como, em redes chaveadas por circuitos (Exemplo: canal de voz de 64Kbyte/seg). Os pacotes do serviço TBC são gerados em intervalos periódicos regulares e entregues com atrasos estritamente limitados (Assunção and Ghanbari, 2000).

Os serviços de taxa de bits disponível (TBD) são utilizados em uma classe mais estreita de aplicações que podem se adaptar a banda disponível e tolerar atrasos (Imer et al., 2001). Essas conexões TBD em geral compartilham a banda do enlace com os tráfegos de TBV e TBC, quando

houver parte não usada. Das características citadas para o tráfego TBD, podemos perceber que esse serviço não é apropriado para aplicações sensíveis a atrasos.

Dado que a fonte TBD ajusta sua taxa de transmissão periodicamente de acordo com a banda disponível no momento, o desempenho da rede será fortemente afetado pela eficiência do mecanismo de ajuste de taxa. Para atingir um nível adequado de eficiência, a fonte é informada das últimas condições da rede, normalmente através de um canal de realimentação com atraso (Boggia et al., 2007). O atraso de realimentação juntamente com o atraso de transmissão direta é chamado de *round-trip time* (representado pelo atraso d na Figura 1) (Wenshan et al., 2006). Este atraso causa diferenças entre a taxa da fonte TBD e a banda disponível, o que exige a presença de *buffers* na rede para acomodar o tráfego TBD durante os intervalos de tempo em que a taxa da fonte excede a taxa disponível no enlace.

O esquema de controle de tráfego proposto tem como objetivo levar em consideração o atraso de *round-trip* d prevendo o comportamento da ocupação do *buffer* a fim de evitar ocorrência de congestionamento. Para isto, desenvolvemos um algoritmo adaptativo de predição de d passos a frente do tamanho da fila no *buffer* $b(k+d)$ baseado nas informações passadas e atuais da taxa da fonte controlável $\mu(s)_{s \leq k}$ e do tamanho da fila no *buffer* $b(s)_{s \leq k}$. Além disso, para manter o tamanho da fila no *buffer* abaixo do nível desejado b^τ e minimizar a sua variância, aplicamos a taxa ótima de controle proposta $\mu^o(k)$ para regular a taxa da fonte controlável (Seção 4). Deste modo, é possível confinar a taxa de perda de bytes dentro dos níveis estabelecidos no contrato de tráfego.

Devido aos diversos tipos de serviços e aplicações, tais como, dados, voz e vídeo sendo multiplexados nos nós das redes, a dinâmica da ocupação do *buffer* é um processo complexo e não-linear. Esta ocupação expressa pelo tamanho de fila no *buffer* do sistema da Figura 1 pode ser descrita através da equação de Lindley (Jacobson, 1995):

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

onde B_{max} é a capacidade máxima do *buffer*.

Foi demonstrado que o modelo fuzzy TSK é capaz de descrever com eficiência o comportamento de processos reais com características não-lineares, como é o caso do sistema descrito pela equação (1) (Ouyang et al., 2005; Takagi and Sugeno, 1985). Além disso, sabe-se que sistemas de controle baseados em lógica fuzzy para aplicações em tempo real podem ser implementados utilizando por exemplo, processadores DSP (Digital Signal Processing) (Iliev et al., 2000) e circuitos programáveis FPGA (Field Programmable Gate

Array) (Juang et al., 2004; Surmann et al., 2006). Estas implementações, incluindo as relacionadas com o modelo TSK, atingem alta velocidade de operação, podendo ser utilizadas em aplicações com restrições de tempo no intervalo entre algumas centenas de nanossegundos a alguns microssegundos. A nossa proposta consiste de um modelo TSK cujo algoritmo de treinamento é adaptativo. Assim, é possível implementar o esquema de controle fuzzy apresentado neste trabalho e empregá-lo no controle de aplicações em tempo real nas redes atuais cujo *round trip time* é da ordem de algumas centenas de milissegundos (Aikat et al., 2003).

Na próxima seção propomos um preditor fuzzy adaptativo para o tamanho da fila no *buffer* d passos à frente $b(k+d)$, que explora as propriedades não-lineares e as incertezas da dinâmica do tamanho da fila no *buffer*.

3 ALGORITMO DE PREDIÇÃO FUZZY TSK ADAPTATIVO

Nesta seção, desenvolvemos o algoritmo de treinamento adaptativo para o preditor fuzzy do tipo TSK da ocupação do *buffer*. Denominado de algoritmo de Agrupamento Regressivo Fuzzy Adaptativo com Reset da Covariância (AR-FARC), o algoritmo de treinamento proposto neste trabalho identifica simultaneamente os agrupamentos (clusters) nos dados e os parâmetros das partes conseqüentes das regras fuzzy.

3.1 Modelo Fuzzy TSK Aplicado à Predição do Tamanho de 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 (Chen et al., 2000; Sugeno and Yasukawa, 1993; Takagi and Sugeno, 1985). O modelo fuzzy TSK é um exemplo de tais técnicas, o qual interpola localmente relações de entrada-saída lineares através de funções de pertinência, resultando em sistemas não-lineares.

Seja um conjunto de N 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 sejam relacionadas por uma função desconhecida $y(k) = f(x_1(k), x_2(k), \dots, x_n(k))$, com $1 \leq k \leq N$, deseja-se construir uma função \hat{f} que possa representar essa relação entrada-saída f , através do modelo fuzzy TSK. No nosso caso, os vetores de entrada consistem dos valores passados e atuais do tamanho da fila no *buffer* e do processo

de tráfego controlável e a saída consiste de uma estimativa do tamanho da fila no *buffer* d passos a frente, isto é,

$$\vec{x}(k) = [b(k-p), b(k-p+1), \dots, b(k), \mu(k-q), \mu(k-q+1), \dots, \mu(k)] \quad (2)$$

$$y(k) = b(k+d). \quad (3)$$

O modelo TSK aplicado à predição do tamanho da fila no *buffer* consiste de regras *Se-Então* que tem a seguinte forma:

$$\begin{aligned} R^i : \text{Se } b(k-p) \in A_1^i(\vec{\theta}_1^i) \text{ e } b(k-p+1) \in A_2^i(\vec{\theta}_2^i), \\ \dots, b(k) \in A_{p+1}^i(\vec{\theta}_{p+1}^i) \text{ e } \mu(k-q) \in A_{p+2}^i(\vec{\theta}_{p+2}^i) \\ \text{ e } \mu(k-q+1) \in A_{p+3}^i(\vec{\theta}_{p+3}^i), \\ \dots, \text{ e } \mu(k) \in A_{p+q+2}^i(\vec{\theta}_{p+q+2}^i) \end{aligned}$$

$$\begin{aligned} \text{Então } y^i(k) = b^i(k+d) = \alpha_0^i + \alpha_1^i b(k-p) \\ + \alpha_2^i b(k-p+1) + \dots + \alpha_{p+1}^i b(k) + \\ + \beta_{p+2}^i \mu(k-q) + \beta_{p+3}^i \mu(k-q+1) \\ + \dots + \beta_{p+q+2}^i \mu(k) \end{aligned} \quad (4)$$

para $i = 1, 2, \dots, C$, onde C é o número total de regras, $p+1$ é o número de amostras de entrada referentes aos valores atuais e passados do tamanho da fila no *buffer*, $q+1$ é o número de amostras de entradas relacionadas aos valores atuais e passados do processo de 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 (Takagi and Sugeno, 1985):

$$\hat{y}(k) = \hat{b}(k+d) = \frac{\sum_{i=1}^C b^i(k+d) w^i(k)}{\sum_{i=1}^C w^i(k)}, \quad (5)$$

onde $b^i(k+d)$ 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 representado pela equação (4) consiste na identificação dos parâmetros das partes antecedentes (isto é, $\vec{\theta}_j^i$) e das partes consequentes (isto é, \vec{a}^i). Para isto, introduzimos na próxima seção o algoritmo de agrupamento ARFARC 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. Na primeira parte do algoritmo, uma inicialização aproximada do modelo TSK é estabelecida. E na segunda parte, um algoritmo de refinamento baseado no gradiente descendente (Haykin, 2001) é utilizado para efetuar um ajuste fino do modelo inicial obtido na primeira parte, otimizando a sua precisão. As simulações realizadas mostram que o algoritmo de treinamento adaptativo ARFARC permite uma rápida convergência e resultados bastante expressivos.

3.2 Algoritmo ARFARC - Parte 1

O algoritmo de agrupamento ARFARC proposto leva em conta a distribuição espacial dos dados através da análise da distância entre os dados de entrada e os centros dos agrupamentos. Além disso, no algoritmo proposto ARFARC considera-se o erro de regressão, ou seja, a diferença entre a saída desejada e a saída obtida pelo modelo TSK.

