

Construção de um Agente SNMP para Gerência de Redes ATM utilizando Redes Neurais

Juliana Amaral Arantes
juliana@lrg.ufsc.br

Elvis Melo Vieira
elvis@npd.ufsc.br

Carlos Becker Westphall
westphal@orion.lrg.ufsc.br



Universidade Federal de Santa Catarina - UFSC
Departamento de Informática e de Estatística - INE
Laboratório de Redes e Gerência - LRG
Caixa Postal: 476 CEP: 88040-970 Florianópolis - SC
Fone: (048) 231-9498 Ramal: 235 Fax: (048) 231-9770

Resumo

As redes ATM (Modo de Transferência Assíncrono) possuem a capacidade de tratar vários tipos de tráfego e, conseqüentemente, oferecer uma grande variedade de serviços como uma de suas principais virtudes. No entanto, estes aspectos podem causar problemas na gerência destas redes, pois o controle destes tráfegos distintos, com características tão particulares, se torna uma tarefa complexa. Sendo assim, o controle de tráfego é um dos aspectos de gerência importantes em um ambiente ATM, e o Controle de Admissão de Conexão (CAC) é uma das principais funções de controle preventivo de tráfego. As redes neurais apresentam como características principais a sua capacidade de aprendizado e adaptação, sendo portanto, capazes de aprender com suas próprias experiências. Desta forma, as redes neurais podem ser utilizadas para prover características de adaptabilidade a agentes, tornando a gerência de redes mais automatizada. Neste contexto, este trabalho propõe a construção de uma rede neural para controle de admissão de conexões ATM que será incorporada a um agente SNMP.

Abstract

The ATM (Asynchronous Transfer Mode) networks are able to handle different types of traffic and consequently one of their main advantages consists on their capacity to support a huge variety of services. Nevertheless, these aspects may cause troubles in their management because the control of different traffic, with so many particular characteristics, became a difficult task. Thus, traffic control is one of the most important management aspects in an ATM environment, and the Connection Admission Control (CAC) is one the major function of preventive traffic control. On the other hand, one of the main characteristics of neural networks lies on their capability to adapt themselves to different situations, and learn with their own experiences. Moreover, the neural network can be used to provide adaptability to an agent, making network management a more automatic application. Within this context, this work proposes the construction of a neural network for the ATM Connection Admission Control, whose mechanisms will be incorporated to an SNMP agent.

Palavras-chave: Gerência de redes ATM, Controle de Admissão de Conexão, Redes Neurais, Agentes SNMP

1 Introdução

Com o crescimento das redes de computadores e da variedade de serviços oferecidos por estas redes, está se tornando cada vez mais essencial um gerenciamento eficiente. O gerenciamento eficiente de uma rede é uma tarefa complexa, e se torna ainda mais complexa quando o ambiente a ser gerenciado corresponde a uma rede de alta

velocidade, mais especificamente arquiteturas ATM. Desta forma, as redes ATM estão atraindo cada vez mais estudos e interesse devido a sua capacidade de transportar vários tipos de tráfego e, conseqüentemente, oferecer uma quantidade muito grande de serviços.

Um dos aspectos de gerência comumente observado em um ambiente ATM consiste de controle de tráfego na rede, e para este controle podem ser tomadas medidas preventivas e reativas. O ideal seria se o controle fosse inteiramente preventivo para evitar as situações de congestionamento e a perda de células. No entanto, em situações reais esta abordagem não é possível de ser implementada, e os dois tipos de controle podem ser utilizados conjuntamente para garantir a qualidade das conexões da rede. Um exemplo dos controles preventivos utilizados é o CAC, Controle de Admissão de Conexão, que é realizado no momento da requisição de uma conexão para decidir sobre sua aceitação ou rejeição com intuito de evitar uma situação de congestionamento e ao mesmo tempo otimizar a utilização da rede.

O advento de agentes tem originado novas perspectivas para o desenvolvimento de soluções em várias áreas, entre as quais a gerência de redes [12]. O fato de se implementar agentes através de redes neurais, que possuem como principal característica a sua capacidade de aprendizado, faz com que os agentes se tornem adaptáveis às mudanças no seu ambiente, sendo chamados de agentes adaptativos ou inteligentes.

Os agentes implementados utilizando redes neurais podem reagir aos problemas encontrados na rede, e de acordo com as suas experiências anteriores encontrar a ação mais apropriada para sanar a falha existente. Desta forma, a gerência de rede se torna mais automatizada, em contraste com a gerência tradicional onde qualquer decisão para solucionar um problema da rede é mantida, em muitos casos, sob responsabilidade do operador humano.

2 Gerência de Redes SNMP para ATM

A gerência de redes engloba métodos para planejar, configurar, controlar, monitorar, corrigir falhas e administrar redes de computadores. Para uma gerência ser considerada eficaz é indispensável que as falhas sejam solucionadas o mais rápido possível, preferencialmente sem o usuário observar alguma anormalidade.

2.1 Modelo de Gerência Internet

Um modelo de gerência visa apresentar uma estrutura através da qual a atividade de gerência possa ser realizada.

O SNMP (*Simple Network Management Protocol*), ou protocolo do modelo Internet, é um protocolo simples que permite uma NMS (*Network Management Station*) centralizada que consulta agentes para obter e modificar informações nas MIBs (*Management Information Base*)[3].

