

# Desenvolvendo Agentes Inteligentes para a Gerência Pró-Ativa de Redes ATM

Adriano Nascimento Miguel Franklin Mauro Oliveira

blau@lia.ufc.br

miguel@lia.ufc.br

mauro@etfce.br

LAR - Laboratório Multiinstitucional de Redes e Sistemas Distribuídos

CEFET/CE - Centro Federal de Educação Tecnológica (Ceará)

LIA - Laboratório de Inteligência Artificial

UFC - Universidade Federal do Ceará

## Resumo

Este trabalho descreve as arquiteturas física e funcional do protótipo RENATA (**RE**des **NE**urais **A**plicadas ao **T**ráfego **ATM**). Este ambiente permite o desenvolvimento de agentes inteligentes, baseados em redes neurais, destinados à gerência pró-ativa de redes ATM. Devido às suas características de aprendizado, generalização e adaptabilidade, as redes neurais possibilitam detecção antecipada de situações anormais em uma rede ATM através da análise de seus estados. O agente desenvolvido na RENATA fornece informações para a rede neural, age segundo sua saída e monitora seu comportamento.

## Abstract

*This work describes RENATA (Neural Networks Applied to the ATM Traffic Management), a tool that enables the development of intelligent agents to perform proactive management at an ATM network. The agents are based on Artificial Neural Networks, as their intelligent component. Due to their learning, generalization and adaptability capabilities, neural networks make possible anticipated detection of abnormal situations on an ATM network, by analyzing its status information. The developed agent provides information to the neural network, acts according to its outputs and monitors its behavior.*

## 1 Introdução

As aplicações em redes de computadores têm exigido maior largura de banda e requisitos mais acurados, tais como mínimo atraso e baixa taxa de perda nos meios utilizados. As tecnologias convencionais têm se mostrado incapazes de garantir Qualidade de Serviço (QoS) ao integrar em uma mesma infra-estrutura serviços de voz, vídeo e dados. O *Asynchronous Transfer Mode* (ATM) preenche satisfatoriamente estes requisitos, tendo sido escolhido pelo ITU-T (*International Telecommunications Union - Telecommunications*) como o suporte das Redes Digitais de Serviços Integrados de Faixa Larga (RDSI-FL).

Devido à complexidade e flexibilidade inerentes ao ATM, um sistema de gerência de redes eficiente torna-se indispensável. Soluções de gerência que eram adequadas para outras tecnologias são insuficientes para o ATM. A diversidade de serviços oferecidos, o ambiente de alta velocidade e a necessidade de uma solução integrada exigem novos requisitos por parte do sistema de gerenciamento.

As ferramentas de gerência devem antecipar possíveis problemas antes que eles aconteçam e não apenas atuar de maneira reativa. É fundamental observar um comportamento anormal da rede, coletar seus sintomas e diagnosticar corretamente um problema maior que possa vir a acontecer. A adoção de uma abordagem pró-ativa é portanto sempre desejável em um sistema de gerência, em especial em ambientes ATM que demandam novos requisitos. Técnicas como sistemas especialistas e redes neurais, entre outras, têm sido utilizadas para dotar os sistemas de gerência, do conhecimento que os tornem capazes de uma atitude pró-ativa.

Este trabalho apresenta RENATA (**RE**des **N**eurais **A**plicadas ao **T**ráfego **A**TM), uma nova proposta para o desenvolvimento de agentes inteligentes, baseados em redes neurais, destinados à gerência pró-ativa de redes ATM. Devido às suas características particulares, as redes neurais possibilitam a esses agentes a predição de situações anormais em uma rede ATM. Os agentes da RENATA poderão assim atuar diretamente ou apenas comunicar o problema ocorrido ao administrador da rede. Este poderá, então, tomar medidas preventivas que evitem maiores complicações no sistema gerenciado.

A arquitetura RENATA tem sido implementada contemplando desde a obtenção das informações necessárias ao treinamento da rede neural, passando pela sua criação até o desenvolvimento do agente. Este interage com os mecanismos de gerência ATM (MIBs SNMP aplicadas ao ATM, ILMI e Células OAM) e a com rede neural para possibilitar o desejado comportamento pró-ativo. Além da especificação das arquiteturas física e funcional, é descrito um protótipo que integra as ferramentas necessárias para a implementação dos agentes.