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). Estabelecemos a seguinte função custo J para o algoritmo ARFARC:

$$J = \sum_{i=1}^C \sum_{k=1}^N u_{ik}^2 (r_{ik} d_{ik})^2 \quad (6)$$

com

$$\sum_{i=1}^C u_{ik} = 1, \quad \text{for } 1 \leq k \leq N, \quad (7)$$

onde 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 (6), 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,

$$r_{ik} = y(k) - f_i(\vec{x}(k); \vec{a}^i(k)), \quad (8)$$

para $i = 1, 2, \dots, C$ e $k = 1, 2, \dots, N$. Na mesma equação (6), d_{ik} é a distância entre a k -ésima entrada e o centro do i -ésimo grupo (*cluster*) denotado como β_i , isto é,

$$d_{ik} = \vec{x}(k) - \beta_i. \quad (9)$$

Para minimizar a função custo J em (6), aplicamos o método de multiplicadores de Lagrange (Bortolossi, 2002), resultando em uma função de Lagrange dada por:

$$L = \sum_{i=1}^C \sum_{k=1}^N u_{ik}^2 (r_{ik} d_{ik})^2 - \sum_{k=1}^N \lambda_k \left(\sum_{i=1}^C u_{ik} - 1 \right), \quad (10)$$

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

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

$$\frac{\partial L}{\partial \bar{a}^i(k)} = \sum_{k=1}^N (u_{ik})^2 (d_{ik})^2 \frac{\partial r_{ik}^2}{\partial \bar{a}^i(k)} = 0, \quad (11)$$

$$\frac{\partial L}{\partial u_{ik}} = 2u_{ik}(r_{ik}d_{ik})^2 - \lambda_k = 0, \quad (12)$$

$$\frac{\partial L}{\partial \beta_i} = \sum_{k=1}^N (u_{ik})^2 (r_{ik})^2 \frac{\partial d_{ik}^2}{\partial \beta_i} = 0. \quad (13)$$

Note que a derivada parcial $\frac{\partial r_{ik}^2}{\partial \bar{a}^i(k)}$ em (11) pode ser obtida através da equação (8) da seguinte forma:

$$\frac{\partial r_{ik}^2}{\partial \bar{a}^i(k)} = 2r_{ik} \frac{\partial r_{ik}}{\partial \bar{a}^i(k)} = 2[y(k) - f_i(\bar{x}(k); \bar{a}^i(k))] \frac{\partial r_{ik}}{\partial \bar{a}^i(k)}. \quad (14)$$

Substituindo (14) na equação (11) temos:

$$\sum_{k=1}^N (u_{ik})^2 (d_{ik})^2 \frac{\partial r_{ik}}{\partial \bar{a}^i(k)} y(k) - \sum_{k=1}^N (u_{ik})^2 (d_{ik})^2 \frac{\partial r_{ik}}{\partial \bar{a}^i(k)} f_i(\bar{x}(k); \bar{a}^i(k)) = 0, \quad (15)$$

onde $\frac{\partial r_{ik}}{\partial \bar{a}^i(k)} = \bar{x}(k)$ é obtido derivando a equação (8).

Para o k -ésimo padrão de treinamento, com $k = 1, 2, \dots, N$, definimos uma matriz $X(k) \in \mathbb{R}^{k \times (n+1)}$ cuja $(j+1)$ -ésima coluna é o vetor x_j . A primeira coluna de $X(k)$ é um vetor unitário. 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 cujo k -ésimo elemento da diagonal é dado por $\omega(k) = u_{ik}^2 d_{ik}^2$. Assim sendo, a equação (15) escrita na forma matricial é dada por:

$$X^T(k)Q_i(k)Y(k) - (X^T(k)Q_i(k)X(k))\bar{a}^i(k) = 0. \quad (16)$$

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

$$P_i(k) = (X^T(k)Q_i(k)X(k))^{-1}, \quad i = 1, 2, \dots, C \quad (17)$$

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

$$\bar{a}^i(k) = P_i(k)X^T(k)Q_i(k)Y(k). \quad (18)$$

Para evitar a inversão matricial do termo $(X^T(k)Q_i(k)X(k))$ em (17) no cálculo da matriz de

covariância, propomos um algoritmo recursivo de estimação do vetor de parâmetros das partes conseqüentes \bar{a}^i . Detalhamos o desenvolvimento matemático para obtenção das equações do algoritmo recursivo no Apêndice A, onde mostramos que a equação (18) pode ser calculada recursivamente por:

$$\bar{a}^i(k+1) = \bar{a}^i(k) + P_i(k+1)x^T(k+1)\omega(k+1) \times (y(k+1) - x(k+1)\bar{a}^i(k)), \quad (19)$$

onde $P_i(k+1)$ é dado por:

$$P_i(k+1) = P_i(k) - \left(\frac{P_i(k)x^T(k+1)\omega(k+1)x(k+1)P_i(k)}{1 + \omega(k+1)x(k+1)P_i(k)x^T(k+1)} \right), \quad (20)$$

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 $Q_i(k+1)$.

Através de simulações, constatamos que o método recursivo proposto de estimação dos parâmetros das partes conseqüentes apresenta um alta taxa de convergência inicial nas simulações; contudo o ganho do algoritmo é reduzido pois a matriz de covariância P diminui rapidamente dentro de poucas iterações. Para evitar isso, a matriz P é inicializada a cada 10 iterações de modo que o ganho não é reduzido e 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 pela equação (12), ou seja:

$$u_{ik} = \frac{\lambda_k}{2r_{ik}^2 d_{ik}^2}. \quad (21)$$

O parâmetro λ_k pode ser obtido substituindo a equação (21) na equação (7) obtendo:

$$\lambda_k = \frac{1}{\sum_{i=1}^C \frac{1}{2r_{ik}^2 d_{ik}^2}}. \quad (22)$$

Substituindo o parâmetro λ_k da equação (22) na equação (21) obtém-se a seguinte equação para o grau de ativação da i -ésima regra para o k -ésimo padrão de treinamento u_{ik} :

$$u_{ik} = \frac{1/(2r_{ik}^2 d_{ik}^2)}{\sum_{i=1}^C \frac{1}{2r_{ik}^2 d_{ik}^2}}. \quad (23)$$

A terceira condição, representada pela equação (13), pode ser simplificada uma vez que a derivada parcial $\frac{\partial d_{ik}}{\partial \beta_i}$ nesta equação é dada em função da distância entre a k -ésima saída

desejada do sistema e a saída da i -ésima regra com a k -ésima entrada (equação (9)). Assim, temos:

$$\frac{\partial d_{ik}^2}{\partial \beta_i} = 2d_{ik} \frac{\partial d_{ik}}{\partial \beta_i} = 2\beta_i - 2\vec{x}(k). \quad (24)$$

Substituindo (24) em (13), obtém-se a seguinte equação para o centro do i -ésimo *cluster* β_i :

$$\beta_i = \frac{\sum_{k=1}^N u_{ik}^2 r_{ik}^2 \vec{x}(k)}{\sum_{k=1}^N u_{ik}^2 r_{ik}^2}. \quad (25)$$

Em resumo, para cada padrão de treinamento $(\vec{x}(k), y(k))$, a saída do modelo TSK será calculada pela equação (5) e os parâmetros r_{ik} , d_{ik} , u_{ik} , β_i e \vec{a}^i serão atualizados, respectivamente, pelas equações (8), (9), (23), (25) e (19), caracterizando assim um algoritmo de treinamento adaptativo. Vale também ressaltar que, a atualização do vetor de parâmetros consequentes $\vec{a}^i(k)$ é recursiva, ou seja, utiliza-se a amostra anterior para se calcular a amostra atual.

Nessa primeira parte do algoritmo de agrupamento proposto ARFARC, os parâmetros das partes antecedentes também são obtidos. Funções gaussianas são utilizadas como funções de pertinência das partes premissas (antecedentes); assim, temos que $A_j^i(\theta_{j1}^i; \theta_{j2}^i) = \exp \left\{ - \left(\frac{x_j(k) - \theta_{j1}^i}{\theta_{j2}^i} \right)^2 \right\}$. Para o cálculo das funções de pertinência, devemos portanto obter os parâmetros θ_{j1}^i e θ_{j2}^i . Tais parâmetros correspondem, respectivamente, a média e o 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 θ_{j1}^i e θ_{j2}^i :

$$\theta_{j1}^i(k) = \frac{\sum_{z=1}^N (u_{iz})^2 x_j(z)}{\sum_{z=1}^N (u_{iz})^2}, \quad (26)$$

$$\theta_{j2}^i(k) = \sqrt{\frac{\sum_{z=1}^N (u_{iz})^2 (x_j(z) - \theta_{j1}^i)^2}{\sum_{z=1}^N (u_{iz})^2}}. \quad (27)$$

3.3 Algoritmo ARFARC - Parte 2 (Algoritmo de Refinamento)

Na segunda parte do algoritmo de agrupamento ARFARC, os parâmetros das partes antecedentes $(\vec{a}^i(k))$ e consequentes $(\theta_{j1}^i$ e $\theta_{j2}^i)$ são ajustados por um algoritmo de refinamento baseado em um procedimento de aprendizagem supervisionado de modo a aumentar a precisão da modelagem. O algoritmo de aprendizagem supervisionado considerado é o gradiente descendente (Haykin, 2001). Este algoritmo assume como função custo E_R a seguinte equação:

$$E_R = \frac{1}{N} \sum_{j=1}^N e^2(k), \quad (28)$$

onde $e(k) = y(k) - \hat{y}(k)$, $y(k)$ é a saída desejada e $\hat{y}(k)$ é a saída do modelo nebuloso TSK.

Uma vez que a saída do modelo TSK é calculada por meio das equações (4) e (5), então, para o k -ésimo padrão de treinamento, os parâmetros das partes antecedentes e consequentes serão atualizados pelas Equações I (29) e II (30), respectivamente, detalhadas abaixo.

Equação I (Ajuste dos Parâmetros de Premissa)

A partir da minimização da função custo E_R (equação (28)), é possível mostrar que os parâmetros θ_{jl}^i das partes de premissa do modelo nebuloso com $l = 1$ e 2 , podem ser ajustados precisamente através da equação abaixo:

$$\Delta \theta_{jl}^i(k) = \eta (y(k) - \hat{y}(k)) (y^i(k) - \hat{y}(k)) \frac{1}{\sum_{i=1}^C w^i(k)} \frac{\partial w^i(k)}{\partial \theta_{jl}^i(k)}, \quad (29)$$

onde η 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.

Seja o vetor de parâmetros $w^i(k)$ dado por $\min_{j=1,2,\dots,n} A_j^i(x_j(k))$ e j^* o índice j que define a minimização em $w^i(k)$, isto é, $j^* = \arg \min_{j=1,2,\dots,n} A_j^i(x_j(k))$.

Então, quando $j = j^*$ e $l = 1$, o termo $\frac{\partial w^i(k)}{\partial \theta_{j1}^i(k)}$ da equação (29) pode ser calculado da seguinte forma:

$$\frac{\partial w^i(k)}{\partial \theta_{j1}^i(k)} = \frac{\partial A_j^i}{\partial \theta_{j1}^i(k)},$$

$$\frac{\partial w^i(k)}{\partial \theta_{j1}^i(k)} = \frac{2}{\theta_{j2}^i(k)} \frac{x_j(k) - \theta_{j1}^i(k)}{\theta_{j2}^i(k)} \exp \left\{ - \left(\frac{x_j(k) - \theta_{j1}^i(k)}{\theta_{j2}^i(k)} \right)^2 \right\}.$$

E para $l = 2$, temos:

$$\frac{\partial w^i(k)}{\partial \theta_{j2}^i(k)} = \frac{\partial A_j^i}{\partial \theta_{j2}^i(k)} = \frac{x_j(k) - \theta_{j1}^i(k)}{\theta_{j2}^i(k)} \frac{\partial w^i(k)}{\partial \theta_{j1}^i(k)}.$$

Quando $j \neq j^*$, tem-se que $\frac{\partial w^i(k)}{\partial \theta_{j2}^i(k)} = \frac{\partial w^i(k)}{\partial \theta_{j1}^i(k)} = 0$, o que significa que nenhum ajuste será feito para esses casos.

