Redes Neurais na Estimativa da Capacidade Requerida em Comutadores ATM

Miguel Franklin
miguel@lia.ufc.br

Adriano Nascimento
blau@lia.ufc.br

Marcelino Pequeno
marcel@lia.ufc.br

Mauro Oliveira
mauro@etfce.br

LAR - Laboratório Multiinstitucional de Redes e Sistemas Distribuídos
CEFET-CE - Centro Federal de Educação Tecnológica do Ceará
LIA - Laboratório de Inteligência Artificial
UFC - Universidade Federal do Ceará

Resumo

A *Capacidade Requerida* é a quantidade mínima de largura de banda que deve ser alocada a uma fonte de tráfego de modo a satisfazer os parâmetros de Qualidade de Serviço de todo o sistema. Esta informação pode ser utilizada como parâmetro para Controle de Admissão de Conexões (CAC) ou para Gerenciamento de Recursos. O objetivo deste trabalho é validar a utilização de Redes Neurais Artificiais para a estimativa da capacidade requerida em comutadores ATM. Com este intuito, foi desenvolvida uma abordagem específica baseada em parâmetros que definem o comportamento do tráfego agregado que chega a um comutador, em detrimento do uso de descritores de tráfego de aplicações em métodos analíticos.

Abstract

The Required Capacity, the minimum amount of bandwidth that must be alocated to a traffic source in order to grant the system's Quality of Service, can be used as parameter for Connection Admission Control (CAC) and Resource Management. The main purpose of this paper is to experiment and validate the usage of Artificial Neural Networks to estimate the required capacity on ATM networks, based on parameters that define the behavior of the aggregate traffic that reaches the switch, instead of using traffic descriptors on analytical methods.

Palavras-Chave: Redes ATM, Redes Neurais, CAC, *Equivalent Bandwidth*, Gerenciamento de Recursos.

1 Introdução

O Modo de Transferência Assíncrono (ATM - *Asynchronous Transfer Mode*) é a tecnologia utilizada no suporte às Redes Digitais de Serviços Integrados de Faixa Larga (RDSI-FL).

Uma das principais características do ATM é a possibilidade de multiplexar conexões de características distintas, com garantias de manutenção da Qualidade de Serviço das diversas aplicações. As fontes de tráfego existentes para conexões ATM apresentam basicamente dois comportamentos: taxa constante (fontes CBR) e taxa variável (fontes ABR e VBR) [12].

As fontes de tráfego CBR (*Constant Bit Rate*) já são bem compreendidas e estudadas, não representando maior complexidade para alocação de recursos e operação.

Por outro lado, as fontes de tráfego VBR (*Variable Bit Rate*) e ABR (*Avaliable Bit Rate*) incorporam maior complexidade devido a novas variáveis e requisitos. Um exemplo deste tipo de tráfego é observado no sistema de telefonia convencional com detecção de silêncio. Neste caso, o tráfego é gerado a uma taxa constante de 64 Kbps apenas enquanto o interlocutor fala. Nos momentos de silêncio, nenhum tráfego é gerado pela fonte [9]. Desta forma, pode-se observar que a caracterização do tráfego variável é bem mais complexa do que a do tráfego constante, que pode ser completamente descrito com a simples medida da taxa de transmissão na qual esta fonte opera.

A obtenção da capacidade requerida para uma fonte de tráfego individual e para o tráfego agregado é considerada uma tarefa complexa devido à natureza essencialmente estocástica do tráfego ATM. Diversos trabalhos têm apresentado métodos analíticos de estimativa deste valor [16] [17]. Métodos inteligentes, como Redes Neurais Artificiais, têm sido bastante utilizados para Gerência de Tráfego e Fluxo [10] [7] [3] [2]. Outros trabalhos [11] [5] utilizam Redes Neurais para o problema da estimativa da Capacidade Requerida através de mecanismos como Controle de Admissão de Conexões, por exemplo.

O cálculo da Capacidade Requerida de uma fonte de tráfego é geralmente baseado na sua caracterização. Entretanto, como o tráfego ATM (especialmente o variável) ainda não é bem compreendido [9], pode haver imprecisão na descrição dos parâmetros de uma aplicação por parte do projetista. Isto pode levar a um cálculo deturpado da capacidade requerida. Assim, faz-se necessário um método que leve em conta o comportamento *real* do tráfego em detrimento à sua caracterização teórica.

Este trabalho descreve a experimentação que valida a utilização de Redes Neurais Artificiais para a estimativa da Capacidade Requerida em comutadores ATM. Com este intuito, foi desenvolvida uma abordagem específica baseada em parâmetros que definem o comportamento geral do tráfego agregado que chega ao comutador. Para este ambiente de experimentação foi implementada a arquitetura RENATA (**Re**des **N**eurais **A**plicadas ao **T**ráfego **A**TM) [12].

Este documento está organizado da seguinte forma. A Seção 2 aborda generalidades sobre Tráfego ATM, com destaque à multiplexação estatística e à capacidade requerida. A Seção 3 descreve a arquitetura RENATA, o ambiente de experimentação prototipado. Na Seção 4, uma experimentação baseada em Redes Neurais acerca do problema abordado é proposta e modelada. A Seção 5 discute os resultados obtidos, enquanto a Seção 6 apresenta conclusões sobre este trabalho e propostas para trabalhos futuros.

2 O Tráfego ATM

Um dos maiores problemas para o controle de tráfego em uma rede ATM é a enorme variedade de tipos de aplicação que podem ser suportadas, mais notadamente em relação ao tráfego multimídia [8]. Cada usuário apresenta uma característica de tráfego distinta e os serviços de comunicação têm diferentes requisitos de Qualidade de Serviço (QoS) [7].