Este trabalho tem a seguinte organização. A seção 2 apresenta aspectos da gerência ATM. A seção 3 comenta as tecnologias envolvidas neste trabalho, discutindo a aplicação de redes neurais na gerência ATM, conceitos de agentes inteligentes e de gerência pró-ativa. A seção 4 descreve RENATA, explicando sua arquitetura funcional e física. A partir dessas definições, foi implementado um protótipo, especificado na seção 5. Nesta seção, também são comentados detalhes do funcionamento do agente. A seção 5 é concluída com a descrição de um estudo de caso realizado a partir de trabalhos relacionados. Por fim, a seção 6 apresenta conclusões e sugestões de trabalhos futuros.

## 2 Gerência de Redes ATM

A tecnologia ATM é caracterizada pela integração, desde seu surgimento, de técnicas e infra-estrutura destinadas ao monitoramento e controle de seus componentes. Quanto à infra-estrutura, trata-se do Plano de Gerência do Modelo de Arquitetura nos Protocolos RDSI-FL. Provavelmente, esta característica adicional na arquitetura ATM que privilegia a questão da gerência deve-se a uma maior complexidade desta tecnologia, necessária ao tratamento eficiente de vários tipos de mídia.

Devido às suas especificidades, gerência ATM exige uma organização diferente daquela utilizada na gerência de pacotes. Em [1], é apresentada uma proposta que melhor organiza esses conceitos:

1. Elementos de Interesse à Gerência (EIGs) semelhantes a Objetos Gerenciados SNMP (*Simple Network Management Protocol*)
2. EIG resultado da interação de dois blocos funcionais em uma rede ATM (ex.: comutador-comutador, comutador-estação)
3. EIG intrínseco à arquitetura de protocolos RDSI-FL, portanto sem similaridades com Objetos Gerenciados SNMP

A opção 1 pode ser tratada, com limitações, da mesma forma pelo modelo de gerência convencional. Afinal, existem EIGs numa rede ATM que podem ser modelados como Objetos Gerenciados em uma MIB (*Management Information Base*) e acessados pelo protocolo de gerenciamento, dentro do paradigma Gerente-Agente. Esta opção trata-se das *MIBs SNMP Aplicadas ao ATM*.

Várias organizações, como o ATM Forum e o IETF (*Internet Engineering Task Force*), têm proposto MIBs SNMP que modelam de aspectos gerais da tecnologia ATM a específicos de determinados meios de transmissão (ex: SONET MIB) ou de determinados ambientes (ex: LANE MIB). Dentre as mais importantes, destacam-se a AToM MIB e as extensões da RMON MIB para ATM.

Já a opção 2 refere-se a EIGs associados às diversas interfaces existentes, em particular à interface UNI (User Network Interface). Esta opção também se baseia no modelo Gerente-Agente, com um diferencial importante: uma série de procedimentos definidos e que funcionam independentemente dos mecanismos de gerência existentes na rede em questão. Esta opção denomina-se *ILMI*.

A ILMI (*Integrated Local Management Interface*) [2] foi definida pelo ATM Forum, para possibilitar a configuração de uma interface UNI e fornecer informações de status e controle. A UNI conecta um equipamento ATM, seja um comutador ou um terminal, a uma rede ATM pública ou privada. A ILMI também define um mecanismo de registro de endereços e serviços pela UNI, além dos procedimentos de conectividade ILMI. Cada UNI de dispositivo possui uma IME (*Interface Management Entity*) e uma instância da ILMI MIB.

A opção 3 é um mecanismo de gerência intrínseco à arquitetura ATM. Trata-se de um conjunto de funções de gerenciamento a nível de camada. Esta opção refere-se às *Funções OAM*. OAM (*Operation and Maintenance*) é um conjunto de funções recomendadas pela ITU-T no contexto das RDSIs. Esta infra-estrutura foi, posteriormente, ratificada pelo ATM Forum, em sua recomendação da interface UNI 3.1. Atualmente, cinco funções de gerenciamento OAM estão definidas: monitoramento de desempenho, detecção de falhas, proteção de sistemas, informações de falha e desempenho e localização de falhas.

RENATA faz uso destes mecanismos, permitindo que seus agentes utilizem estes padrões da gerência ATM, tanto na coleta de informações quanto nas suas ações. Os padrões facilitam uma configuração mais genérica dos agentes, aumentando sua aplicabilidade.