Equação II (Ajuste dos Parâmetros Consequentes)

Os parâmetros das partes consequentes do modelo nebuloso podem ser ajustados aplicando o algoritmo de gradiente descendente, resultando na seguinte equação de atualização:

$$\Delta a_j^i(k) = \zeta (y(k) - \hat{y}(k)) \frac{w^i(k) x_j(k)}{\sum_{i=1}^C w^i(k)}, \quad (30)$$

onde ζ é outra taxa de aprendizagem, $y(k)$ é a saída desejada e $\hat{y}(k)$ é a saída do modelo nebuloso TSK.

4 CÁLCULO DA TAXA ÓTIMA DE CONTROLE DE FLUXOS

Nesta seção, como parte do esquema de controle de congestionamento adaptativo proposto, apresentamos uma expressão para o cálculo da taxa ótima de controle para regular a taxa $\mu(k)$ da fonte controlável a partir da minimização da função custo J abaixo:

$$J(k+d) = E \left[\frac{1}{2} (b(k+d) - b^\tau)^2 + \frac{\lambda}{2} \mu^2(k) \right], \quad (31)$$

onde λ é o fator de ponderação e $E[.]$ denota a esperança matemática. Na equação (31), 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^τ . Os erros de modelagem (imprecisão do modelo), mesmo sendo pequenos, podem causar oscilações no tamanho de fila do *buffer* $b(k)$ de tal forma a ultrapassar o nível de referência b^τ . Devido a este fato, introduzimos na equação (31) o parâmetro de projeto λ para limitar o valor de $\mu^2(k)$, evitando situações como essa.

A taxa de controle ótima é dada em função do valor do tamanho da fila no *buffer* d passos à frente. Uma estimativa do tamanho da fila $b(k)$ no *buffer* d passos à frente é fornecida pela saída do preditor fuzzy proposto quando aplicado à predição de amostras deste processo pela seguinte equação:

$$\hat{b}(k+d) = \sum_{i=1}^C b^i(k+d) h^i(k), \quad (32)$$

onde

$$h^i(k) = \frac{w^i(k)}{\sum_{i=1}^C w^i(k)} \quad \text{com} \quad w^i(k) = \min_{j=1 \dots n} A_j^i(\vec{\theta}_j^i; x_j(k)) \quad (33)$$

e

$$\begin{aligned} b^i(k+d) = & \alpha_0^i + \alpha_1^i b(k-p) + \\ & \alpha_2^i b(k+p+1) + \dots + \alpha_{p+1}^i b(k) + \beta_{p+2}^i \mu(k-q) + \\ & \beta_{p+3}^i \mu(k-p+1) + \dots + \beta_{p+q+2}^i \mu(k). \end{aligned} \quad (34)$$

Assim sendo, podemos enunciar a seguinte proposição relacionada à taxa ótima de controle.

Proposição 1. *A taxa ótima de controle μ^o que pode ser utilizada para regular a taxa da fonte controlável e que minimiza*

a função custo J (31) é dada pela seguinte equação:

$$\begin{aligned} \mu^o(k) = & \beta_0(k) \times (b^\tau - \sum_{i=1}^C (\alpha_0^i + \sum_{t=1}^{p+1} \alpha_t^i b(k-p-1+t)) \\ & + \sum_{t=1}^q \beta_{t+p+1}^i \mu(k-q-1+t)) h^i(k), \end{aligned} \quad (35)$$

onde

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

Prova: A prova se encontra no Apêndice B.

Segundo a equação proposta (35), a taxa ótima de controle de fluxo $\mu^o(k)$ é obtida em função 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^τ . 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*.

5 AVALIAÇÃO DO PREDITOR NEBULOSO PROPOSTO

Nessa seção, serão realizadas avaliações comparativas entre o desempenho do preditor nebuloso proposto e o desempenho de outros preditores, quando aplicados a traços de tráfego reais. Inicialmente, apresentamos as medidas de desempenho de predição utilizadas.

5.1 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).

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,

$$EQM = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2, \quad (36)$$

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). No nosso trabalho, duas medidas de erro

relativas são empregadas: o EQM normalizado em relação à variância da série predita, e o EQM normalizado em relação ao erro quadrático médio do preditor ótimo para o processo passeio aleatório (*random walk*) (Papoulis, 2002).

Erro Quadrático Médio Normalizado 1 - EQMN1: Seja σ_y^2 a variância do processo Y , dada por $\sigma_y^2 = \frac{1}{N} \sum_{i=1}^N (\mu - y_i)^2$, onde μ é a média do processo Y . O erro quadrático médio do tipo 1 é dado por:

$$EQMN1 = \frac{EQM}{\sigma_y^2} = \frac{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2}{\frac{1}{N} \sum_{i=1}^N (\mu - y_i)^2}. \quad (37)$$

Analisando a equação (37), 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 μ . 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.

Erro Quadrático Médio Normalizado 2 - EQMN2: Considerando o preditor simplificado descrito por:

$$\hat{y}_i = y_{i-1}. \quad (38)$$

Ou seja, o valor predito é igual ao valor da amostra imediatamente anterior. Para o processo passeio aleatório (*random walk*), esse é o preditor ótimo no sentido do erro quadrático médio (Papoulis, 2002).

Seja \hat{y}_i^{pa} o valor predito da amostra y_i do processo Y , estimado através do preditor descrito pela equação (38). O erro quadrático médio do tipo 2 será dado por:

$$EQMN2 = \frac{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2}{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i^{pa} - y_i)^2}. \quad (39)$$

Analisando a equação acima, temos que 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.

5.2 Análise do Erro Quadrático Médio de Predição

O desempenho do preditor proposto é analisado através de simulações com traços TCP/IP (Transmission Control Protocol/Internet Protocol) que correspondem ao registro do tráfego de pacotes IP transmitidos entre a Digital Equipment Corporation¹ e o resto do mundo (*dec-pkt-1.tcp*, *dec-pkt-*

¹<http://ita.ee.lbl.gov/html/contrib/DEC-PKT.html>

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

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

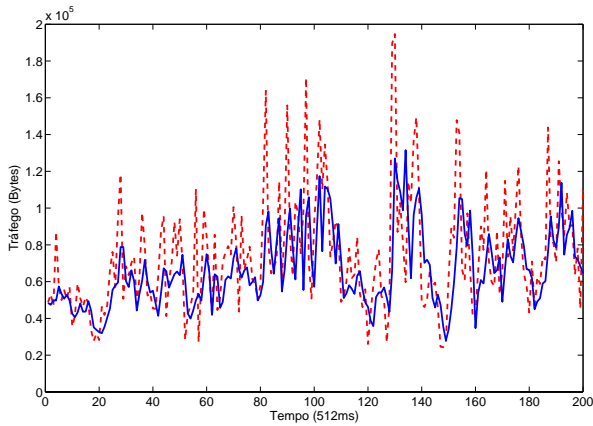
2.tcp e *dec-pkt-3.tcp*). Outro traço utilizado (*OctInt*) refere-se a amostras de tráfego LAN Ethernet disponibilizadas pela Bellcore². Este traço de tráfego (*OctInt*) corresponde a fluxos de bytes trafegados internamente na rede local da Bellcore. Utilizamos também nas simulações traços de tráfego (ex. traço *10-7-S-1*) 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 (Perlingeiro and Ling, 2005). Todos os traços de tráfego utilizados representam volume de tráfego em bytes por unidade de tempo.

A escolha desses traços de tráfego, TCP/IP e LAN Ethernet, justifica-se por terem sido empregados em outros trabalhos relacionados à controle de congestionamento e predição (Barreto and Lemos, 2001; Vieira and Lee, 2003; Bianchi et al., 2004; Dangy et al., 2003; Banerjee et al., 2000), portanto, servem bem ao propósito de comparação. A Figura 2 exibe as respostas do preditor de tráfego proposto a um passo para o traço de tráfego TCP/IP *dec-pkt-1* na escala de tempo de agregação de 512ms, e também para o traço de tráfego da Petrobrás *10-7-S-1*, na escala de 100ms.

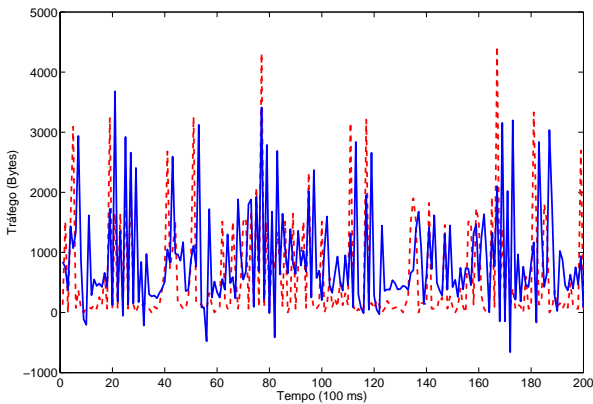
Com relação a escolha do número de amostras passadas do tráfego, ou seja, a ordem do preditor, sabe-se que ao se utilizar poucas amostras tem-se um menor tempo de processamento das informações, em contrapartida, esse pequeno número de amostras utilizado pode comprometer o desempenho de predição. Neste trabalho foi escolhido $n = 6$ como a ordem do preditor pois verificamos que deste modo se obtém um bom equilíbrio entre tempo de processamento e desempenho de predição. Mais especificamente, constatamos que o erro quadrático médio para o preditor proposto diminui de forma insignificativa com o aumento da ordem do preditor acima de $n = 6$.

Outro fator importante para o desempenho do algoritmo de predição é a escolha das taxas de aprendizagem. As taxas de aprendizagem do algoritmo de refinamento foram estipuladas de tal forma a obter melhores resultados possíveis. Assim, os valores encontrados foram $\eta = 0.62$ e $\zeta = 0.72$.