O protocolo SNMP consiste de 3 tipos de operações:

- GET: a NMS recupera uma informação específica do agente;
- SET: a NMS altera uma informação específica no agente;
- TRAP: um agente reporta um evento para a NMS.

O protocolo SNMP inclui a definição de uma MIB e de um mecanismo de acesso a MIB. O paradigma SNMP, apresentado em [3], é mostrado na Figura 1.

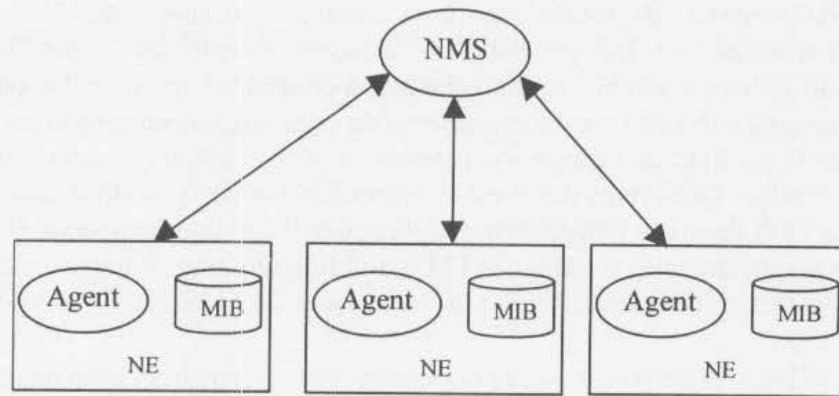


Figura 1 - Paradigma SNMP

Uma estação central de gerenciamento de rede (NMS – *Network Management Station*) troca mensagens com agentes residentes nos elementos de rede que estão sendo gerenciados. O agente em cada NE (Network Element) mantém uma MIB, que contém a descrição dos objetos gerenciados.

2.2 Ambiente de Gerência ATM

A Figura 2 representa um sistema genérico de gerência ATM. A IME (*Interface Management Entity*) é definida pelo *ATMForum* na *ILMI* (*Integrated Local Management Interface*). Cada estação possui uma IME que se comunica com uma IME do *switch* ATM através de uma interface única UNI (*User Network Interface*) e utilizam o protocolo SNMP para trocar informações de gerência. Uma IME de um *switch* ATM pode também estabelecer conexão com a IME de outro *switch* ATM.

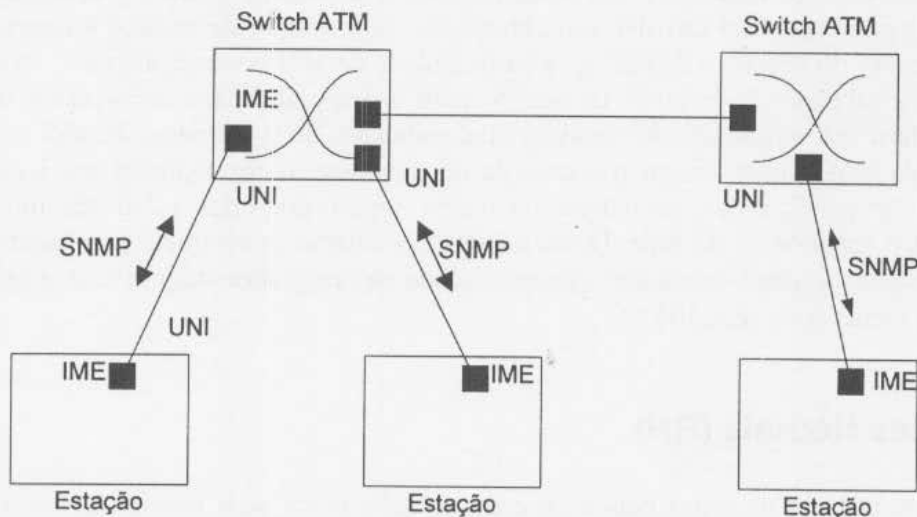


Figura 2 - Ambiente Genérico de Gerência ATM

3 Controle de Tráfego em Redes ATM

O Modo de Transferência Assíncrono (ATM - *Asynchronous Transfer Mode*) nasceu dos esforços para a padronização das redes B-ISDN (*Broadband Integrated Services Digital Network*) no final dos anos 80, sendo uma tecnologia capaz de suportar diferentes serviços com o intuito de satisfazer aos requisitos exigidos pelos diferentes tipos de tráfegos, a altas velocidades de transmissão.

Parte do processo de estabelecer uma conexão em redes ATM consiste da negociação do contrato de tráfego entre as estações envolvidas, especificando as características do tráfego a serem mantidas durante a conexão. Uma lista dos parâmetros de tráfego, requisitados durante o estabelecimento da conexão, é armazenada no descritor de tráfego fonte. O contrato de tráfego compreende os parâmetros no descritor de tráfego, a qualidade de serviço (QoS) requisitada e a tolerância da variação do atraso das células.