### 3 Tecnologias Envolvidas

Tanto a aplicação de redes neurais em gerência ATM, como o desenvolvimento de agentes inteligentes têm despertado atenção, motivando pesquisas. A grande motivação deste trabalho é facilitar a integração destas tecnologias, através da implementação de um

protótipo que permita a criação de agentes inteligentes baseados em redes neurais, visando a dotá-los de comportamento pró-ativo e de maior autonomia. Estes conceitos e tecnologias serão explicados a seguir.

### 3.1 Redes Neurais em Gerência ATM

As redes neurais são sistemas computacionais maciçamente paralelos, consistindo de um grande número de processadores simples com muitas interconexões. As redes neurais possibilitam um novo paradigma para a solução de problemas devido às suas características de aprendizado, adaptabilidade, robustez e suporte à falha [3].

Vários trabalhos aplicam redes neurais ao problema de Gerenciamento de Recursos (RM - *Resource Management*) de uma rede ATM. RM é realizado em dois níveis. No nível mais baixo, existe a necessidade de mecanismos de controle on-line que evitem congestionamentos na rede a curto e longo prazo, dada a topologia da rede virtual. Em um nível mais alto, um conjunto de mecanismos é necessário para determinar a topologia da rede virtual a partir da física [4].

No nível mais baixo, a tarefa de admitir conexões garantindo a Qualidade de Serviço (tarefa também conhecida como CAC - *Connection Admission Control*) tem recebido atenção especial. No trabalho [5], foi proposto um controlador ATM que usa uma rede neural *back-propagation* para aprender as relações entre o tráfego oferecido e o QoS. Já o trabalho [6] propõe um esquema de unidades hierárquicas, onde cada unidade é uma rede *feed-forward*. A rede neural calcula a largura de banda necessária através de medidas on-line do tráfego. O método neural obteve resultados melhores que as técnicas convencionais. Já no nível mais alto do RM, podemos citar o trabalho [7], que trata do problema de otimização do dimensionamento de sub-redes virtuais através de uma CPNN (*Convex Programming Neural Network*). Encontram-se outras aplicações de redes neurais em ATM na predição de tráfego multimídia, no roteamento multicast e no controle de fluxo.

### 3.2 Agentes Inteligentes em Gerência Pró-Ativa

Agente inteligente é definido em [8] como um “elemento independente de software responsável por uma tarefa. Para realizá-la, o agente inteligente contém algum nível de inteligência, variando de simples regras pré-definidas a máquinas de inferência. Ele age no lugar de um usuário ou de um processo, permitindo a automação de tarefas. Agentes operam de maneira autônoma (sendo freqüentemente ativados por eventos ou por tempo) e podem se comunicar com o usuário, com recursos do sistema e com outros agentes para realizar sua tarefa. Agentes mais sofisticados podem cooperar para realização de tarefas além de suas capacidades individuais. Agentes podem também ser móveis”.

A gerência pró-ativa pode fazer uso de agentes inteligentes. Ela tem como objetivo detectar situações anormais, utilizando um perfil de comportamento da rede que permita ao sistema evitar problemas. Uma atitude pró-ativa compreende medidas preventivas ou reativas de menor impacto, dentro de uma ação planejada. Esta abordagem pode ser associada a técnicas de Inteligência Artificial. Por exemplo, o trabalho [9] utiliza agentes com regras de produção e *frames* para diagnosticar situações problemáticas antecipadamente numa rede ATM.

## 4 RENATA

### 4.1 Arquitetura Funcional da RENATA

A figura 1 mostra a Arquitetura Funcional da RENATA, que possui três módulos: *Módulo de Treinamento*, *Módulo Neural* e *Módulo de Gerência*. A arquitetura define a funcionalidade de cada módulo e como estes interagem para o desenvolvimento dos agentes.