²<http://ita.ee.lbl.gov/html/contrib/BC.html>



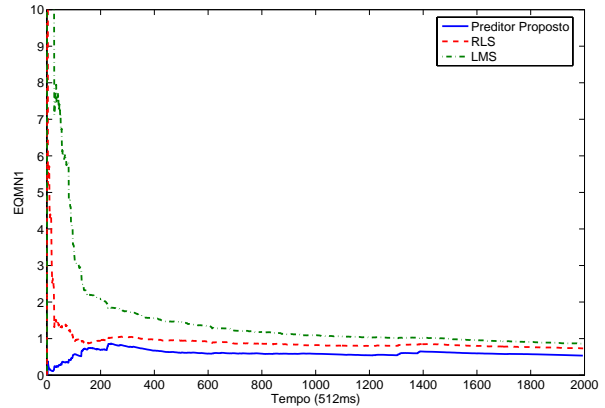
(a) Traço de tráfego dec-pkt-1



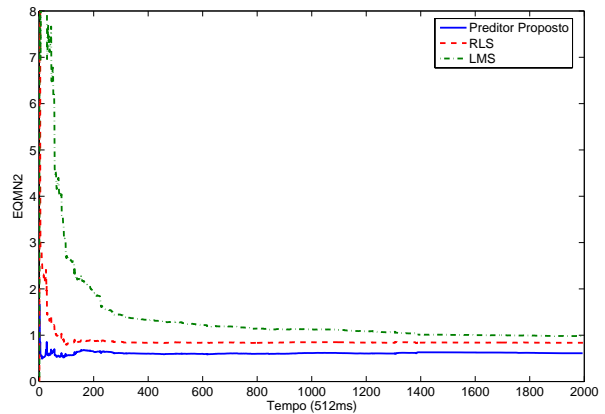
(b) Traço de tráfego 10-7-S-1

Figura 2: Predição a um Passo de Séries Temporais de Tráfego. Série Real (Linha Tracejada), Série Predita (Linha Cheia).

Dentre os preditores lineares adaptativos existentes na literatura, os algoritmos LMS (*Least Mean Square*) (Toreyin et al., 2006) e RLS (*Recursive Least Square*) (Chong et al., 1995) merecem atenção. O primeiro, o filtro linear adaptativo LMS, pertence à família de algoritmos de gradiente estocástico, que não requer medições das funções de correlação, nem inversão da matriz de autocorrelação. Enquanto o filtro linear RLS, é visto como um caso especial do filtro de Kalman (Haykin, 2001). Os filtros LMS e RLS são bastante empregados na predição de séries temporais (Adas, 1998; Toreyin et al., 2006; Chong et al., 1995). Para o preditor RLS, fixou-se um fator de esquecimento igual a $\lambda = 0.99$. E para o caso do LMS, foi utilizado uma taxa de aprendizagem de $\eta = 0.03$. A ordem para os preditores comparados foi a



(a) Comparação do EQMN1 para o Traço dec-pkt-1



(b) Comparação do EQMN2 para o Traço dec-pkt-1

Figura 3: Comparação dos Erros entre o Preditor Proposto e Outros Preditores.

mesma utilizada para o preditor proposto, ou seja, $n = 6$.

As Tabelas 1 e 2 mostram as medidas de desempenho obtidas utilizando o preditor nebuloso proposto assim como para os algoritmos LMS e RLS. A Figura 3 compara as medidas de desempenho de predição (EQM, EQMN1, EQMN2) entre o preditor proposto e os outros preditores utilizados como comparação. Pode-se verificar que o preditor proposto apresentou os menores EQM, EQMN1 e EQMN2, ou seja, o melhor desempenho entre os preditores analisados. Com o uso de poucas amostras iniciais, o erro quadrático do preditor proposto decai e se mantém próximo a zero. Na mesma figura, percebe-se que os outros preditores analisados requerem um maior intervalo de tempo para atingir resultados similares. Desse modo, pode-se concluir que a aplicação do modelo fuzzy adaptativo na predição de tráfego conduz

Tabela 2: Desempenho do Preditor Proposto e dos Outros Preditores Lineares Quando Aplicados ao Traço de Tráfego Real 10-7-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

a ganhos expressivos de desempenho. É conveniente frisar que o preditor proposto possui características adaptativas, ou seja, o ajuste do parâmetro do modelo é realizado a cada nova amostra de tráfego, não sendo necessário o uso de todas as amostras. Essa característica permite que o preditor proposto seja usado em aplicações de tempo real, mantendo o desempenho de predição desejado.

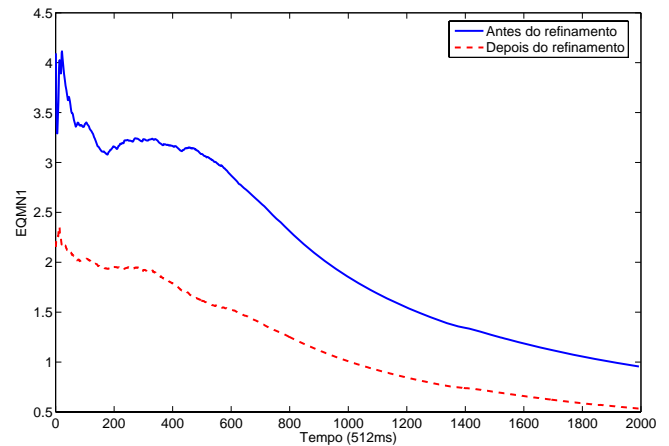
5.3 Desempenho do Algoritmo de Gradiente Descendente aplicado ao Modelo Fuzzy

Aplica-se na segunda etapa do treinamento do preditor fuzzy proposto um algoritmo de refinamento baseado no gradiente descendente como ajuste fino dos parâmetros do modelo inicial obtido. A Figura 4 mostra as medidas de desempenho da predição (EQMN1 e EQMN2) para o traço de tráfego *dec-pkt-1* onde pode-se observar as melhorias no desempenho de predição. De fato, as medidas de erro utilizadas apresentam seus valores reduzidos após a aplicação do algoritmo, conforme se pode observar também pela Tabela 3.

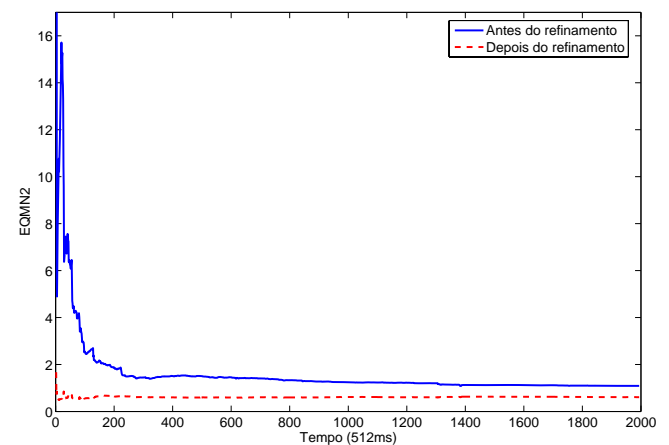
6 AVALIAÇÃO DO ESQUEMA DE CONTROLE DE CONGESTIONAMENTO PROPOSTO

O esquema de controle de congestionamento proposto é constituído de um módulo de predição adaptativa de tráfego representado pelo algoritmo de predição proposto e um módulo relativo ao cálculo da taxa ótima para os fluxos. Os parâmetros obtidos na predição do tamanho de fila no *buffer* são usados para calcular a taxa da fonte de tal forma que o tamanho da fila no *buffer* se aproxime do nível desejado e a sua variância se torne mínima. O esquema de controle de fluxos proposto é inserido no bloco “Controle de Fluxo Nebuloso Preditivo Adaptativo” da Figura 1.

Nessa seção, avaliamos o esquema de controle de congestionamento proposto e o comparamos a outros métodos de controle de congestionamento. Inicialmente, apresentamos os



(a) EQMN1



(b) EQMN2

Figura 4: Comparação das Medidas de Erro Antes e Depois do Refinamento para o Traço de Tráfego *dec-pkt-1*.

métodos de controle e as medidas de desempenho utilizados na comparação. Em seguida, validamos o esquema de controle de congestionamento proposto através de simulações. Para validação do esquema de controle proposto considerou-se um cenário de rede composto por 3 nós e 5 fontes de tráfego utilizando os traços *dec-pkt-2* e *dec-pkt-3* na escala de 512ms.

Os métodos utilizados na comparação de desempenho do esquema de controle de congestionamento proposto são: método de Realimentação Binária (Ramakrishnan and Jain, 1990) e método de Controle Proporcional (Habib and Sadawi, 1992). Tais métodos foram escolhidos por serem bastante utilizados na literatura (Karnik and Kumar, 2005;

Tabela 3: Comparação dos Resultados de Desempenho do Preditor Proposto Antes e Depois do Refinamento.

Traço	EQMN		EQMN1		EQMN2	
	antes	depois	antes	depois	antes	depois
dec-pkt-1	0.0120	0.0067	0.9547	0.5340	1.0915	0.6106
de-pkt-2	0.0112	0.0071	0.6515	0.4113	0.8746	0.5577
Octint	0.0436	0.0156	1.3179	0.4691	1.3401	0.4787
10-7-S-1	0.0029	0.0025	2.0708	1.7997	0.9885	0.8601

Zhang et al., 2000) 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, sendo também independentes de protocolos específicos de rede. Além disso, assim como no esquema de controle proposto, nestes métodos de controle um valor limitante para o tamanho da fila no *buffer* pode ser estabelecido.

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* (Chen et al., 1996; Karnik and Kumar, 2005). 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^τ é o nível desejado para o tamanho da fila no *buffer*.

Controle de Congestionamento Proporcional (Proportional Congestion Control) - Esse método de controle de congestionamento monitora o tamanho da fila no *buffer* e usa esse valor para regular a taxa da fonte controlável através de uma variável de controle $c(k)$ (Habib and Saadawi, 1992).