Suprir a QoS desejada por diferentes aplicações é bastante complexo. Para tornar mais fácil esse gerenciamento, o tráfego ATM é dividido em cinco categorias de serviço que relatam as características do tráfego e as requisições de QoS para o comportamento da rede:

- **CBR (Constant Bit Rate):** usada por conexões que requerem uma quantidade de largura de banda estática que está disponível continuamente durante todo o tempo de vida da conexão.
- **VBR (Variable Bit Rate):** usado por conexões com taxa de transmissão variável. É subclassificado em VBR tempo real (rt-VBR) e VBR tempo não real (nrt-VBR), dependendo da sensibilidade à variação de atraso de célula.
- **ABR (Available Bit Rate):** serviço sem garantia de atraso ou taxa de transmissão, ocorrendo ajuste da taxa de transmissão em caso de congestionamento.
- **UBR (Unspecified Bit Rate):** serviço para utilizar a banda restante, onde as células que são perdidas durante o congestionamento são retransmitidas.

O controle de tráfego é um dos aspectos de gerência importantes em uma rede ATM, e seus objetivos são proteger a rede contra congestionamentos, garantir o cumprimento das metas de desempenho da rede e otimizar os recursos da rede. A principal característica desse controle é o seu caráter preventivo.

3.1 Controle de Admissão da Conexão (CAC)

O Controle de Admissão de Conexão é uma técnica de controle de tráfego, sendo portanto de caráter preventivo. O CAC é um controle de acesso que é realizado pela rede no momento do estabelecimento da conexão, com o intuito de decidir pela aceitação ou rejeição da conexão. Isso envolve uma checagem do contrato de tráfego requerido pelo usuário através do descritor de tráfego e da qualidade de serviço desejada para verificar se é possível estabelecer a conexão de acordo com a disponibilidade de recursos da rede, sem interferir na qualidade de serviços das conexões já existentes. Sendo assim, os objetivos da CAC são restringir o acesso de novas chamadas de maneira que a classe de qualidade de serviço das conexões existentes sejam mantidas e ao mesmo tempo maximizar o rendimento da rede. Desta maneira, o critério de bloqueio (não admissão da conexão) utilizado deve balancear a probabilidade de congestionamento com a utilização eficaz dos recursos da rede [9].

4 Redes Neurais (RN)

Um modelo de redes neurais é caracterizado pelos seus neurônios, as conexões entre eles e pelo seu esquema de aprendizado. As redes neurais artificiais possuem inspiração biológica, apresentando portanto algumas características semelhantes ao cérebro humano como:

- **Aprendizagem:** As RNs podem modificar seu comportamento em resposta a eventos ou fatos que ocorreram no meio externo e que fornecem um conjunto de entradas, ao qual pode ser associado um conjunto de saídas desejadas. Através de um algoritmo de treinamento, este conjunto provoca um auto-ajuste da rede para produzir um conjunto de respostas adequado e consistente com os padrões de entrada. [2]

- Generalização: Após o aprendizado de um conceito, a rede é capaz de funcionar com conceitos similares, que ainda não foram aprendidos.

A topologia, ou seja, a maneira como os neurônios estão conectados uns aos outros em uma RN causa um enorme efeito na sua operação, determinando o tipo de processamento que irá ocorrer.

O comportamento da rede neural é determinado pelos pesos das suas conexões que são estabelecidas no seu processo de treinamento. Uma rede está treinada quando aplicando-se um conjunto de entradas na rede se obtém um conjunto de saídas desejadas. Quando se atinge esse estágio, os valores dos pesos da rede estão adequados para o contexto do problema em que a rede foi treinada. Então, treinar uma rede é uma questão de ajuste de pesos. Assim, a fase de treinamento consiste de se aplicar um conjunto de entradas seqüencialmente à rede que através de um algoritmo de treinamento (aprendizado) ajusta o peso das conexões para se chegar à saída desejada. O treinamento é a única fase de uma rede neural onde se tem aprendizado. Durante o treinamento da rede é importante monitorar seu desempenho para evitar que um treinamento muito prolongado leve a rede a um estado de *overlearning*. Nesse estado a rede perde sua capacidade de generalização, tentando memorizar individualmente os pares de entrada e saída utilizados no treinamento.

5 Redes Neurais x Gerência de Redes

A tendência atual é se ter uma gerência de redes cada vez mais automatizada e as redes neurais, devido às suas características de aprendizagem e generalização, podem ser extremamente úteis na realização de várias tarefas que dependem do operador humano. Sendo assim, serão apresentadas a seguir, algumas das possíveis utilizações satisfatórias de RNs na gerência de redes de computadores.

5.1 Redes Neurais na Construção de Agentes

Genericamente, um agente é um hardware ou software que apresenta propriedades como autonomia (controle de suas ações e estados internos), interação entre si ou com as pessoas e reagem às mudanças que ocorrem no seu ambiente de forma adequada.

As redes neurais, como já foi dito, têm como uma de suas principais características a capacidade de aprendizado, ou seja, de se adaptar às mudanças de contexto. Sendo assim, estas redes podem ser utilizadas para construir agentes capazes de se adaptarem aos diferentes eventos que ocorrem em uma rede de computadores. Esses agentes são chamados de adaptativos porque além de atuarem como agentes de regulação e de planejamento, possuem capacidade de aprendizagem, sendo considerados então agentes inteligentes, em virtude de possuírem a capacidade de aprender com as próprias experiências [12].