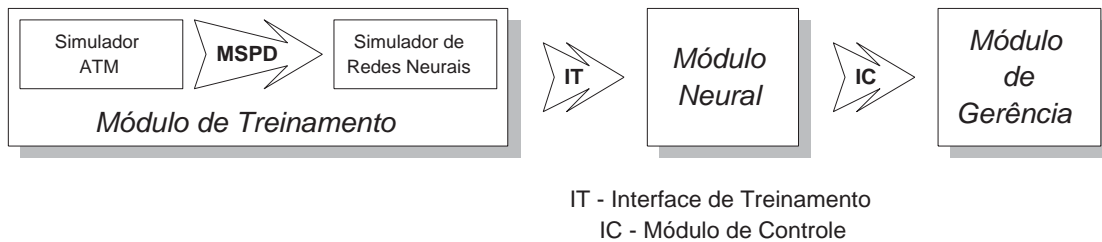


Figura 1: Arquitetura Funcional da RENATA

No Módulo de Treinamento, a rede neural é projetada, treinada e validada. Para isto, este módulo é dividido em três: o Simulador ATM, o MSPD (Módulo de Seleção e Preparação de Dados) e o Simulador de Redes Neurais. O Módulo Neural consiste na rede neural resultante da saída do Módulo de Treinamento e de informações sobre sua arquitetura e objetivo. O Módulo de Gerência é responsável pela integração e ativação do Módulo Neural, através do desenvolvimento de um agente que fornece dados para a predição da rede neural e, de acordo com sua saída, toma determinadas ações.

O treinamento da rede neural é realizado *off-line* para que esta tenha melhor desempenho em um ambiente de alta velocidade durante a fase de operação. Por isso, o Módulo de Treinamento é separado do Módulo de Gerência. Considerando o tipo de aprendizado, RENATA pode aceitar vários modelos de rede neural.

#### 4.1.1 Módulo de Treinamento - Simulador ATM

Normalmente, as redes neurais precisam de uma grande quantidade de dados de treinamento para aprenderem. Em situações reais de telecomunicações, a aplicabilidade de redes neurais depende da disponibilidade de padrões de teste, tanto de funcionamento normal quanto anormal da rede. Se não houver registros históricos, torna-se difícil a obtenção de dados, pois não se pode parar uma rede em produção apenas para testá-la com problemas. Assim, o uso de um simulador ATM na RENATA se justifica por essa dificuldade na obtenção de dados e pela flexibilidade de se simular uma rede nas mais diversas configurações.

A partir da definição do problema a ser diagnosticado pela rede neural, são feitas simulações visando à obtenção dos dados necessários para o aprendizado da rede neural. Inicialmente, é descrita a topologia da rede ATM no Simulador: comutadores, enlaces e terminais, com suas características. A partir da topologia, simulam-se aplicações executando pela rede, descrevendo carga e tipo de tráfego. Pode-se também simular falha em algum segmento ou grande carga sobre um nó para observação do comportamento da rede e, assim, obter parâmetros que caracterizem estas situações.

De acordo com o problema, determinadas opções de log no Simulador ATM são habilitadas de modo que este gere os dados que servirão como base para o treinamento da rede neural. Estes dados devem corresponder aos parâmetros que serão posteriormente monitorados por um agente numa rede ATM real.

#### **4.1.2 Módulo de Treinamento - MSPD**

Antes de serem submetidos à rede neural, os dados precisam ser selecionados, divididos e tratados. Essas funções são realizadas pelo MSPD (Módulo de Seleção e Preparação dos Dados).

Segundo [3], aproximadamente 98% destes devem ser de funcionamento normal e 2% devem caracterizar a situação que a rede neural deve detectar. A seguir, os dados são separados entre treinamento e validação da rede neural, na proporção de 80% para treinar, 10% para testar e o restante para validação da rede neural.

Depois de selecionados, os dados devem ser tratados para que possam ser submetidos à rede neural. Os dados devem ser traduzidos do formato de log do Simulador ATM para o formato do Simulador de Redes Neurais. Além disso, alguns modelos de redes neurais só aceitam entradas binárias, enquanto outros aceitam reais na escala de 0 a 1 ou -1 a +1. Nestes casos, técnicas como normalização, escalonamento e codificação binária 1-N são utilizadas. Por exemplo, o número de conexões ativas passando por um comutador é um número inteiro que deve ser escalonado para real no intervalo de 0 a 1 para ser aceito como uma entrada válida de uma determinada rede neural.

Aspectos como o tipo de treinamento devem ser também considerados na preparação dos dados. Por exemplo, se o treinamento for supervisionado, os dados devem conter também a saída desejada da rede neural para cada entrada.