Seja b^τ 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:

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

onde 0.998 pode ser visto como um limiar inferior e 1.002 como um limiar superior para o sinal de controle. O sinal de controle da taxa $c(k)$ é inversamente proporcional a ocupa-

ção do *buffer* quando $0.998b^\tau < b(k) < 1.002b^\tau$. Quando $b(k)$ é maior do que b^τ , 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^τ , 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 $\mu(k+1) = c(k)\mu(k)$.

Outros métodos de controle de congestionamento podem ser encontrados na literatura mas que são específicos para determinados protocolos e mecanismos de redes, como é o caso do RED (*Random Early Detection*) e de suas variantes (Gevros et al., 2001; Basar and Srikant, 2005; Wang et al., 2007) e algoritmos baseados em gerência de fila para redes TCP/IP (Wang et al., 2006). O RED por exemplo se baseia especificamente nos mecanismos de controle de congestionamento do protocolo TCP/IP. Por outro lado, para se implementar o esquema de controle de fluxos proposto em redes atuais, é necessário apenas que a rede tenha mecanismos de realimentação (canal de retorno) para controlar a taxa das fontes controláveis, algo comum em várias tecnologias de rede (Wenshan et al., 2006; Boggia et al., 2007; Karnik and Kumar, 2005). Portanto, devido as suas características, o esquema proposto pode ser aplicado a vários tipos de redes e poderá ser absorvido por várias tecnologias (protocolos) futuras de redes de comunicações.

6.1 Medidas de Desempenho Utilizadas para Avaliação do Controle de Fluxos

A avaliação do desempenho do esquema de controle de congestionamento proposto se baseia nas seguintes medidas de desempenho: TPB (Taxa de Perda de Bytes), Utilização do enlace 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 transmissão de dados. Ou seja,

$$TPB(\%) = \frac{\text{No de Bytes Perd.}}{\text{No Total de Bytes Transm.}} \times 100\% \quad (40)$$

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

$$\text{Utilização}(\%) = \frac{\text{No de Bytes Transm.}}{\text{Capacidade do Enlace}} \times 100\% \quad (41)$$

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

$$\text{Vazão}(\text{bytes}/\text{seg}) = \frac{\text{No de Bytes Transm.}}{\text{Tempo de Observação}}. \quad (42)$$

6.2 Controle de Fluxos de Rede - Traços de Tráfego *dec-pkt-2* e *dec-pkt-3*

6.2.1 Cenário de Simulação

O desempenho do esquema de controle proposto é avaliado através de simulações em um cenário de rede composto por 3 nós e vários fluxos de tráfego de entrada (Figura 5). 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, isto é, que devem adaptar a sua taxa de acordo com a capacidade disponível no enlace correspondente.

Como fluxos de tráfego de entrada, são considerados os traços TCP/IP *dec-pkt-2* e *dec-pkt-3* em uma escala de agregação de 512ms. As especificações detalhadas dos componentes do cenário da Figura 5 são descritas 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: 210Kbytes/s, $d = 1$ 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: 200 Kbytes/s, $d = 2$ intervalos de tempo ($2 \times 512ms$).

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

Nó 3: Capacidade do Enlace: 200Kbytes/s, $d = 1$ intervalo de tempo (512ms).

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

6.2.2 Desempenho do Esquema de Controle de Congestionamento Proposto

Para a simulação do esquema de controle de congestionamento proposto neste cenário, estipulamos 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 para o tamanho da fila no *buffer* é estabelecido como $b^{\tau} = 30\%B_{max}$, ou seja, $b^{\tau} = 4.5 \times 10^5$ bytes. O parâmetro λ do cálculo da taxa ótima de controle de fluxo é fixado inicialmente em 1 e o algoritmo de controle é implementado em cada nó do cenário considerado.

Verificamos que as intensidades das taxas dos fluxos controlável e não-controlável em média possuem a mesma ordem de grandeza, com exceção do nó 3 onde a média da taxa não-controlável ultrapassa ligeiramente a média da taxa controlável. Na Figura 6, 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^{τ} , exceto para o nó 3, no qual o tamanho da fila no *buffer* em um breve instante ultrapassa o nível desejado. Nota-se 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.

A Figura 7 apresenta uma comparação da taxa da fonte controlável antes e depois da aplicação do controle de congestionamento proposto. Através da Figura 7, é possível observar que a taxa controlável foi reduzida e algumas vezes até mesmo levada a um valor nulo. Como foi já dito anteriormente, 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.

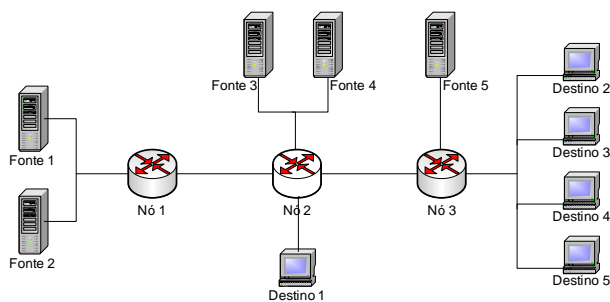


Figura 5: Topologia de Rede Proposta para Análise.

Logo, é possível que em determinados instantes, o tráfego dessas fontes controláveis seja todo vetado em favor das fontes não-controláveis.

Comparamos o desempenho do esquema de controle de congestionamento proposto com os dos métodos de Realimentação Binária e o método de Controle Proporcional. A Figura 8 mostra o tamanho da fila no *buffer*, onde em cada figura referente a cada nó da rede, apresenta-se respectivamente o método de Realimentação Binária, o método de Controle Proporcional e o método proposto. Através da Figura 8, pode-se observar que o método proposto obtém um processo de fila no *buffer* bem mais comportado e abaixo do nível desejado.

No nó 1, os tamanhos da fila no *buffer* obtidos com a aplicação dos dois métodos de comparação são limitados ao tamanho do *buffer* $B_{max} = 1.5 \times 10^6$ bytes, o que provoca a ocorrência de perdas no *buffer*. Essas perdas acontecem pois os métodos comparados respondem, aumentando ou diminuindo sua taxa, de acordo com o tamanho da fila no *buffer*. Como se pode perceber, o tamanho da fila no nó 1 (Figura 6.a) é inicialmente nulo e assim permanece até as 200 primeiras amostras. Assim, como mostra a Figura 9, os métodos comparados, que aumentam a taxa quando o tamanho da fila no *buffer* está pequeno, 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 das fontes controláveis e, consequentemente, o tamanho da fila no *buffer*. Essa lentidão ocorre principalmente no método de Controle Proporcional, conforme se pode observar na Figura 9, e tal método é o que obteve maiores perdas. Esse comportamento não ocorre com o método proposto, como podemos ver pela Figura 7 onde a taxa é controlada dinamicamente e assim não ocorrem perdas, ou seja, temos TPB igual a 0%. Na Tabela 4, os resultados de desempenho obtidos são apresentados, onde podemos observar que nos métodos 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 pelo adequado desempenho do algoritmo em controlar a taxa controlável, eliminando as perdas em consequência de um pequeno decremento na utilização do enlace.

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 5. 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 Biná-

ria resultou em um *buffer* com perdas, fato já explicado anteriormente. Mais uma vez, uma menor variância do processo de fila no *buffer* foi obtida utilizando o método proposto. Os resultados são apresentados na Tabela 6.

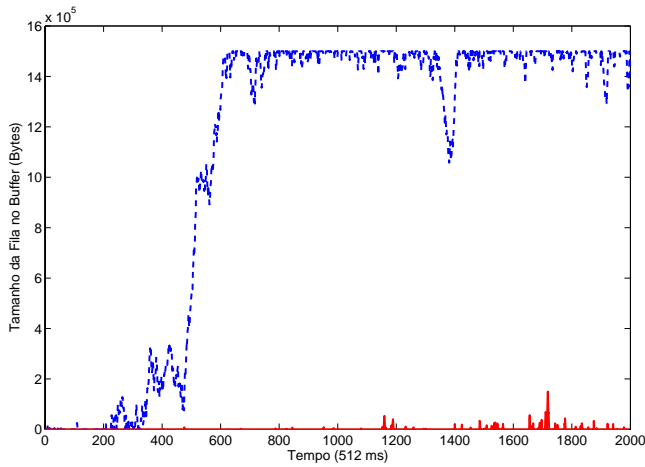
Através das Tabelas 4, 5 e 6, pode-se notar que a média e a variância do tamanho da fila no *buffer* diminuem consideravelmente em todos os nós do cenário considerado após a aplicação do método proposto. Além disso, o TPB é reduzido a zero. Assim, o esquema de controle proposto oferece melhorias de desempenho em termos de taxa de perdas e do comportamento da fila no *buffer*. De fato, observou-se um comportamento de fila no *buffer* mais bem comportado com a aplicação do esquema de controle proposto uma vez que foram obtidas médias e a variâncias menores para a taxa de ocupação do *buffer*. A utilização do enlace, após aplicação do controle proposto, no entanto, é ligeiramente reduzida em todos os nós (em no máximo 11,89% no caso do Nó 1) devido a habilidade do esquema de controle em limitar a taxa da fonte controlável em favor da fonte não-controlável.

Conforme mencionado na seção 2, é possível implementar modelos fuzzy (e consequentemente o esquema de controle proposto neste trabalho) diretamente em DSPs ou FPGAs. Assim, não trataremos neste artigo de questões relacionadas ao custo computacional dos esquemas de controle visto que os mesmos podem ser implementados em hardware para aplicações em tempo real.