Toda esta variedade torna a tarefa de gerenciar tráfego em uma rede ATM bem mais complexa do que em redes convencionais.

Para o estabelecimento de uma nova conexão em uma rede ATM, um contrato de tráfego deve ser estabelecido entre esta aplicação e a rede. Os parâmetros que regem este contrato incluem as características de QoS requeridas e o descritor de tráfego da aplicação. Usualmente, todos os esquemas de gerenciamento de tráfego ATM se utilizam desta caracterização do tráfego que é apresentada à rede neste contrato [1].

2.1 Caracterização de Tráfego

Um dos requisitos para a negociação de novas conexões em redes ATM é a descrição do tráfego gerado por esta aplicação. Estes parâmetros variam de acordo com a classificação do tráfego.

A taxa de transmissão em uma conexão do tipo CBR permanece constante durante toda a sua duração. Este tipo simples de tráfego pode ser facilmente caracterizado por sua taxa de transmissão média (SBR - *Sustainable Bit Rate*).

O tráfego do tipo VBR apresenta uma natureza mais complexa do que a do CBR. A taxa de transmissão de uma fonte VBR varia ao longo do tempo. Neste caso, considera-se que uma conexão VBR possui mais de um estado, onde cada estado é usualmente caracterizado pela sua taxa de transmissão e a seu tamanho médio.

Há ainda uma outra variedade de tráfego variável, que é denominada ABR (*Avaliable Bit Rate*), onde a taxa de transmissão durante a conexão varia não de acordo com a necessidade da aplicação e sim de acordo com a disponibilidade de recursos na rede.

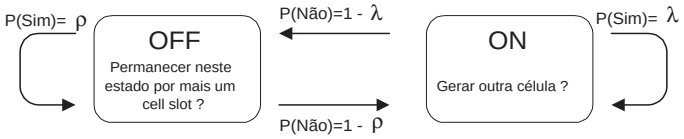
No escopo deste trabalho, foram consideradas apenas aplicações VBR caracterizadas como ON-OFF, *i.e.*, que apresenta somente dois estados: transmitindo a taxa de pico (período ativo) ou sem nenhum tráfego (período inativo). O comportamento de uma fonte de tráfego VBR ON-OFF pode ser representado por uma Cadeia de Markov de dois estados, com diagrama de estados ilustrado na Figura 1(a), onde ρ e λ representam as probabilidades da fonte entrar em período ativo e inativo, respectivamente. Fontes de tráfego com durações dos períodos ativo e inativo modelados de acordo a distribuição exponencial negativa têm sido mais freqüentemente estudadas. Esta modelagem vem sendo aplicada para o tráfego de dados ou de qualquer variedade de tráfego em rajadas [14]. A Figura 1(b) exemplifica uma fonte de tráfego ON-OFF.

Para este tipo de tráfego, os parâmetros mais usuais para sua caracterização são: Taxa de Transmissão de Pico (PCR), Tamanho médio do período ativo (t^{on})[1] e Tamanho médio do período inativo (t^{off}).

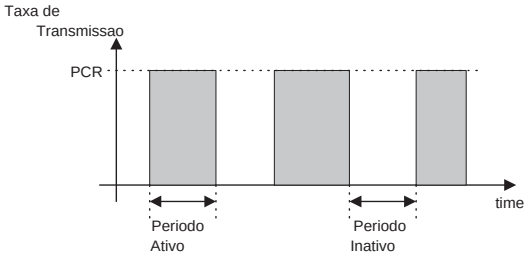
2.2 Capacidade Requerida

Uma fonte de tráfego ATM pode variar a sua taxa de transmissão ao longo do tempo. Por isso, se a alocação de recursos para cada aplicação for definida pela sua taxa de transmissão de pico (PCR) haverá momentos de desperdício, pois a quantidade de largura de banda deixada ociosa por uma aplicação em um determinado momento poderia ser reaproveitada por outras aplicações. Assim, a *Multiplexação Estatística* permite que seja alocada a cada fonte de tráfego uma quantidade de largura de banda menor que sua taxa de pico, com o objetivo de fazer o aproveitamento dos recursos deixados ociosos pelas aplicações de taxa

[1] Também conhecido como tamanho da rajada.



(a) Diagrama de Estados



(b) Exemplo de Fonte ON-OFF

Figura 1: Fontes de Tráfego VBR ON-OFF

variável nos momentos inativos. Para tanto, mecanismos estatísticos são utilizados para garantir a Qualidade de Serviço desejada para todo o sistema.

A capacidade requerida é, portanto, a quantidade mínima de largura de banda que deve ser alocada a uma fonte de tráfego de modo a satisfazer os parâmetros de Qualidade de Serviço de todas as fontes de tráfego envolvidas no nó. O ganho devido à multiplexação estatística é, então, uma medida da economia de recursos que se obtém com a sua adoção. Apesar deste artifício expor o sistema a uma circunstância de congestionamento, onde a taxa de transmissão agregada supera a capacidade de saída do comutador até que haja um estouro de ocupação no *buffer*, a probabilidade disto ocorrer é mantida pelo mecanismo de estimativa da capacidade requerida abaixo de um certo limiar ε. Este valor representa um dos requisitos de Qualidade de Serviço do sistema, o CLP (*Cell Loss Probability*).

O cálculo do ganho devido à multiplexação estatística é obtido pela diferença proporcional entre a taxa de pico de uma aplicação e sua capacidade requerida. Portanto, quanto menor é a capacidade requerida em relação à taxa de pico, maior é o ganho estatístico. Por exemplo, se uma aplicação j tem como características sua taxa de pico PCR_j e sua capacidade requerida CR_j. Então, o ganho estatístico GE_j desta aplicação é dado como:

$$GE_j = 1 - \frac{CR_j}{PCR_j}$$

Então, o ganho médio para todas as aplicações que compõe o tráfego agregado é definido por:

$$GE = \frac{\sum_{j=1}^{N} GE_j}{N}$$

O ganho devido à Multiplexação Estatística é influenciado por diversas variáveis. Dentre elas estão a descrição do comportamento das conexões, o tamanho do *buffer* do comu-

tador em questão e a capacidade dos enlaces de saída [14].