5.2 Redes Neurais na Gerência de Redes ATM

Atualmente estão sendo desenvolvidas várias aplicações em redes de alta velocidade utilizando redes neurais e, apesar desta área ser bastante recente, já foram obtidos progressos significativos. Análises de performance mostram que as redes neurais normalmente alcançam resultados melhores que os algoritmos convencionais [4][7][10]. Com uma arquitetura apropriada de redes neurais é possível prever facilmente o comportamento de sistemas não lineares variáveis com o tempo, que é o caso das redes ATM.

Em [10] os autores informam que o ponto que mais favorece as redes neurais sobre os algoritmos convencionais na resolução de problemas das redes ATM é sua

capacidade de aprendizado e adaptação. Essa capacidade pode ser utilizada para construir algoritmos adaptativos (e computacionalmente inteligentes) para a alocação de recursos (por exemplo largura de banda, *buffers*, etc.), o que favorece o desenvolvimento de ferramentas altamente efetivas para o controle de congestionamento.

Já em [7], os autores citam que as RNs possuem diversas propriedades que motivam seu uso na implementação de controle de tráfego ATM. A primeira dessas propriedades é a adaptabilidade que faz com que não seja necessário nenhum modelo explícito do tráfego como nos métodos tradicionais, sendo necessária somente uma boa representação do problema. A segunda destas propriedades é que a estrutura paralela das RNs pode ser explorada em implementações de hardware, que provêm tempo de resposta curto, possibilitando ações de controle rápidas.

6 Construção do Agente SNMP

Anteriormente, foi apresentada a importância do controle de tráfego para a gerência de redes ATM, bem como as vantagens da utilização de RNs na gerência destas redes e na construção de agentes. Aqui será descrita a implementação de um agente que contém funções de controle de admissão de conexões ATM fazendo uso de redes neurais. O objetivo desta implementação é o estudo do emprego de redes neurais para o controle de admissão de conexão no lugar do método estatístico tradicional, que além de lento pode se tornar muito complexo [4][7][10].

A Figura 3 mostra o agente SNMP implementado.

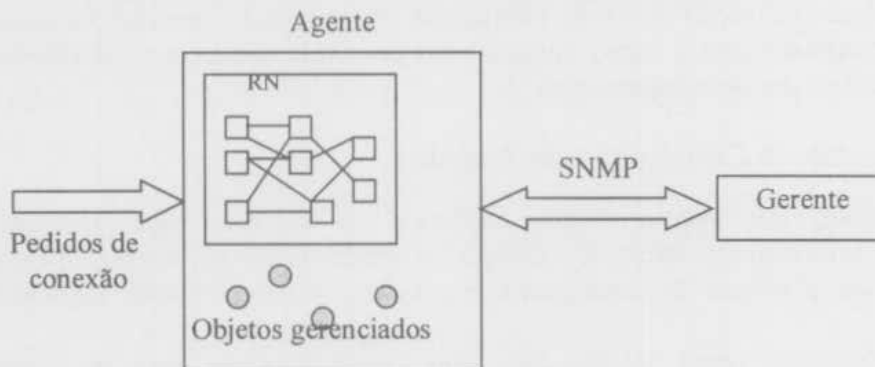


Figura 3 - Agente baseado em redes neurais

A construção deste agente passa pelas seguintes etapas:

- Construção e treinamento da rede neural para o controle de admissão de conexão;
- Implementação do agente, utilizando a biblioteca SNMPLib e construção de módulos de interface entre rede e *switch*, que são dependentes do fabricante.

6.1 Construção da Rede Neural para Controle de Admissão de Conexão

Como já foi observado, o CAC decide sobre a aceitação ou não de uma conexão baseando-se nos parâmetros de tráfego requisitados pelo usuário e nas condições da rede no momento, isto é, se a rede tem capacidade para cumprir o contrato de tráfego a ser assumido com o usuário sem degradar os serviços que já estão sendo utilizados. Como existem vários tipos de tráfego ATM, com características e requisições bem distintas, não é possível estipular um conjunto de parâmetros comuns que seriam importantes na decisão da aceitação de conexões de cada um destes tipos de tráfego. Sendo assim, é necessário a construção de uma rede neural para o CAC de cada um destes tipos de tráfego (CBR, VBR, ABR, UBR). Inicialmente foi construída somente uma rede neural

para conexões de tráfego VBR, até mesmo para provar a viabilidade da utilização de RNs para controle de admissão. Para tanto foi usado o SNNS, descrito a seguir.

6.1.1 O Simulador SNNS (*Stuttgart Neural Network Simulator*)

Para a construção e treinamento da rede neural foi utilizado o SNNS (*Stuttgart Neural Network Simulator*), um simulador de redes neurais desenvolvido pelo Instituto de Sistemas Paralelos e Distribuídos de Alta Performance da Universidade de Stuttgart na Alemanha. Segundo [8], o objetivo do projeto do SNNS é criar um ambiente de simulação eficiente e flexível para pesquisa e aplicação de redes neurais.

A escolha do SNNS se deve ao fato de ser um dos simuladores de redes neurais *freeware* mais completos disponíveis e, principalmente, por este gerar o código em linguagem C da RN, o que facilita consideravelmente a implementação do agente.