#### **4.1.3 Módulo de Treinamento - Simulador de Redes Neurais**

Com o tipo de problema definido e os dados selecionados e preparados, o modelo da rede neural deve ser escolhido. A partir dessa escolha, o Simulador de Redes Neurais é utilizado para modelar a rede neural e controlar seu treinamento. Parâmetros gerais como a taxa de aprendizado e função de ativação, além de alguns específicos a cada modelo devem ser configurados. Tais parâmetros determinam a velocidade do treinamento e o grau de generalização da rede neural, entre outros fatores. O erro da rede neural deve ser monitorado durante o treinamento para determinar se a rede neural está convergindo ou se algum parâmetro foi mal escolhido. Em [3] e [10], encontram-se discussões sobre o treinamento de vários modelos de redes neurais. A meta é otimizar o desempenho da rede neural sobre os padrões de teste e validação. Para isso, recomenda-se testar seu desempenho sobre os dados de teste, enquanto o treinamento é feito. O critério de aprovação, o parâmetro que determina quando a rede neural está treinada, depende do seu modelo, podendo ser a simples proporção de acertos sobre tentativas.

Por fim, a saída do Módulo de Treinamento é a rede neural treinada e testada. A Interface de Treinamento comunica o Módulo de Treinamento ao Módulo Neural, transmitindo os dados resultantes do simulador de Redes Neurais (a matriz de pesos da rede neural). Estes pesos representam o conhecimento adquirido durante o treinamento.

#### 4.1.4 Módulo Neural

O Módulo Neural consiste na rede neural resultante da saída do Módulo de Treinamento e de informações sobre sua arquitetura e objetivo. A rede neural treinada e devidamente testada é um módulo de aplicação, que deve receber as entradas (normalizadas e codificadas) para realizar a predição sobre o estado da rede. Algum processamento sobre a saída pode ser também necessário.

O desempenho da rede neural deve ser monitorado. Se os resultados estiverem abaixo do critério de aprovação, é possível que a dinâmica da rede ATM tenha mudado: talvez por alterações na topologia ou pela introdução de novos serviços. Então, a rede neural deve ser novamente treinada. Contudo, pequenas mudanças na rede ATM devem ser absorvidas pela capacidade de generalização e adaptabilidade da rede neural.

A Interface de Controle comunica o Módulo Neural ao Módulo de Gerência. Esta interface repassa ao Módulo de Gerência as informações relativas à arquitetura da rede neural e ao seu propósito. Estas informações serão usadas no desenvolvimento do agente.

#### 4.1.5 Módulo de Gerência

O Módulo de Gerência é responsável pela integração e ativação do Módulo Neural, através do desenvolvimento de um agente que acionará a rede neural e tomará as devidas ações de acordo com suas predições, visando a ter uma atitude pró-ativa.

Antes do desenvolvimento do agente, é necessário que seja configurado no Módulo de Gerência o ambiente de gerência ATM da rede em questão: os dispositivos que serão monitorados e seus respectivos mecanismos (MIBs SNMP, ILMI e células OAM).

Quando do desenvolvimento de um agente, as informações da rede neural providas pela Interface de Controle são processadas. Para cada entrada, é definido o dado que o agente consultará para alimentá-la. Assim como, para cada saída, é determinada a ação que deve ser tomada, utilizando os mecanismos de gerência ATM previamente configurados. A seguir, o agente é gerado, devendo ser depois instalado e ativado.

O Módulo de Gerência também é responsável pelo monitoramento dos agentes, além de possibilitar o gerenciamento direto da rede. As ações tomadas pelo agente devem ser registradas para posterior análise. Além disso, o agente deve alertar o Módulo de Gerência quando a rede neural estiver abaixo de seu critério de aprovação.

## 4.2 Arquitetura Física da RENATA

A figura 2 apresenta a Arquitetura Física da RENATA, os seus módulos e como são realizadas as trocas de informações. Ela introduz um novo elemento no Módulo Treinamento: o Documentador de Redes Neurais (DocRN). Esta ferramenta é responsável pela criação do arquivo com as informações sobre a rede neural presente no Módulo Neural.

Fisicamente, o Módulo de Gerência é dividido em dois: o Módulo de Prototipação e o Módulo de Gerenciamento. O primeiro é responsável pela criação do agente a partir das informações da rede neural e da configuração dos mecanismos de gerência da rede. O outro módulo é responsável pelo monitoramento dos agentes e ainda permite acesso direto à gerência da rede.