Uma das grandes dificuldades de se encontrar um valor para a capacidade requerida de uma conexão é ter uma descrição o mais confiável possível de seu tráfego gerado. Isto nem sempre é conseguido, visto que o tráfego ATM ainda não é muito bem desvendado [9]. Portanto, por mais aproximado que seja o método de estimativa da capacidade requerida, uma caracterização errônea do tráfego pode levar tanto a um valor superestimado – causando desperdício de recursos – quanto a um valor subestimado – podendo afetar a Qualidade de Serviço de todo o sistema.

2.3 O Método Equivalent Bandwidth

Diversas abordagens tentam estimar o ganho obtido pela multiplexação estatística. O método *Equivalent Bandwidth* (ou *Equivalent Capacity*), abreviado como EB, proposto em [4], faz a estimativa da capacidade requerida de cada conexão individual, utilizando para isto as seguintes informações:

- Informações das Aplicações (para cada aplicação j):
  - Taxa de transmissão de pico PCR_j, tamanho médio do período ativo t_j^{on}, tamanho médio do período inativo t_j^{off} e taxa de utilização da fonte $\rho_j = \frac{t_j^{on}}{t_j^{on}+t_j^{off}}$
- Informações do Sistema:
  - Tamanho do buffer ξ

Assim, a Capacidade Requerida de uma fonte de tráfego j segundo o método *Equivalent Bandwidth* é estimada como:

$$EB_j = PCR_j \times \frac{y_j - \xi + \sqrt{(y_j - \xi)^2 + 4\xi\rho_j y_j}}{2y_j} \tag{1}$$

onde

$$y_j = \alpha t_j^{on}(1-\rho_j)PCR_j \quad \text{e} \quad \alpha = ln(1/\varepsilon)$$

Uma das características do método *Equivalent Capacity* é a sua escalabilidade. Desta forma, o valor *Equivalent Bandwidth* do tráfego agregado é obtido do somatório dos EBs das fontes de tráfego individuais [13].

$$EB = \sum_{j=1}^{N} EB_j$$

O método *EB* é considerado como uma forma rápida e suficientemente precisa para o cálculo da capacidade requerida. Entretanto, a utilização de uma caracterização de tráfego equivocada, seja sub ou superestimada, pode levar a uma valor indesejado da capacidade requerida. Portanto, surge a necessidade de um método que leve em conta principalmente o comportamento real do tráfego, em detrimento de descritores de tráfego que podem deturpar o cálculo da capacidade requerida.

3 A Arquitetura RENATA

RENATA (**RE**des **N**eurais **A**plicadas ao **T**ráfego **A**TM) é uma arquitetura desenvolvida há dois anos no LAR (Laboratório Multiinstitucional de Redes e Sistemas Distribuídos) [12] destinada à gerência pró-ativa de redes ATM. Este protótipo faz uso de Redes Neurais como componente inteligente.

O ambiente é composto por diversos módulos, que interoperam com o intuito de apresentar, no final do processo, um módulo neural acoplado a um módulo de gerência, que deverá realizar o monitoramento e o controle sobre os recursos de rede.

O módulo neural da RENATA se utiliza de informações provenientes de uma simulação da rede ATM para que possa ser realizado o seu treinamento. Isto permite uma maior flexibilidade para testes em estados críticos [14].

Após treinada, a rede neural que compõe o Módulo Neural é acoplada a um Módulo de Gerência, que fará com que o sistema possa se comunicar com a rede real, deixando o ambiente de simulação.

A arquitetura funcional da RENATA é ilustrada na Figura 2.

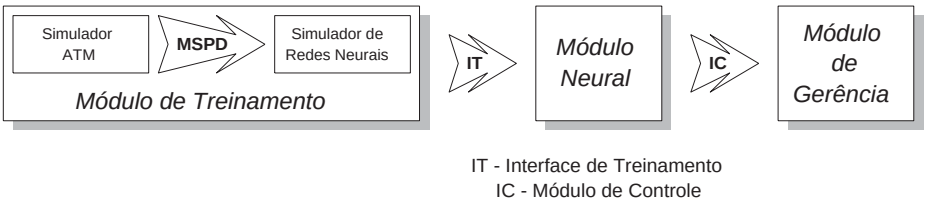


Figura 2: Arquitetura Funcional da RENATA

Os módulos da arquitetura RENATA serão brevemente descritos a seguir.

3.1 Módulo de Treinamento

O *Módulo de Treinamento* é composto por um simulador de redes ATM, um *Módulo de Seleção e Preparação de Dados* (MSPD) e um simulador de redes neurais. O simulador de redes ATM deverá receber as características de multiplexação da rede real como entrada. Estas informações incluem a descrição dos parâmetros do comutador utilizado e dados acerca de número e descrição de fontes de tráfego. A partir da simulação deste modelo, serão produzidos arquivos de *log*, que conterão todas as informações acerca dos resultados do período simulado.

O *Módulo de Seleção e Preparação de Dados* (MSPD) contém as informações fornecidas sobre a rede ATM simulada e filtra os parâmetros necessários acerca do comportamento do tráfego agregado para que seja formado um banco de exemplos. Este conjunto irá servir como entrada para o simulador de Redes Neurais quando do treinamento do Módulo Neural.