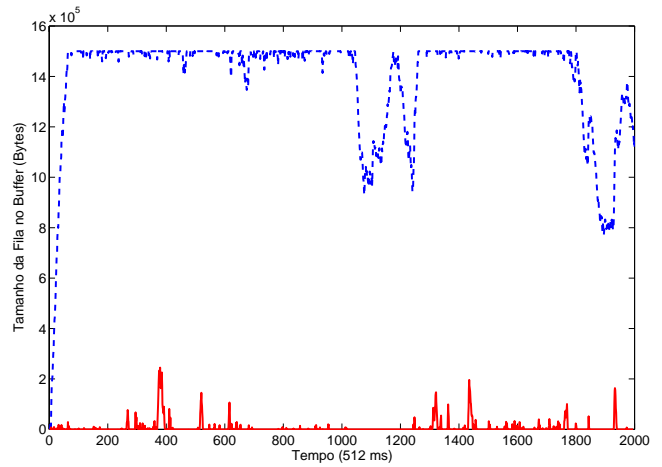
7 CONCLUSÕES

No presente trabalho introduzimos um novo esquema de controle não-linear de fluxos de tráfego. Para tal, desenvolvemos um algoritmo de treinamento adaptativo para um modelo preditivo fuzzy do tipo TSK. A estratégia do controle de congestionamento proposto consiste em prever o comportamento do sistema, mais especificamente, o tamanho da fila no *buffer*, e em seguida, com os parâmetros obtidos na predição, controlar a taxa da fonte controlável de tal forma que a variância do tamanho da fila seja mínima possível. Na primeira etapa do processo de predição, com a aplicação de um algoritmo de treinamento com *reset* da matriz de covariância, os agrupamentos (*clusters*) fuzzy e os parâmetros do modelo nebuloso são obtidos de uma forma adaptativa. Na segunda etapa, um algoritmo de refinamento baseado no algoritmo de gradiente descendente é usado, obtendo predições precisas de tráfego de redes.

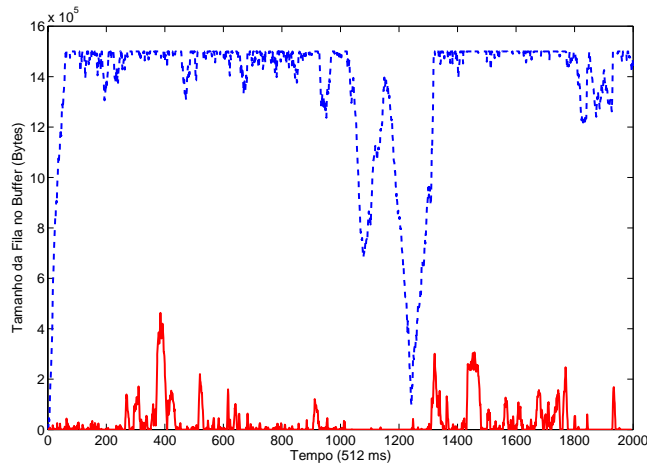
Análises comparativas entre o preditor fuzzy TSK proposto e outros preditores adaptativos mostraram que a aplicação da lógica fuzzy na modelagem de fluxos de tráfego proporcionou ganhos de desempenho de predição e resultados expressivos com relação a medidas de erro analisadas (EQM, EQMN1 e EQMN2).



(a) Nó 1



(b) Nó 2

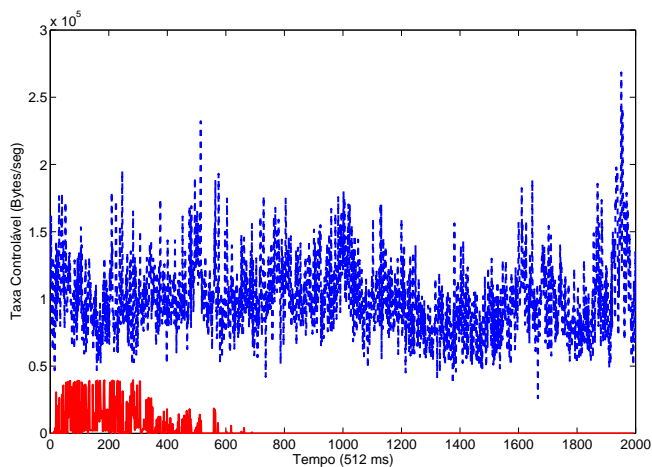


(c) Nó 3

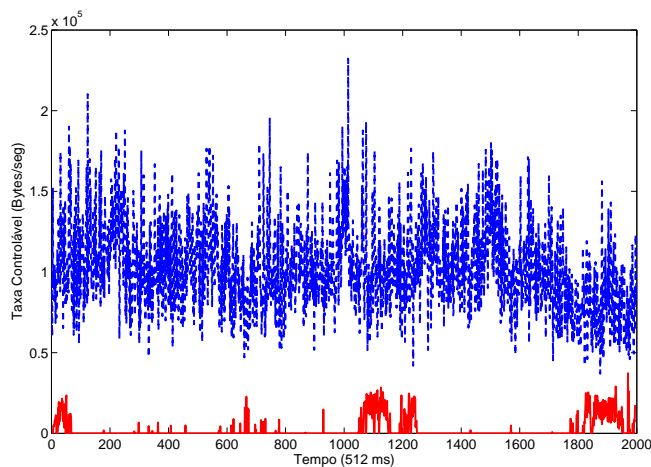
Figura 6: Tamanho da Fila no *Buffer* Antes (Linha Tracejada) e Depois (Linha Cheia) da Aplicação do Controle de Congestionamento Proposto para Cada Nó do Cenário.

Tabela 4: Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento no Nó 1 - Cenário 1.

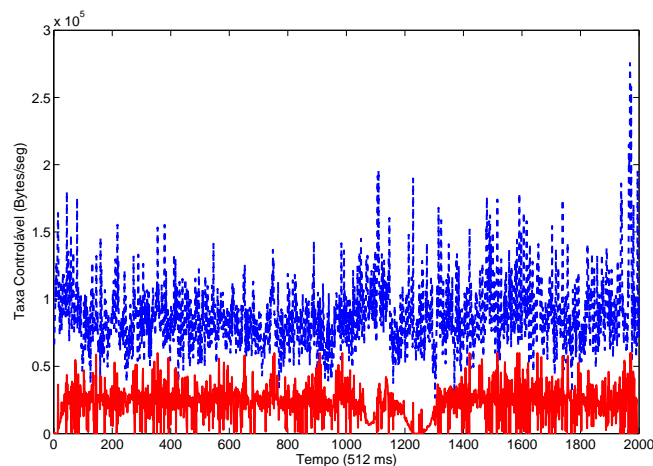
Parâmetro	Binary Feedback		Proportional Cong. Control		Controle de Congestion. Proposto	
	antes	depois	antes	depois	antes	depois
Média do Buffer	1.1003×10^6	4.5068×10^5	1.1003×10^6	7.1245×10^5	1.1003×10^6	1.1294×10^3
Variância do Buffer	3.6265×10^{11}	4.5109×10^{11}	3.6265×10^{11}	4.9275×10^{11}	3.6265×10^{11}	5.6721×10^7
TPB	9.0304%	45.3379%	9.0304%	11.0908%	9.0304%	0.0000%
Utilização	99.9022%	99.95%	99.9022%	99.9500%	99.9022%	88.0151%
Vazão	1.0025×10^5	6.3900×10^4	1.0025×10^5	8.9847×10^4	1.0025×10^5	2.3483×10^3



(a) Nó 1



(b) Nó 2

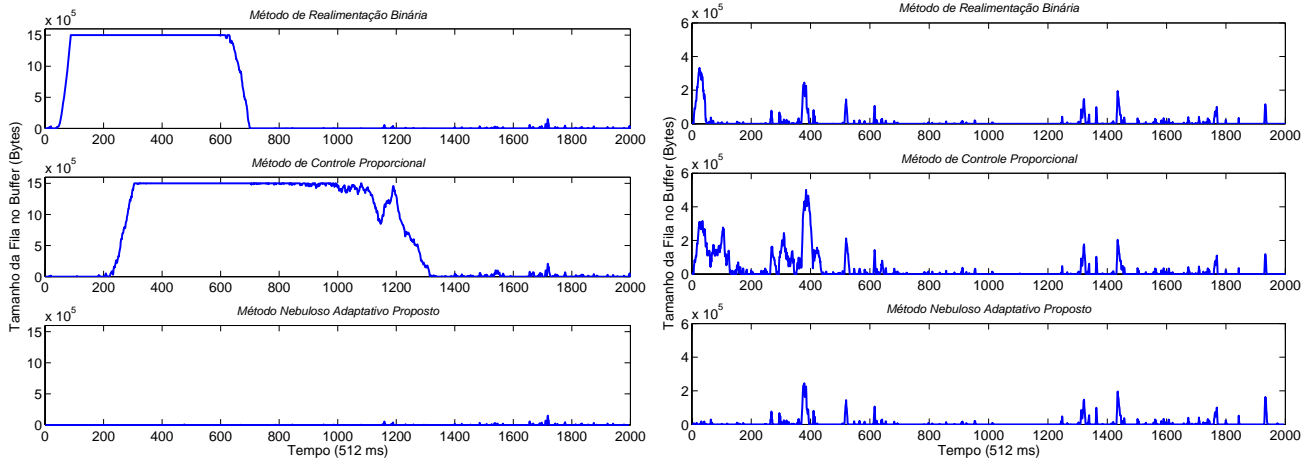


(c) Nó 3

Figura 7: Taxa da Fonte Controlável Antes (Linha Tracejada) e Depois (Linha Cheia) da Aplicação do Controle de Congestionamento Proposto para Cada Nó do Cenário.

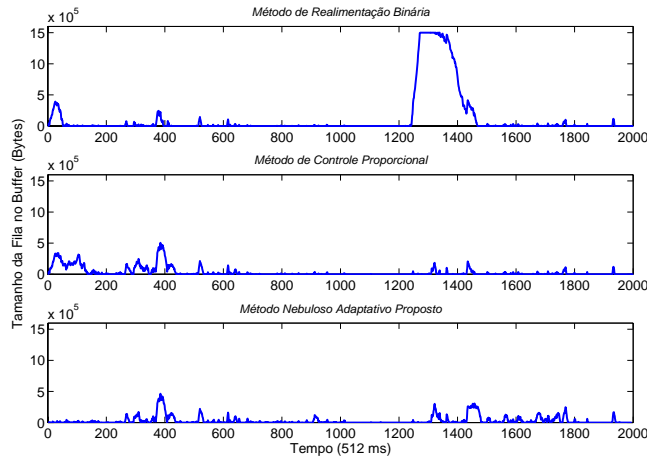
Tabela 5: Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento no Nó 2 - Cenário 1.