6.1.2 Topologia da RN construída

A rede apresenta a arquitetura *feedforward*, que é uma rede sem ciclos e que depois de treinada apresenta sempre a mesma resposta para a mesma entrada.

A rede neural que foi construída para a resolução do problema é formada por 5 neurônios de entrada, uma camada intermediária com 20 neurônios e um neurônio de saída.

6.1.2.1 Camada de Entrada

As entradas que são aplicadas aos neurônios da camada de entrada são os parâmetros requisitados pelo usuário para a conexão a ser estabelecida e os dados da rede para verificar a possibilidade de estabelecimento da conexão, atendendo ao que foi requisitado e mantendo a qualidade das conexões já existentes. É importante salientar que a conexão requisitada tem que ser para tráfego VBR, já que foram determinados os parâmetros a seguir para serem aplicados à entrada da rede justamente por serem os mais importantes para a caracterização de um tráfego VBR.

Os parâmetros requisitados pelo usuário são:

- Taxa média de transmissão de células;
- *Burst* (tolerância);

De acordo com estes dois parâmetros, é calculada a largura de banda requisitada que será uma das entradas da RN. As demais entradas são os seguintes parâmetros da rede:

- Número máximo de VCCs (*Virtual Channel Connection*);
- Número de VCCs ativos;
- Largura de banda disponível;
- Percentagem de ocupação do *buffer*.

Todos esses dados se referem a apenas uma porta do *switch*. Sendo assim, a rede neural decide se a conexão pode ou não ser estabelecida naquela porta do *switch* associada aos parâmetros.

6.1.2.2 Camada Intermediária

Como a literatura afirma que não existe nenhuma fórmula para se calcular precisamente quantos neurônios deve ter a camada intermediária, foram feitos testes com alguns valores e, dentre os valores testados, a rede aprendeu mais rapidamente com 20 neurônios em uma única camada.

6.1.2.3 Camada de Saída

A saída da rede é um valor binário, 0 se a conexão foi rejeitada ou 1 se a conexão foi aceita. Foi considerado aceitável na saída da rede um erro menor ou igual a 0.09, já que com esse valor de erro é possível identificar claramente as duas possíveis saídas reais da rede (0 e 1). Os valores entre 0.3 e 0.6 foram considerados indefinidos.

6.1.3 Treinamento da RN

É nesta fase de treinamento que a rede adquire o seu conhecimento, isto é, a fase onde acontece o aprendizado.

Para o treinamento da rede neural foi utilizado o aprendizado supervisionado, ou seja, para cada entrada fornecida à rede no seu período de treinamento foi apresentada a saída correspondente. O algoritmo de treinamento empregado foi o *Backpropagation Momentum*, que é basicamente uma versão do *Backpropagation*, o algoritmo mais conhecido para o treinamento de uma RN. Este algoritmo visa a eliminação da "parte horizontal" do gráfico do erro, ou seja, do período que não há variação expressiva no valor do erro. Essa eliminação é realizada através da mudança de peso antiga como um parâmetro para a computação da nova mudança de peso. Tal procedimento evita o problema da oscilação, comum no algoritmo *Backpropagation* tradicional quando a superfície do erro varia em uma área bastante estreita. A nova mudança de pesos é dada por:

$$\Delta w_{ij}(t+1) = \eta * \delta_j * o_j + \alpha \Delta w_{ij}(t)$$

onde: w_{ij} = peso da conexão do neurônio i com o neurônio j

η = fator de aprendizagem (constante)

δ_j = erro (diferença entre a saída real e a saída desejada) no neurônio j

o_j = saída do neurônio j

α = taxa de aprendizagem

t = ciclo ou época

O efeito dessa intensificação é que a parte horizontal da superfície do erro é atravessada relativamente rápida com poucos passos grandes, enquanto o tamanho do passo é decrementado a medida que a superfície vai se tornando mais aproximada. Essa adaptação do tamanho do passo aumenta a velocidade de aprendizado significativamente. O algoritmo *Backpropagation Momentum* já está implementado pelo SNNS.

6.1.3.1 Arquivos padrão de treinamento e validação

Para o treinamento da rede no SNNS, as entradas e suas respectivas saídas que são apresentadas à rede estão na forma de um arquivo padrão de treinamento. Para evitar o *overlearning* foi apresentado à rede, além do padrão para treinamento, um padrão para validação da rede. A Figura 4 mostra o gráfico de treinamento da RN.