O simulador de Redes Neurais recebe como entrada o produto do Módulo de Seleção e Preparação de Dados (MSPD). Será, portanto, um banco de exemplos que irá servir de base para o treinamento supervisionado da rede neural definida. No final do processo de treinamento, são realizados testes e validação do resultado obtido. É produzido, ao final do processo de treinamento, um código em linguagem C que representa a implementação

da rede neural *treinada*, que irá compor o *Módulo Neural*. Este código é incorporado ao Módulo Neural através da *Interface de Treinamento* (IT).

3.2 Módulo Neural

O *Módulo Neural* é basicamente composto pelo código em linguagem C produzido pelo simulador de redes neurais, que implementa a rede neural treinada. Além disto, este módulo possui uma interface chamada *Interface de Controle* (IC), que propicia a comunicação com o Módulo de Gerência, fornecendo o acesso aos recursos de rede reais.

3.3 Módulo de Gerência

O *Módulo de Gerência* é responsável por oferecer ao Módulo Neural comunicação com os recursos de rede. Esta comunicação poderá ser realizada através de algum protocolo Gerente × Agente, como o SNMP (*Simple Network Management Protocol*) ou através de alguma interface específica ao comutador. Através deste módulo, as informações que alimentam o Módulo Neural são obtidos, assim como serve como agente modificador dos recursos de rede, realizando as operações sugeridas pelo Módulo Neural.

4 Experimentação Baseada em Redes Neurais

A experimentação aqui descrita consiste em modelar uma rede neural capaz de realizar a estimativa da capacidade requerida do tráfego agregado em uma rede ATM. Para isto, utiliza-se como entrada não os descritores de tráfego das fontes e sim parâmetros que podem descrever o comportamento do tráfego agregado.

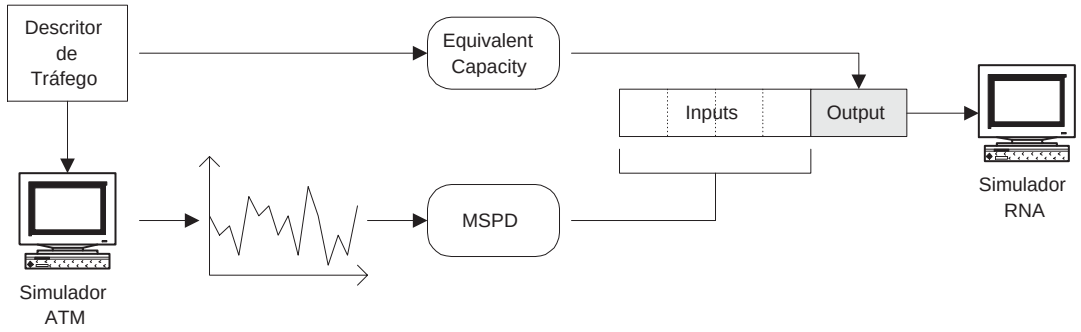


Figura 3: Esquema de Geração do Banco de Exemplos

Frente ao problema exposto, assume-se:

- O mesmo parâmetro de Qualidade de Serviço CLP (*Cell Loss Probability*) para todas as aplicações como $\varepsilon = 10^{-5}$;
- A existência de apenas conexões do tipo VBR ON-OFF;
- As fontes de tráfego têm tamanhos de estados (ON e OFF) distribuídos exponencialmente;

- Os intervalos entre chegadas de células de uma fonte de tráfego seguem a Distribuição de Poisson;

O banco de exemplos que servirá de base para o treinamento da rede neural é composto por um conjunto de valores de entrada (*inputs*) e um valor de saída (*output*). Os *inputs* representam as variáveis que descrevem a configuração que se deseja inferir, enquanto o *output* representa a própria classificação desta configuração.

Métodos de estimativa da capacidade requerida freqüentemente utilizam os descritores de tráfego das aplicações como *inputs*, através dos quais métodos analíticos apresentam o *output* correspondente. Na abordagem apresentada neste trabalho, o descritor de tráfego das aplicações é utilizado apenas para a composição dos *outputs* dos exemplos, sendo os *inputs* obtidos a partir da simulação das aplicações descritas. Desta forma, este tipo de estimativa não é afetada pela imprecisão dos descritores de tráfego, pois a classificação é gerada a partir do comportamento simulado destas aplicações.

Assim sendo, o mapeamento entre os parâmetros de *input*, que caracterizam o comportamento do tráfego agregado, e a estimativa da capacidade requerida será realizada por uma Rede Neural Artificial. A escolha deste meio se deve à inexistência de um método analítico para esta função, e por causa do requisito de tempo-real para esta classificação.

O processo de adaptação da arquitetura RENATA ao problema proposto consiste de duas fases:

- Projeto da Rede Neural
- Implementação do Módulo de Treinamento

Cada fase será descrita a seguir.

4.1 Projeto da Rede Neural

Uma das fases mais relevantes do projeto de uma rede neural é a definição das variáveis dos vetores de entrada e de saída da rede neural. Para cada problema, deve existir um conjunto de valores que represente o mais fielmente possível o estado real que se deseja classificar ou inferir sobre. Portanto, para que uma rede neural possa interpolar uma função o mais confiavelmente possível, deve-se definir sua(s) variáveis(s) e representá-las no vetor de entrada da RN, com os respectivos valores desejados de saída.