Parâmetro	Binary Feedback		Proportional Cong. Control		Controle de Congestion. Proposto	
	antes	depois	antes	depois	antes	depois
Média do Buffer	1.4166×10^6	1.0217×10^4	1.4847×10^6	2.8546×10^4	1.4011×10^6	6.3494×10^3
Variância do Buffer	3.6078×10^{10}	1.5317×10^9	1.1178×10^{10}	4.8836×10^9	4.5773×10^{10}	6.4434×10^8
TPB	51.7846%	0.0000%	39.05%	0.0000%	17.0417%	0.0000%
Utilização	99.9000%	99.9000%	99.9000%	99.9000%	99.9000%	99.8492%
Vazão	1.3251×10^5	1.9704×10^3	1.6549×10^5	1.5011×10^4	1.0220×10^5	2.3280×10^3



(a) Nó 1

(b) Nó 2

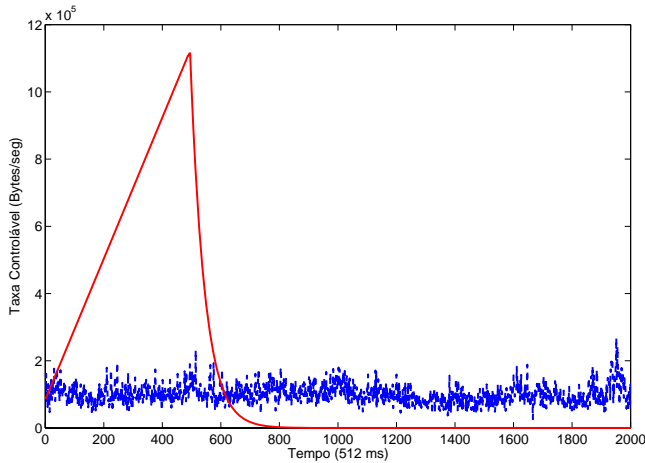


(c) Nó 3

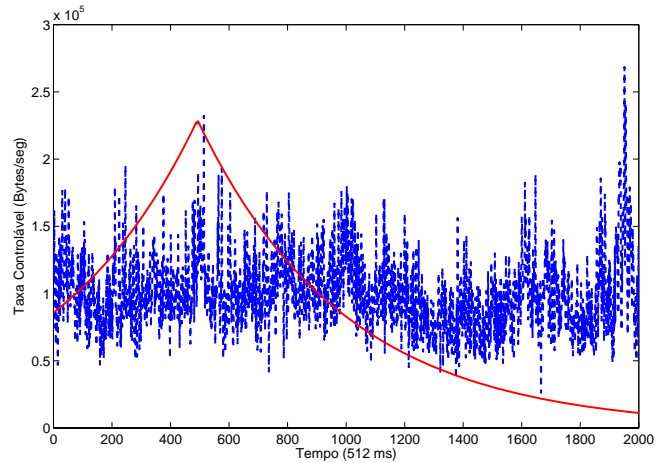
Figura 8: Comparação do Tamanho da Fila no *Buffer* Resultante do Controle de Congestionamento Proposto e dos Outros Métodos de Controle para Cada Nó do Cenário.

Tabela 6: Comparação dos Resultados de Desempenho entre os Métodos de Controle de Congestionamento no Nó 3 - Cenário 1.

Parâmetro	Binary Feedback		Proportional Cong. Control		Controle de Congestion. Proposto	
	antes	depois	antes	depois	antes	depois
Média do Buffer	1.3002×10^6	1.2097×10^5	1.3552×10^6	3.1177×10^4	1.3663×10^6	2.8431×10^4
Variância do Buffer	1.4450×10^{11}	1.2805×10^{11}	9.1041×10^{10}	5.5436×10^9	7.5793×10^{10}	4.2827×10^9
TPB	11.7364%	1.7121%	16.2299%	0.0000%	11.4931%	0.0000%
Utilização	99.9500%	99.9500%	99.9500%	99.9500%	99.9500%	99.9498%
Vazão	8.6965×10^4	1.1513×10^4	9.9982×10^4	1.5392×10^4	8.7301×10^4	2.2879×10^4



(a) Método de Realimentação Binária



(b) Método de Controle Proporcional

Figura 9: 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).

Em seguida, avaliamos o esquema de controle de congestionamento proposto em um cenário de rede com traços de tráfego real. Os resultados das simulações mostraram que o método proposto apresentou, entre os métodos de controle analisados, 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.

Verificou-se que a aplicação do esquema de controle proposto garante que o processo de fila no *buffer* não ultrapasse o valor estipulado. Com isso, é possível se obter ao mesmo tempo um retardo máximo e taxa de perda nula para os fluxos, ou seja, se garante alguns parâmetros de qualidade de serviço aos fluxos de tráfego. Várias aplicações em redes atuais podem se beneficiar desses resultados, como tráfego de vídeo, voz sobre IP, etc (Hong et al., 2001; Baldi and Ofek, 2000). Enfim, o algoritmo de controle preditivo adaptativo baseado no modelo fuzzy proposto é uma ferramenta promissora em situações reais para controle adaptativo dos fluxos de tráfego com garantia de QoS.

8 APÊNDICES

8.1 Apêndice A: Cálculo do Vetor de Parâmetros Consequentes \vec{a}^i

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

$$\vec{a}^i(k) = P_i(k)X^T(k)Q_i(k)Y(k). \quad (43)$$

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

$$\vec{a}^i(k+1) = P_i(k+1)X^T(k+1)Q_i(k+1)Y(k+1). \quad (44)$$

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

$$\vec{a}^i(k+1) = P_i(k+1) \begin{pmatrix} X(k) \\ x(k+1) \end{pmatrix}^T \times \begin{bmatrix} Q_i(k) & 0 \\ 0 & \omega(k+1) \end{bmatrix} \begin{pmatrix} Y(k) \\ y(k+1) \end{pmatrix}, \quad (45)$$

ou

$$\vec{a}^i(k+1) = P_i(k+1)[X^T(k)Q_i(k)Y(k) + x^T(k+1)\omega(k+1)y(k+1)]. \quad (46)$$

Sabe-se que, da equação (43), $X^T(k)Q_i(k)Y(k) = P_i^{-1}(k)\vec{a}^i(k)$. Então, substituindo este resultado em (46), temos que:

$$\vec{a}^i(k+1) = \vec{a}^i(k) + (P_i(k+1)P_i^{-1}(k) - I)\vec{a}^i(k) + P_i(k+1)x^T(k+1)\omega(k+1)y(k+1). \quad (47)$$

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

$$P_i(k+1) = (X^T(k)Q_i(k)X(k) + x^T(k+1)\omega(k+1)x(k+1))^{-1}. \quad (48)$$

Substituindo $X^T(k)Q_i(k)X(k) = P_i^{-1}(k)$ na equação (48) temos:

$$P_i(k+1) = (P_i^{-1}(k) + x^T(k+1)\omega(k+1)x(k+1))^{-1}. \quad (49)$$

Finalmente, substituindo (49) em (47), pode-se calcular recursivamente o vetor dos parâmetros consequentes pela seguinte equação:

$$\begin{aligned} \bar{a}^i(k+1) &= \bar{a}^i(k) + P_i(k+1)x^T(k+1)\omega(k+1) \times \\ & (y(k+1) - x(k+1)\bar{a}^i(k)). \end{aligned} \quad (50)$$

O vetor $\bar{a}^i(k+1)$ é calculado utilizando $P_i(k+1)$, o qual é dado pela equação (49). Entretanto, conforme citado anteriormente, a inversão matricial de $P_i(k)$ adiciona um custo computacional muito alto. Para solucionar esse problema, aplicamos o teorema da inversão matricial (Haykin, 2001). Dessa forma, a equação (49) correspondente ao cálculo da matriz de covariância $P_i(k+1)$ que pode ser reescrita como:

$$P_i(k+1) = P_i(k) - \left(\frac{P_i(k)x^T(k+1)\omega(k+1)x(k+1)P_i(k)}{1 + \omega(k+1)x(k+1)P_i(k)x^T(k+1)} \right), \quad (51)$$

onde $x(k+1)$ é a $(k+1)$ -ésima linha da matriz $X(k)$ e $\omega(k+1)$ é o $(k+1)$ -ésimo elemento da matriz diagonal $Q_i(k+1)$.

8.2 Apêndice B: Taxa Ótima de Controle de Fluxos

Sendo a estimativa $\hat{b}(k+d)$ do tamanho da fila no *buffer* dada pelo preditor nebuloso proposto equivalente ao valor esperado para o tamanho da fila no *buffer* d passos à frente $b(k+d)$, podemos escrever a função custo (31) da seguinte forma:

$$J(k+d) = \frac{1}{2} \left[(\hat{b}(k+d) - b^\tau)^2 + \lambda \mu^2(k) \right]. \quad (52)$$

A taxa de controle de tráfego ótima pode ser obtida através da minimização da função custo $J(k+d)$. Para isso, diferenciamos a equação (52) com respeito a $\mu(k)$ e obtemos a seguinte igualdade:

$$\left[\hat{b}(k+d) - b^\tau \right] \frac{\partial \hat{b}(k+d)}{\partial \mu(k)} + \lambda \mu(k) = 0 \quad (53)$$

O termo $\frac{\partial \hat{b}(k+d)}{\partial \mu(k)}$ em (53) corresponde a derivada da equação (32) em relação a $\mu(k)$, o que resulta em:

$$\frac{\partial \hat{b}(k+d)}{\partial \mu(k)} = \sum_{i=1}^C \beta_{p+q+2}^i h^i(k). \quad (54)$$

A equação (53) pode ser resolvida com base nas equações do preditor proposto. Assim, substituindo as equações (54) e (32) na equação (53), obtém-se:

$$\left[\left(\sum_{i=1}^C b^i(k+d)h^i(k) \right) - b^\tau \right] + \frac{\lambda}{\sum_{i=1}^C \beta_{p+q+2}^i h^i(k)} \mu(k) = 0. \quad (55)$$

Note que $b^i(k+d)$ é dado pela equação (34). Isolando o termo $\beta_{p+q+2}^i \mu(k)$, a equação (34) pode ser reescrita como:

$$\begin{aligned} b^i(k+d) &= \alpha_0^i + \sum_{t=1}^{p+1} \alpha_t^i b(k-p-1+t) + \\ &+ \sum_{t=1}^q \beta_{t+p+1}^i \mu(k-q-1+t) + \beta_{p+q+2}^i \mu(k). \end{aligned} \quad (56)$$

Substituindo (56) em (55), e isolando o termo $\mu(k)$, obtemos a taxa ótima de controle para regular a taxa da fonte controlável, dada por:

$$\begin{aligned} \mu^o(k) &= \beta_0(k) \times \left(b^\tau - \sum_{i=1}^C (\alpha_0^i + \sum_{t=1}^{p+1} \alpha_t^i b(k-p-1+t) + \right. \\ &+ \left. \sum_{t=1}^q \beta_{t+p+1}^i \mu(k-q-1+t)) h^i(k) \right), \end{aligned} \quad (57)$$

onde

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

REFERÊNCIAS

- Adas, A. M. (1998). Using adaptive linear prediction to support real-time VBR video under RCBP, *IEEE/ACM Trans. Net.* **6**(5): 635–645.
- Aikat, J., Kaur, J., Smith, F. D. and Jeffay, K. (2003). Variability in TCP round-trip times, *3rd ACM SIGCOMM Conference on Internet Measurement*, Miami Beach, FL, USA, pp. 279 – 284.
- Assunção, P. A. and Ghanbari, M. (2000). Buffer analysis and control in CBR video transcoding, *IEEE Transactions on Circuits and Systems for Video Technology*, **10**(1): 83–92.