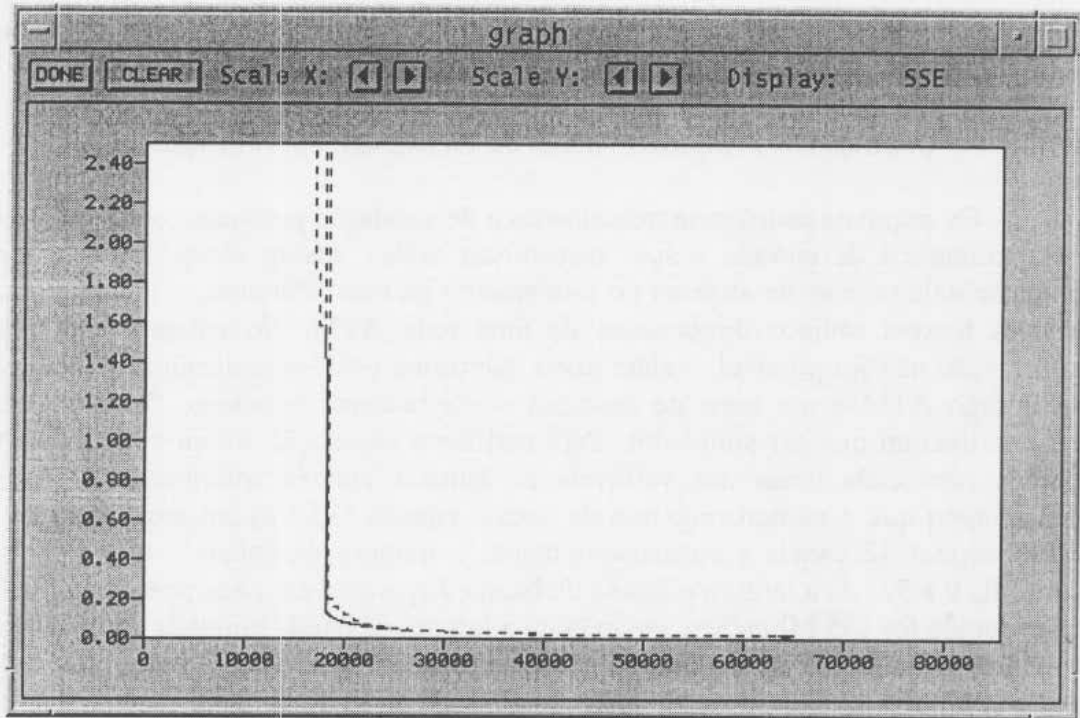


Figura 4 - Gráfico do treinamento da RN

O eixo horizontal representa a taxa de erro e o eixo vertical representa o número de ciclos ou épocas. Cada ciclo é uma unidade de treinamento onde todos os padrões de treinamento são apresentados a rede. A linha contínua representa a evolução do treinamento da rede e a linha pontilhada representa a evolução da validação.

O treinamento de uma rede neural pode ser uma tarefa extremamente demorada, necessitando de horas de processamento. Segundo [8] é aconselhável que esta tarefa seja realizada por um processo executado em *background*. Assim foi utilizada uma ferramenta do SNNS chamada *batchman* para a construção de uma rotina para o treinamento da rede para ser executada em *background*. A rotina pode ser vista na Figura 5.

```

loadNet("vbr.net")
loadPattern("vbr.pat")
setInitFunc("Randomize_Weights", 0.2, 0.1)
setLearnFunc("BackpropMomentum")
setUpdateFunc("Topological_Order")
mode:=TRUE
setShuffle(mode)
initNet()
while SSE > 0.099 do
  if CYCLES mod 100 == 0 then
    print("ciclos = ", CYCLES, "   SSE = ", SSE)
  endif
  trainNet()
endwhile
saveResult("vbr.res", 1, PAT, TRUE, TRUE, "create")
saveNet("vbr.trained.net")
print("Cilcos treinados: ", CYCLES)
print("Treinamento interrompido no erro: ", SSE)

```

Figura 5 - Rotina para treinamento da RN

Após a rede ter sido treinada e validada foi usada novamente uma ferramenta do SNNS (*snms2c*) para gerar o código C, que foi usado na implementação do agente. A partir deste ponto a rede não é mais treinável, isto é, não pode mais aprender. Por isso é

indispensável que, antes de se gerar o código C, a rede tenha atingido o nível de aprendizado desejado para a resolução do problema.

6.1.3.1.1 Definição dos Arquivos Padrão de Treinamento e Validação

Os arquivos padrões de treinamento e de validação possuem conjuntos de valores dos parâmetros de entrada e suas respectivas saídas. Sendo assim, esses valores são fundamentais para se ter sucesso no treinamento da rede. Portanto, o ideal seria se estes valores fossem obtidos diretamente de uma rede ATM. No entanto, esta coleta de informação não foi possível, e além disso, conforme estudos realizados, a caracterização de tráfego ATM é um tema de pesquisa ainda bastante nebuloso. Desta forma, estes valores tiveram que ser simulados. Para realizar a simulação foram estipulados valores limites para cada uma das variáveis de entrada citadas anteriormente. Assim, foi considerado que o número máximo de canais virtuais (VCCs) em uma porta do switch ATM seriam 32 canais e conseqüentemente o número de canais virtuais ativos pode variar de 0 a 32. Já a largura máxima de banda disponível em cada porta do switch ATM considerada foi 155 Mbits/Seg, então tanto a largura de banda utilizada como a largura de banda requisitada variam de 0 a 155 Mbits/Seg.. Além disso, como neste caso cada porta do *switch* pode possuir apenas conexões para tráfego VBR, foi considerado que uma das portas tem um *buffer* associado. Assim, quando não existe largura de banda disponível compatível com a requisição do usuário, são considerados a percentagem da largura de banda que falta para satisfazer a requisição e a percentagem do *buffer* ocupada, de acordo com a Figura 6.