Para o problema proposto, considera-se um comutador caracterizado por sua capacidade máxima de conexões N_{max} e por sua capacidade de buffer ξ. Em um determinado momento, este equipamento multiplexa N fontes de tráfego advindas de enlaces de entrada, os quais totalizam uma capacidade de L_{in}. Todo o tráfego destas fontes deverão ser roteadas para um mesmo enlace de saída de capacidade L_{out}. Tais aplicações (fontes) são caracterizadas pelo tráfego VBR ON-OFF. O processo de chegada de células destas aplicações segue o Modelo Estocástico de Poisson. Consequentemente, os tamanhos de cada estado (ON e OFF) seguem uma distribuição exponencial. Cada fonte de tráfego n é caracterizada por sua taxa de transmissão de pico P_n e pelos seus tamanhos médios dos períodos ativo (t^{on}) e inativo (t^{off}), com $n \leq N \leq N_{max}$, sendo que o valor máximo para a taxa de transmissão de pico de aplicações é representada por P_{max}.

A rede neural proposta se utiliza de informações obtidas a partir de leituras sobre a taxa de transmissão do tráfego agregado em momentos anteriores. A partir destas

informações, a rede neural deverá determinar a capacidade requerida agregada no instante presente, denotado por T_0.

Ao longo do tempo define-se *Pontos de Medição* ($MP's$) e *Pontos de Checagem* (CP's). Os MP's marcam os instantes onde as leituras acerca da taxa de transmissão agregada são realizadas. Estes pontos distam entre si em $\Delta\tau$. Os CP's são os MP's onde todas as informações são totalizadas. Os CP's são eqüidistantes entre si em ω intervalos $\Delta\tau$, isto é, entre dois CP's existem $(\omega - 1)$ MP's. Os valores coletados nos MP's são totalizados e processados no CP subseqüente. A distribuições de MP's e CP's ao longo do tempo é ilustrada na Figura 4. Denomina-se o instante presente como T_0, e os instantes anteriores,

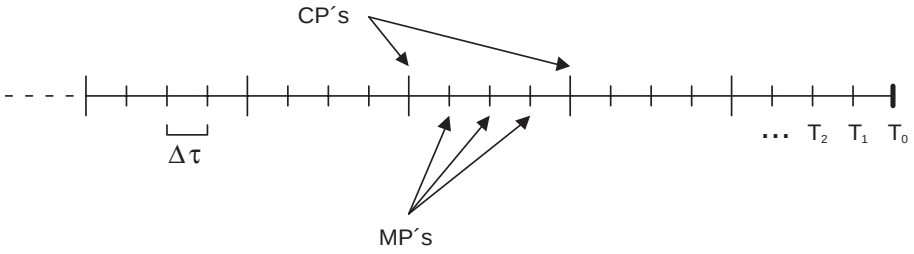


Figura 4: Diagrama de Pontos de Checagem e Medição

eqüidistantes entre si, como T_i, com $i = 0, 1, \cdots$. Assim:

$$T_i - T_{i+1} = \Delta\tau \quad \text{para qualquer } i \geq 0$$

Define-se *história* como sendo a medida do tempo durante o qual o sistema é observado para que a rede neural possa coletar as informações suficientes para compor um vetor de entrada. Este período pode ser caracterizado pela quantidade de CP's que o compõe. Portanto, se o intervalo entre dois CP's é de $\Delta\tau$, e o vetor de entrada da RN contém informações de h CP's, então o tráfego deverá ser observado e medido por um período de $h \times \omega \times \Delta T$ para que seja composto um vetor de entrada para a rede neural.

Seja R_t^j a taxa de transferência instantânea da aplicação j no momento T_t. Portanto, sendo C_δ^j o número de bits transmitidos pela fonte de tráfego j durante o intervalo δ, temos:

$$R_t^j = \lim_{\delta \to 0} \frac{C_\delta^j}{\delta}$$

Então, a *taxa de transmissão agregada instantânea* que chega ao comutador no momento T_t é expressa como:

$$R_t = \sum_{j=1}^{N} R_t^j$$

N é o número de aplicações de utilizazam o comutador para atingir um único enlace de saída.

Em cada MP, mede-se a taxa de transferência agregada instantânea e em cada CP os últimos ω valores medidos – incluindo o valor medido no próprio CP – são totalizados e processados.

Portanto, em um instante T_m onde $m \ mod \ \omega = 0$, isto é, o momento T_m é um CP, define-se σ_t como sendo o *desvio padrão* da taxa de transmissão agregada R_t, com t variando entre $[0; m]$.

$$\sigma_m = \sqrt{\frac{\sum_{t=0}^{m} (\bar{R}_m - R_t)^2}{m-1}}$$

onde $\bar{R}_m$ é a média das taxas de transmissão agregadas instantâneas nos momentos entre $[0; m]$.

$$\bar{R}_m = \frac{\sum_{t=0}^{m} R_t}{m}$$

Baseado nestes valores, define-se as seguintes funções:

$$f(t) = \frac{2 \times \bar{R}_t}{PCR_{max}} \qquad\qquad g(t) = \frac{4 \times \sigma_t}{PCR_{max}}$$

Onde $PCR_{max} = P_{max} \times N_{max}$

Assim sendo, os valores selecionados para compor o vetor $\mathcal{I}$ de entrada da rede neural são:

$$\mathcal{I} = \left(\frac{N}{N_{max}}, \frac{\sum P_j}{PCR_{max}}, \overbrace{f(1 \times \omega), g(1 \times \omega), \cdots, f(h \times \omega), g(h \times \omega)}^{h \times} \right)$$

Os valores N_{max}, PCR_{max} e $\sum P_j$, utilizados como denominadores no vetor de entrada, representam os fatores de normalização utilizados para que todos os valores se apresentem no intervalo $[0; 1]$.