A parte mais complexa do desenvolvimento de uma aplicação na RENATA se dá realmente no Módulo de Treinamento: toda modelagem e treinamento da rede neural para

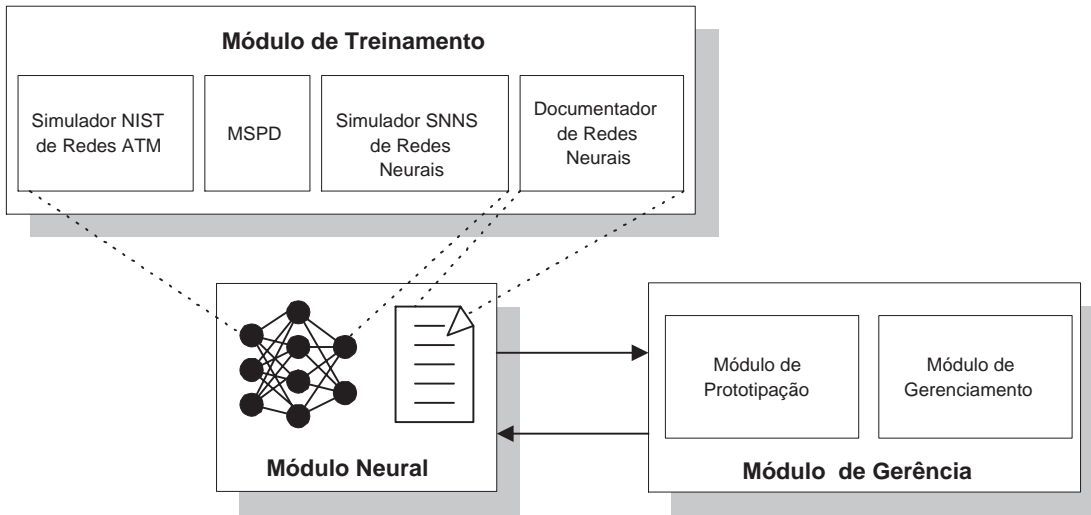


Figura 2: Arquitetura Física da RENATA

diagnosticar determinado problema. A partir do Módulo Neural, o Módulo de Gerência permite o desenvolvimento dos agentes.

O agente resultante deste processo consistirá basicamente dos mecanismos de gerência para monitorar e controlar a rede ATM (determinados no Módulo de Gerência) e do conhecimento para prever determinada situação (Módulo Neural).

A seguir, as opções para implementação de cada módulo da RENATA serão apresentadas. No caso do simulador ATM e do simulador de redes neurais, dada a complexidade de suas funções, produtos prontos são utilizados. Estes produtos são integrados visando à criação da rede neural. Para o restante do processo, foram implementadas as seguintes ferramentas: o MSPD, o Documentador de Redes Neurais e o Módulo de Gerência.

#### 4.2.1 Produtos Utilizados

##### Simulador ATM

O Simulador NIST de redes ATM foi desenvolvido no *National Institute of Standards and Technology* (NIST) para fornecer um ambiente de estudo e determinação de desempenho de redes ATM. Dentro de um ambiente gráfico interativo, o Simulador ATM permite que o usuário crie diferentes topologias e configure parâmetros da operação de cada componente. Enquanto a simulação é executada, várias medidas de desempenho podem ser mostradas na tela ou salvas em arquivo para análise posterior [11].

##### Simulador de Redes Neurais

O SNNS (*Stuttgart Neural Network Simulator*) é um simulador de redes neurais desenvolvido na Universidade de Stuttgart desde 1989. A meta do projeto é a criação de um ambiente flexível e eficiente de simulação para a pesquisa e aplicação de redes neurais [10]. O simulador pode ser usado para criar, modelar, treinar, testar, realizar podar, analisar e visualizar redes neurais de diversos modelos.



## Módulo Neural

O Módulo Neural consiste da rede neural produzida pelo Módulo de Treinamento e pela sua documentação gerada pelo DocRN. A partir da rede neural treinada e validada no SNNS, é utilizada uma ferramenta do pacote do SNNS, o compilador *snns2c*, que gera código C com a representação da rede neural.

### 4.2.2 Ferramentas Implementadas

#### MSPD

O MSPD é o módulo responsável pela seleção e preparação dos dados que serão submetidos à rede neural. O MSPD realiza a tradução do arquivo do formato do simulador NIST para o formato do SNNS, selecionando os dados que são relevantes para o aprendizado da rede neural. Além disso, o tratamento necessário do dado bruto para que este possa ser uma entrada da rede neural também é realizado pelo MSPD.