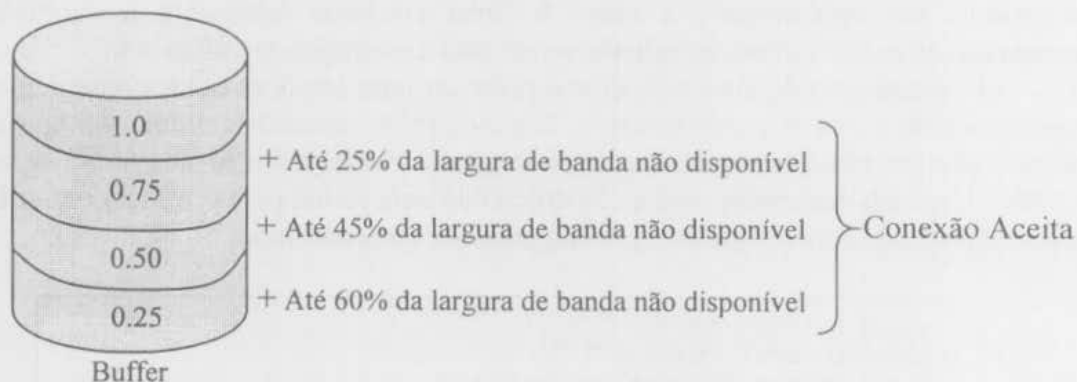


Figura 6 - Relação entre taxa de ocupação do buffer e largura de banda para tráfego VBR

É importante salientar que os valores acima foram pré-estabelecidos.

Os arquivos de treinamento e validação são formados então por várias combinações possíveis dos valores de entrada e a saída desejada para cada uma destas combinações. Uma amostra do arquivo de treinamento pode ser observado na Figura 7.

```
SNNS pattern definition file V3.2
generated at Wed Oct 22 17:30:00 1997

No. of patterns : 702
No. of input units : 5
No. of output units : 1

#VCCAt VCCMax BandaReq BandaDisp Ocupacao Saida

#pattern 1
0.00000 1 0.16129 0.64516 0.25 1
#pattern 2
0.00000 1 0.16129 0.64516 0.50 1
(...)
```

Figura 7 - Amostra do arquivo padrão de treinamento

Na figura acima, é possível observar que o arquivo de treinamento é formado por 702 padrões, ou seja, 702 combinações de valores de entrada e suas saídas desejadas. Como é possível observar, são apresentados os valores das variáveis de entrada na seguinte ordem: número de VCCs ativos (VCCAt), número máximo de VCCs (VCCMax), largura de banda requisitada para a conexão (BandaReq), largura de banda disponível na porta do *switch* (BandaDisp) e a taxa de ocupação do *buffer* (Ocupação). E, em seguida, é apresentada a saída compatível com as entradas, 0 se a conexão é rejeitada e 1 se a conexão é aceita. Os valores aparecem nesta forma porque foram normalizados para estarem sempre entre 0 e 1. Assim os valores relativos à VCCs foram divididos por 32, os valores relativos à largura de banda foram divididos por 155 e a taxa de ocupação do *buffer* foi dividida por 100. Teoricamente isto não é necessário, mais na prática facilita e acelera o processo de aprendizado. Aqui fica inviável mostrar todo o arquivo de treinamento devido a sua extensão, já que são 702 padrões.

6.2 Implementação do agente

A implementação do agente para controle de admissão de conexão foi realizada utilizando a biblioteca SNMPLib, desenvolvida em [5]. Esta biblioteca é um conjunto de classes para construção de aplicações de gerência de redes de computadores que tem como principal objetivo fornecer o suporte necessário a construção de agentes SNMP. Uma definição completa desta biblioteca, bem como dos métodos e classes utilizados, pode ser obtida em [5].

O agente construído recebe mensagens SNMP e realiza as operações pedidas por esta mensagem (*get*, *set*, *get-next*) sobre os objetos gerenciáveis. Além disso, a cada sinalização de pedido de conexão, o agente tem que receber os parâmetros do usuário e passá-los pela rede neural para estabelecer ou não a conexão.

Com o intuito de simular a sinalização de pedidos de conexão, foi utilizado um arquivo onde se encontram as requisições de conexão de usuários com seus respectivos parâmetros, ordenados por tempo. Sendo assim, de tempo em tempo, o agente lê o arquivo e consulta a rede neural para verificar a possibilidade de estabelecimento da conexão requisitada. Este processo de consultar a RN se resume em chamar a rotina C++ que foi gerada pelo SNNS a partir da rede treinada, como foi descrito anteriormente.

As operações requisitadas pelo gerente através de mensagens SNMP são realizadas com a ajuda de outro arquivo, que simula a interface da rede com o *switch*, contendo o valor atual de todos os objetos gerenciados. Esta interface é dependente do fabricante do *switch*.

O problema foi modelado de acordo com a OMT (*Object Modeling Technique*), que é uma metodologia de análise e projeto baseada em objetos, e foi implementado em C++.

A análise e implementação do agente ficam visivelmente facilitadas devido, principalmente, ao encapsulamento provido pelas classes contidas na SNMPLib. Isso pode ser observado na construção dos diagramas na fase de análise do problema, que expressam muito pouco sobre os conceitos de gerência de rede e até mesmo os conceitos do protocolo SNMP.