O valor da classificação, representado pelo vetor de saída com uma posição, é obtido a partir da aplicação do método EB (Equação 1) nos descritores de tráfego utilizados para a simulação. Portanto, o vetor de saída $\mathcal{O}$ é definido como:

$$\mathcal{O} = \left(\frac{\sum_{j=1}^{N} EB_j}{L_{out}} \right)$$

O tipo de rede neural escolhida para este experimento é a *feed-forward*, utilizando para o processo de treinamento o algoritmo *Backpropagation Momentum* [15], que é uma otimização do método *Backpropagation* que desconsidera mínimos locais. A topologia escolhida consiste de 3 camadas de neurônios (entrada, intermediária e saída). O número de neurônios na entrada varia em função da história $(2 \times (h + 1))$. A camada de saída é fixa em 1 neurônio, representando o valor da Capacidade Requerida normalizada. Fez-se variar o número de neurônios na camada intermediária em busca de uma maior precisão.

A Figura 5 ilustra a rede neural projetada.

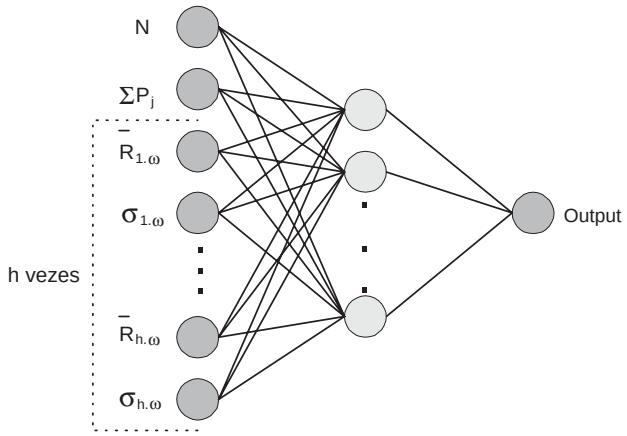


Figura 5: Parâmetros de Entrada da Rede Neural

4.2 Implementação do Módulo de Treinamento

Para fins de implementação, foram definidos limites para os valores dos descritores de tráfego. Desta maneira, os valores de PCR variam no intervalo de 0,1 a 2,5 $Mbps$, com granularidade de 0,01 $Mbps$; os valores de t^{on} se encontram entre 0,01 e 1,5 ms, com granularidade de 0,01 ms; e os valores de t^{off} estão compreendidos entre 0,01 e 1,5 ms.

Cada configuração é composta randomicamente e representa uma situação em que N fontes de tráfego atingem um comutador com capacidade total dos enlaces de entrada de $L_{in} = 155,52\,Mbps$. Portanto, uma configuração é formada por, no máximo, $N_{max} = 65$ aplicações (fontes de tráfego), cada qual com valores de PCR, t^{on} e t^{off} randomicamente gerados dentro dos intervalos supracitados.

Para a composição do banco de exemplos de treinamento da rede neural, foram geradas 3.500 configurações, sendo que 1.500 foram utilizadas para gerar o banco de exemplos de treinamento e o restante comporá o banco de exemplos de validação.

Cada configuração de rede é simulada por um período virtual de 1 s. A ferramenta utilizada foi o Simulador de Redes ATM do NIST (*National Institute of Standards and Technology*) [6]. O tempo médio de processamento de simulação para extração dos *logs* de 1 s de operação em 2000 configurações é de, em média, 10 horas em um supercomputador IBM SP-2 com 4 nós (CENAPAD-NE).

Para cada configuração simulada, um arquivo de *log* é gerado com o registro das taxas de transmissão agregadas instantâneas ao longo do tempo, com granularidade de 10 μs.

O MSPD (Módulo de Seleção e Preparação de Dados) recebe como entrada os arquivos de *log* produzidos pelo simulador de redes ATM. Este módulo é responsável por selecionar randomicamente intervalos ao longo do tempo de simulação para a coleta dos dados que, quando processados, comporão os vetores de entrada correspondentes. Uma configuração pode gerar mais de um exemplo. No caso desta experimentação, escolheu-se extrair apenas um exemplo de cada configuração. O produto do MSPD é um arquivo de padrões de treinamento e/ou validação compatível com a ferramenta de simulação de redes neurais adotada.

O simulador de redes neurais escolhido para o projeto, treinamento e validação da rede neural em questão é o SNNS (*Stuttgart Neural Network Simulator*) [15].

Após treinada e validada utilizando o simulador SNNS um código em linguagem C pode ser gerado, contendo um *stub* para a rede neural treinada.

5 Discussão dos Resultados

Para os experimentos aqui descritos as redes neurais foram treinadas com 1.500 exemplos, utilizando como critério de parada o ponto mínimo do somatório do erro quadrático (SSE - *Sum of Squared Error*) no conjunto de exemplos de validação. A medida SSE é descrita por:

$$SSE = \sum (o_{desj} - o_{obt})^2$$

Onde o_{desj} representa a saída desejada (banco de exemplos), e o_{obt} representa a saída obtida a partir da Rede Neural.

Em todas as experimentações foram utilizados 20 neurônios na camada de entrada, o que corresponde a uma história de 9 CP's; 6 neurônios na camada intermediária e um neurônio na camada de saída. Exceções foram feitas na Figuras 8 e 9, em que variou-se o número de neurônios nas camada de entrada e intermediária, respectivamente. Em todos os experimentos, os resultados são medidos no conjunto de validação compreendendo 2.000 exemplos, não utilizados no processo de treinamento.

A Figura 6 mostra a acuracidade da Capacidade Requerida obtida pela Rede Neural em relação à medida desejada, conseguida pelo método *Equivalent Bandwidth*. Para isto, computou-se o ganho estatístico médio para as capacidades requeridas obtidos pelo método EB e pela Rede Neural nos 2.000 exemplos de validação. A variação dos ganhos médios mantém-se pela ordem de grandeza de 10^{-3} para todas as medidas de *buffer* variando de 100 a 1.000 células. Vale-se notar que quanto maior é o tamanho do *buffer*, maior é a precisão dos resultados.