- Baldi, M. and Ofek, Y. (2000). End-to-end delay analysis of videoconferencing over packet-switched networks, *IEEE/ACM Transactions on Networking* **8**(4): 479 – 492.
- Banerjee, G., Rosenberry, R. and Sidhu, D. (2000). Performance Analysis of IP Switching and Tag Switching, *Lecture Notes in Computer Science - LNCS* **1815/2000**: 752–762.
- Barreto, P. S. and Lemos, R. P. (2001). ATM Traffic Prediction Using Artificial Neural Networks and Wavelet Transforms, *Lecture Notes in Computer Science - LNCS* **2094/2001**.
- Basar, S. L. and Srikant, T. R. (2005). Exponential-RED: a stabilizing AQM scheme for low- and high-speed TCP protocols, *IEEE/ACM Transactions on Networking* **13**(5): 1068– 1081.
- Bianchi, G. R., Vieira, F. H. T. and Lee, L. L. (2004). A Novel Network Traffic Predictor Based on Multifractal Characteristic, *IEEE Globecom*, Dallas, Texas, EUA.
- Boggia, G., Camarda, P., Grieco, L. A. and Mascolo, S. (2007). Feedback-based control for providing real-time services with the 802.11e MAC, *IEEE/ACM Transactions on Networking* **15**(2): 323–333.
- Bortolossi, H. J. (2002). *Cálculo Diferencial a Várias Variáveis - Uma Introdução A Teoria de Otimização*, Loyola.
- Chen, B., Liu, X. and Tong, S. (2007). Adaptive Fuzzy Output Tracking Control of MIMO Nonlinear Uncertain Systems, *IEEE Transactions on Fuzzy Systems* **15**(2): 287 – 300.
- Chen, B. S., Peng, S. C. and Wang, K. C. (2000). Traffic Modeling, Prediction and Congestion Control for High-Speed Networks: A Fuzzy AR Approach, *IEEE Transactions on Fuzzy Systems* **8**(5): 491–508.
- Chen, B.-S., Yang, Y. S., Lee, B.-K. and Lee, T.-H. (2003). Fuzzy adaptive predictive flow control of ATM network traffic, *IEEE Transactions on Fuzzy Systems* **11**(4): 568–581.
- Chen, T. M., Liu, S. S. and Salamam, V. K. (1996). The Available Bit Rate Service for Data in ATM Networks, *IEEE Communications Magazine* **34**(5): 56–58, 63–71.
- Cherry, S. (2005). Seven myths about voice over IP, *IEEE Spectrum* **42**(3): 52–57.
- Chong, S., Li, S. Q. and Ghosh, J. (1995). Predictive dynamic bandwidth allocation for efficient transport of real-time VBR video over ATM, *IEEE Journal on Selected Areas in Communications* **13**(1): 12–23.
- Dangy, T. D., Molnár, S. and Maricza, I. (2003). Queueing Performance Estimation for General Multifractal Traffic, *International Journal of Communication Systems* **16**: 117–136.
- Ditze, M. and Jahnich, I. (2005). Towards end-to-end QoS in service oriented architectures, *IEEE International Conference on Industrial Informatics INDIN '05*, pp. 92 – 97.
- Durresi, A., Sridharan, M. and Jain, R. (2006). Congestion control using adaptive multilevel early congestion notification, *International Journal of High performance Computing and Networking* **4**(5).
- Gevros, P., Crowcroft, J., Kirstein, P. and Bhatti, S. (2001). Congestion Control Mechanisms and the Best Effort Service Model, *IEEE Network* **15**: 16–26.
- Habib, I. W. and Saadawi, T. N. (1992). Access Flow Control Algorithms in Broadband Networks, *Military Communications Conference* **1**: 252–256.
- Hatano, T., Shigeno, H. and Okada, K. (2007). TCP-friendly Congestion Control for HighSpeed Network, *IEEE International Symposium on Applications and the Internet (SAINT)*, pp. 10–14.
- Haykin, S. (2001). *Adaptive Filter Theory*, 4 edn, Prentice Hall.
- Hong, D. P., Albuquerque, C., C.Oliveira and Suda, T. (2001). Evaluating the impact of emerging streaming media applications on TCP/IP performance, *IEEE Communications Magazine* **39**(4): 76–82.
- Hu, R. Q. and Petr, D. W. (2000). A Predictive Self-tuning Fuzzy-logic Feedback Rate Controller, *IEEE/ACM Transactions on Networking* **8**(6): 697–709.
- Iliev, B., Kalaykov, B. and R.Tero (2000). DSP-Based Fast Fuzzy Logic Controllers, <http://focus.ti.com/pdfs/univ/04-Control.pdf>.
- Imer, O. C., Compans, S., Basar, T. and Srikant, R. (2001). Available bit rate congestion control in ATM networks: developing explicit rate control algorithms, *IEEE Control Systems Magazine* **21**(1): 38–56.
- Jacobson, V. (1995). Congestion Avoidance and Control, *ACM SIGCOMM Computer Communication Review* **25**: 157–187.
- Juang, C.-F., Chen, J.-S. and Huang, H.-J. (2004). Temperature control by hardware implemented recurrent fuzzy controller, *IEEE International Conference on Fuzzy Systems*, Budapest, Hungria, pp. 795 – 799.

- Karnik, A. and Kumar, A. (2005). Performance of TCP congestion control with explicit rate feedback, *Networking, IEEE/ACM Transactions on* **13**(1): 108–120.
- Kharaajoo, M. J. (2004). *A Novel High Performance Fuzzy Controller Applied to Traffic Control of ATM Networks*, Vol. 3066, Springer Berlin / Heidelberg, pp. 327–333.
- Lee, I. W. C. and Fapojuwo, A. O. (2005). Stochastic processes for computer network traffic modeling, *Computer Communications* **29**: 1–23.
- Lee, T.-J. and Veciana, G. D. (2000). Model and performance evaluation for multiservice network link supporting ABR and CBR services, *IEEE Communications Letters* **4**(11): 375 – 377.
- Nejad, H. V., Yaghmaee, M. H. and Tabatabaee, H. (2006). Fuzzy TCP: Optimizing TCP Congestion Control, *Asia-Pacific Conference on Communications (APCC '06)*, pp. 1–5.
- Ouyang, C.-S., Lee, W.-J. and Lee, S.-J. (2005). A TSK-type neurofuzzy network approach to system modeling problems, *IEEE Transactions on Systems, Man and Cybernetics, Part B* **35**(4): 751–767.
- Papoulis, A. (2002). *Probability, Random Variables and Stochastic Processes*, fourth edn, McGraw-Hill, New York.
- Perlingeiro, F. R. and Ling, L. L. (2005). Estudo de estimação de banda efetiva para tráfego auto-similar com variação infinita, *Simpósio Brasileiro de Telecomunicações*, Campinas, pp. 326–331.
- Pitsillides, A. and Lambert, J. (1997). Adaptive congestion control in ATM based network: quality of service and high utilization, *Computer Communications* **20**: 1239–1258.
- Ramakrishnan, K. K. and Jain, R. (1990). A binary feedback scheme for congestion avoidance in computer networks, *ACM Transactions on Computer Systems* **8**(2): 158 – 181.
- Sugeno, M. and Yasukawa, T. (1993). A Fuzzy-Logic-Based Approach to Qualitative Modeling, *IEEE Trans. Fuzzy Systems* **1**: 7–31.
- Surmann, H., Ungerling, A. and Goser, K. (2006). *Optimized fuzzy controller architecture for field programmable gate arrays*, Lecture Notes in Computer Science.
- Takagi, T. and Sugeno, M. (1985). Fuzzy Identification of Systems and Its Applications to Modeling and Control, *IEEE Trans. Syst. Man and Cybernetics* **SMC-15**: 116–132.
- Toreyin, B. U., Trocan, M. and Cetin, A. E. (2006). LMS based adaptive prediction for scalable video coding, *ICASSP*, Toulouse, France.
- Vieira, F. H. T. and Lee, L. L. (2003). Alocação Dinâmica de Taxa de Transmissão em Redes de Pacotes Utilizando Redes Neurais Recorrentes Treinadas com Algoritmos em Tempo Real, *IEEE Latin America* **1**.
- Wang, C., Liu, J., Sohraby, K. and Hou, Y. (2007). LRED: A Robust and Responsive AQM Algorithm Using Packet Loss Ratio Measurement, *Parallel and Distributed Systems, IEEE Transactions on* **18**(1): 29–43.
- Wang, F., Yang, H., Qiao, Z. and Wang, X. (2006). Adaptive AQM Scheme of Internet Congestion Control Systems, *The Sixth World Congress on Intelligent Control and Automation WCICA 2006.*, pp. 4475 – 4478.
- Wenshan, H., P.Liu, G. and Rees, D. (2006). Design and implementation of networked predictive control systems based on round trip time delay measurement, *American Control Conference*.
- Zhang, H., Yang, M., Yang, O. W. and Mouftah, H. (2000). Design of proportional congestion control for high-speed network switch with variable delays, *IEEE International Conference on Communications ICC*, New Orleans, LA, USA, pp. 1435 – 1439.