7 Conclusões

A construção da rede neural para o CAC realmente se mostrou eficiente, e assim também se comprovou os pontos favoráveis que as redes neurais apresentam para realizar o controle de tráfego ATM. Essas vantagens se devem à capacidade de adaptação e aprendizagem das redes neurais, e devido a isso, só se faz necessária uma boa representação do problema para a sua resolução satisfatória. Neste caso, a representação do problema é feita através do padrão de treinamento submetido à rede. Portanto, a veracidade do resultado do treinamento da rede neural é fortemente dependente do padrão aplicado à rede durante seu período de aprendizagem. Sendo assim, o ideal seria a obtenção de valores reais para serem aplicados como padrões de treinamento e validação/teste. Mas como não foi possível realizar uma operação de comunicação real com a rede ATM para a obtenção de valores, estes tiveram que ser gerados por simulação, considerando os limites de cada um dos atributos que foram utilizados como entrada da RN. Apesar deste inconveniente, foram atingidos os resultados esperados de rejeição ou aceitação de conexão nos testes realizados.

Neste caso, como já foi citado, a rede neural foi treinada considerando um *switch* onde cada porta suporta até 32 VCCs e tem largura de banda de 155 Mbts/Seg. Caso o controle de admissão de conexão for realizado por um *switch* com diferentes configurações é interessante que a rede neural seja treinada novamente com os novos valores limites para a obtenção dos resultados esperados.

Outro ponto importante é que se pretendia construir uma rede neural para controle de conexão para todos os tipos de tráfego. Isto não foi possível devido às características bastante específicas de cada um destes tipos. Chegou-se então à conclusão que seria necessária uma rede neural para cada tipo de tráfego ATM. Neste trabalho foi implementada apenas a rede neural para controle de admissão de conexão de tráfego VBR.

Uma proposta de trabalho futuro seria a construção de uma rede neural para cada tipo de tráfego ATM, ou seja para tráfego ABR e UBR, já que para VBR foi desenvolvida neste trabalho. Para conexões de tráfego CBR não é necessária uma rede neural para tomar a decisão de aceitação devido à pouca complexidade na decisão, em virtude da característica de taxa de transmissão constante. Outra possibilidade de continuação seria o estudo da possível utilização de redes neurais para resolver outros problemas em redes ATM.

8 Bibliografia

- [1] ARANTES, J.A.A. **Construção de um Agente SNMP para Gerência de Redes ATM utilizando Redes Neurais**. Trabalho de Conclusão do Curso de Ciências da Computação – UFSC. Dezembro de 1997
- [2] BARRETO, J. M. **Introdução às Redes Neurais Artificiais**, Anais da V Escola

- Regional de Informática da SBC Região Sul. Florianópolis (SC), maio de 1997.
- [3] CHEN, M. T.; LIU, S.S. **ATM Switching Systems**. Editora Artech House, 1995.
- [4] IEEE NETWORK - The Magazine of Computer Communications. ATM Flow Control: Rate vs. Credit. Março/Abril 1995. Volume 9, número 2.
- [5] FIORESE, A.; SILVA, A. SNMPLib, **Biblioteca de Classes de Gerência de Redes Internet**. Trabalho de Conclusão do Curso de Ciências da Computação – UFSC. Dezembro de 1997
- [6] NEVES, J.E.; LEITÃO, M.J.; ALMEIDA, L.B. **Neural Network in B-isdn Flow Control: ATM Traffic Prediction or Network Modelling?** IEEE COMMUNICATIONS MAGAZINE. Neurocomputing in High-Speed Networks. Outubro 1995. Volume 33, número 10, pág 50-56.
- [7] NORDSTROM, E.; Carlstrom, J; Gallmo, O.; Asplund, L. **Neural Network for Adaptative Traffic Control in ATM Networks**. IEEE COMMUNICATIONS MAGAZINE. Neurocomputing in High-Speed Networks. Outubro 1995. Volume 33, número 10, pág 43-49.
- [8] Stuttgart Neural Network Simulator. **User Manual**, Version 4.1. Stuttgart – Germany, 1995.
- [9] SOARES, L. F. G., Lemos, G. e Colcher, S., **Redes de Computadores - Das LANs, MANs e WANs às Redes ATM**, Rio de Janeiro: Editora Campus, 1995.
- [10] TARRAF, A.A.; Habib, I.W.; Saadawi, T.N. **Intelligent Traffic Control for ATM Broadband Networks**. IEEE COMMUNICATIONS MAGAZINE. Neurocomputing in High-Speed Networks. Outubro 1995. Volume 33, número 10, pág 76-82.
- [11] VIEIRA, E. M. **Métodos para Desenvolver Agentes Adaptativos em Gerência e Redes usando Redes Neurais**. Dissertação de Mestrado CPGCC UFSC, março de 1997.
- [12] VIEIRA, E. M.; Westphall, C.B.; Sari, S. T. **Desenvolvimento de Sistemas Adaptativos para Gerência de Redes de Computadores usando Redes Neurais**. Anais do Simpósio de Redes Neurais. Florianópolis, julho de 1997.
- [13] WANGENHEIM, A. **Curso Introdução de Redes Neuronais – Projeto Cyclops – INE/CTC/UFSC**