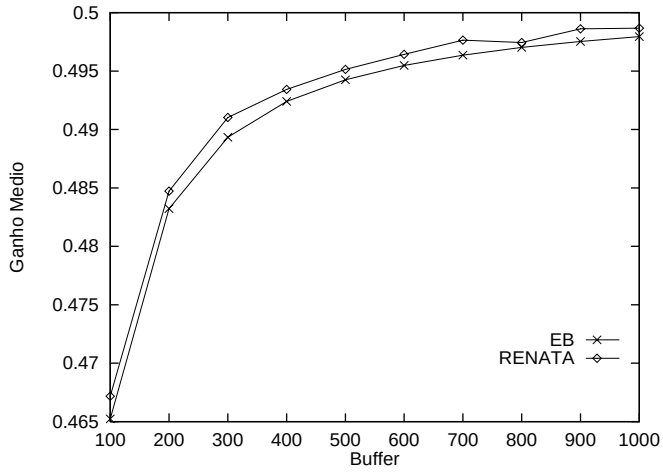


Figura 6: Gráfico Buffer × Ganho Estatístico Médio

A Figura 7 ilustra como o erro SSE diminui à medida que o tamanho do *buffer* cresce, embora, mesmo no pior caso, não ultrapasse a 2.35. Em termos práticos, em um *link*

de capacidade $51,84$ *Mbps*, para o EB desejado de $38,266$ *Mbps*, calculou-se o valor de $37,938$ *Mbps*, o que representa uma estimativa bastante razoável, visto que o EB é, em si, um método aproximativo.

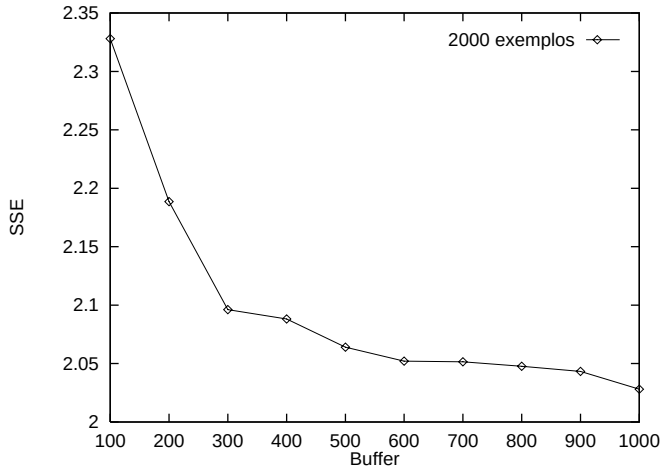


Figura 7: Gráfico Buffer × SSE

A Figura 8 mostra que o erro SSE diminui a medida que o número de MP's (história) aumenta. Entretanto, um aumento da história implica na necessidade de um maior tempo de observação do comutador ATM para que seja gerada a estimativa da capacidade requerida. Além disto, há um incremento no custo computacional, visto que o número de neurônios na camada de entrada aumenta de acordo com a história.

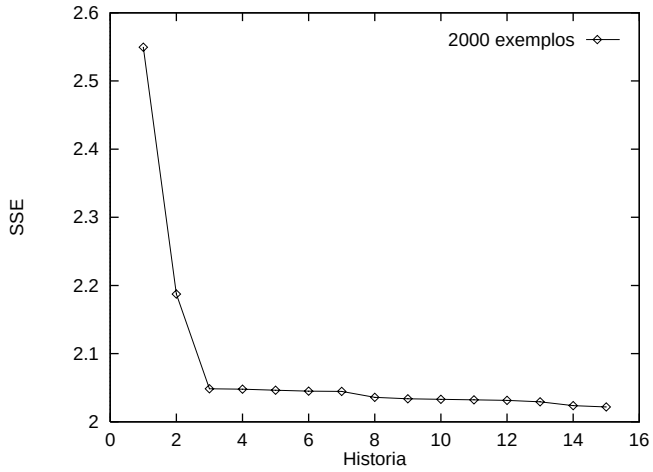


Figura 8: Gráfico História × SSE

A Figura 9 mostra as tentativas de se conseguir um número ideal de neurônios para a camada intermediária para um tamanho de *buffer* de 1.000 células. Observamos que o erro SSE oscila não-monotonicamente entre 2 e $2,1$ à medida que aumentamos o número de neurônios na camada intermediária. Resolveu-se, então, utilizar $N_i = 6$ em todos os experimentos.

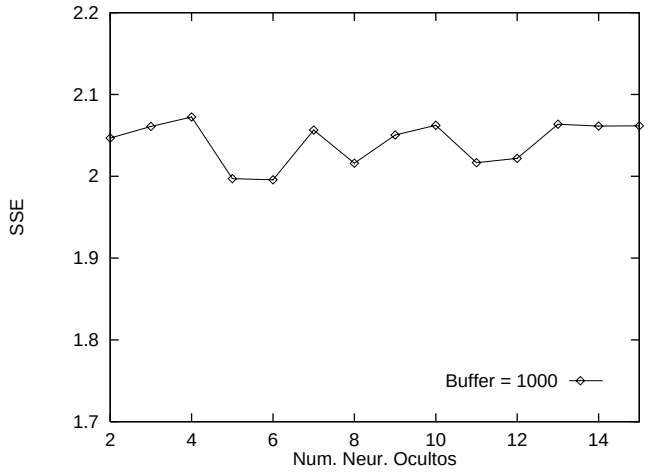


Figura 9: Gráfico Num. de Neurônios Ocultos × SSE

6 Conclusões

A grande vantagem da nova proposta de gerenciamento idealizada na arquitetura RENATA consiste na possibilidade de antecipar situações reais de interesse dos mecanismos de gerenciamento através do uso de simulação para a geração de *baselines*. A utilização de Redes Neurais nesta arquitetura ajuda sobremaneira, baseado em resultados simulados, na tomada de decisão acerca de situações reais. Generalização, tempo-real e precisão de estimativa são características importantes inerentes às Redes Neurais, sem as quais os métodos de gerência proativa tornam-se ineficazes.

Neste contexto proativo que caracteriza a ação de gerenciamento da RENATA, tem-se que a estimativa da Capacidade Requerida em comutadores ATM exige uma abordagem específica baseada nesta arquitetura. Esta abordagem resultou na experimentação objeto deste trabalho, permitindo ilustrar de forma concreta a importância da idéia proposta na arquitetura RENATA. Percebe-se, pelos resultados obtidos descritos na Seção 5, que as Redes Neurais são capazes de atingir, com grau de precisão aceitável, um valor estimado da capacidade requerida em um comutador ATM, utilizando como base o comportamento real de seu tráfego agregado. Assim, os descritores de tráfego tornam-se parcialmente dispensáveis, a medida que apenas a taxa de transmissão de pico (PCR) é de interesse da abordagem proposta, sendo todas as demais informações obtidas através de medição de tráfego.

Visualiza-se como próximas etapas da experimentação realizada as seguintes pesquisas: escolha de melhores fatores de normalização, generalização de seu campo de atuação e comparação desta abordagem com métodos analíticos de estimativa da Capacidade Requerida. Estes trabalhos estão sendo desenvolvidos no *Département Réseaux et Services de Telecommunications* do INT (*Institut National des Télécommunications*), na França, no contexto do projeto Neuraltel (*Neural Networks on Telecommunications*). Este projeto, que pertence ao Programa ALFA (*Amérique Latine - Formation Académique*) da Comunidade Européia, se interessa pelo uso de Redes Neurais em Telecomunicações, tendo como parceiros a UFC (Universidade Federal do Ceará) e o CEFET-CE (Centro Federal de Educação Tecnológica do Ceará), além do próprio INT.

Referências

[1] Martin Bernhardt. Design and Implementation of a Web Based Tool for ATM Connection Management. Master's thesis, International Computer Science Institute, Berkeley University, August 1996.

[2] József Biró, Zoltán Koronkai, Tibor Trón, Miklós Boda, András Faragó, and Tamás Henk. Neurocomputing in Logical Partitioning of ATM Networks. *Journal on Communications, Special Issue on ATM Networks II*, 47:7–11, 1995.

[3] Peter K. Campbell, Alan Christiansen, Michael Dale, Herman L. Ferrá, Adam Kowalczyk, and Jacek Szymanski. Experiments with Simple Neural Networks for Real-Time Control. *IEEE J. on Selected Areas in Communications*, 15(2):165–178, Feb 1997.

[4] A. Elwalid and D Mitra. Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High-Speed Networks. *IEEE/ACM Transactions on Networking*, 1(3):329–343, Jun 1993.

[5] Olle Gallmo, Ernst Nordstrom, Mats Gustafsson, and Lars Asplund. Neural Networks for Preventive Traffic Control in Broadband ATM Networks. In *International Workshop on Mechatronical Computer Systems for Perception and Action (MCPA '93)*, pages 139–145, Halmstad, Sweden, Jun 1993.

[6] Nada Golmie, Frederic Mouveaux, Lance Hester, Yves Saintllan, Alfred Koenig, and David Su. *The NIST ATM/HFC Network Simulator: Operation and Programming Guide - Version 4.0.* NIST - National Institute of Standards and Technology - U.S. Department of Commerce, Dec 1998.

[7] Atsushi Hiramatsu. *Handbook of Neural Computing*, chapter ATM Network Control by Neural Network. Institute of Phisics Publishing and Oxford University Publishing, 1997.

[8] Raj Jain. Congestion Control and Traffic Management in ATM Networks: Recent Advances and A Survey. In *Computer Networks and ISDN Systems*, February 1995.

[9] Daniel Minoli and Thomas Golway. *Planning & Managing ATM Networks.* Ed. Manning, Feb 1997.

[10] Shane Naughton and Fergal Somers. Asynchronous Transfer Mode (ATM) Source Traffic Prediction using Neural Netwoks. In *Irish Neural Networks Conference (INNC)*, Sep 1995.

[11] E. Nordström, O. Gällmo, L. Asplund, M. Gustafsson, and B. Eriksson. Neural Networks for Admission Control in an ATM Network. In L.F. Niklasson and M.B. Bodén, editors, *Connectionism in a Broad Perspective: Selected Papers from the Swedish Conference on Connectionism - 1992*, pages 239–250. Ellis Horwood, 1994.

[12] Mauro Oliveira, Miguel Franklin, Adriano Nascimento, and Marcelo Vasconcelos. *Introdução à Gerência de Redes ATM.* Editora CEFET-CE, 2nd. edition, 1998.

[13] R. O. Onvural. *Asynchronous Transfer Mode Network - Performance Issues.* Artech House Publishers, 2nd. edition, 1995.

[14] J. M. Pitts and J. A. Schormans. *Introduction to ATM Design and Performance.* John Wiley & Sons, 1996.

[15] University of Stuttgart - Institute for Parallel and Distributed High Performance Systems (IPVR). *The Stuttgart Neural Network Simulator - Version 4.0*, 1995.

[16] Tao Yang and Danny H. K. Tsang. A Novel Approach to Estimating the Cell Loss Probability in an ATM Multiplexer Loaded with Homogeneous On-Off Sources. *IEEE Transactions on Communications*, 43(1):117–126, Jan 1995.

[17] Tao Yang and Jun Yei. Optimal Solutions for a Dynamic Bandwidth Allocation Scheme in High-speed Networks. *Telecommunication Systems*, 5:389–412, 1996.