#### Documentador de Redes Neurais

Para que o agente possa alimentar a rede neural e interpretar suas saídas é necessário que o Módulo de Gerência obtenha estas informações quando da criação do agente. Estes dados são gerados a partir do Documentador de Redes Neurais (DocRN).

No DocRN, são documentadas informações sobre o objetivo da rede neural, sua arquitetura (entradas e saídas), como esta deve ser ativada e avaliada, e alguns parâmetros do seu treinamento, entre outros dados. As informações sobre as entradas definem que dados a rede neural deve receber para fazer a predição e como esses dados brutos devem ser tratados para poderem ser submetidos à rede neural. Para cada saída, deve ser fornecido a interpretação de tal predição e uma sugestão de ação a ser tomada. O DocRN foi implementado na linguagem Java, principalmente por suas características de portabilidade.

#### Módulo de Gerência

As funcionalidades do Módulo de Gerência estão divididas entre seus dois sub-módulos. O Módulo de Prototipação é responsável pela geração do código do agente a partir das informações geradas pelo DocRN e pelas informações dos mecanismos de gerência da rede ATM em questão; isto é, a configuração das fontes para as entradas da rede neural e do comportamento do agente mediante as predições da rede neural. O Módulo de Gerenciamento oferece acesso à gerência da rede ATM, além de ser responsável pelo monitoramento dos agentes gerados pelo Módulo de Prototipação.

Ambos foram implementados em Java, com auxílio da API Java SNMP da *Advent Network Management, Inc.* Esta biblioteca foi criada para permitir a implementação de *applets* e aplicações Java que utilizem o protocolo SNMP para se comunicar com nós gerenciados. Foi utilizada a versão 1.3.1 da API para SNMPv2C, dado que o agente desenvolvido na RENATA pode consultar MIBs SNMP para obter as informações necessárias à rede neural, além de também permitir seu monitoramento.

O Módulo de Prototipação gera código Java para o agente. A rede neural é acessada através de chamadas a métodos nativos. O código do agente deve ser compilado e depois instalado no dispositivo preestabelecido, que deve possuir a máquina virtual Java.

### 4.3 Utilizando RENATA

### 4.4 Acesso aos Mecanismos de Gerência ATM

Após uma fase inicial de configuração (utilizando a ferramenta RenataSetup), onde os mecanismos de gerência da rede em questão são incorporados à RENATA, os agentes e o usuário através do Módulo de Gerenciamento poderão iniciar um fluxo de células OAM e consultar as MIBs SNMP da rede através de um Navegador de MIBs. Desse modo, RENATA concilia os mecanismos de gerência ATM com a inteligência das redes neurais, visando a dotar os agentes desenvolvidos de uma atitude pró-ativa.

Além de oferecer acesso direto aos mecanismos de gerência ATM (MIBs SNMP Aplicadas ao ATM, ILMI e Células OAM), RENATA permite a seleção através desses mecanismos das informações que serão monitoradas pelo agente. Este, por sua vez, repassará essas informações devidamente tratadas para a rede neural que realizará com esses dados a predição sobre o estado da rede.

Para o caso da detecção de uma situação anormal, deve-se também configurar as ações que o agente deve tomar entre aquelas permitidas pelos mecanismo de gerência ATM. Por exemplo, se o agente poderá atuar no recurso (via SNMP, ILMI ou OAM) ou se deve apenas notificar o usuário.

### 4.5 Interface com Usuário

RENATA tem dois tipos de usuários: o *desenvolvedor*, que utiliza RENATA para criar aplicações de redes neurais para a gerência ATM; e o *administrador de redes*, que configura essas aplicações para executarem na sua rede, através da prototipação de agentes.

O desenvolvedor se preocupa com todo processo, principalmente com a modelagem e treinamento da rede neural para detectar situações anormais numa rede ATM ou para realizar tarefas como as descritas na seção 3.1. Enquanto que o administrador se preocupa com a configuração, instalação e monitoramento dos agentes. Cada usuário tem sua própria interface, diferenciadas por suas ferramentas e tarefas. A figura 3 mostra o protótipo com as suas interfaces.

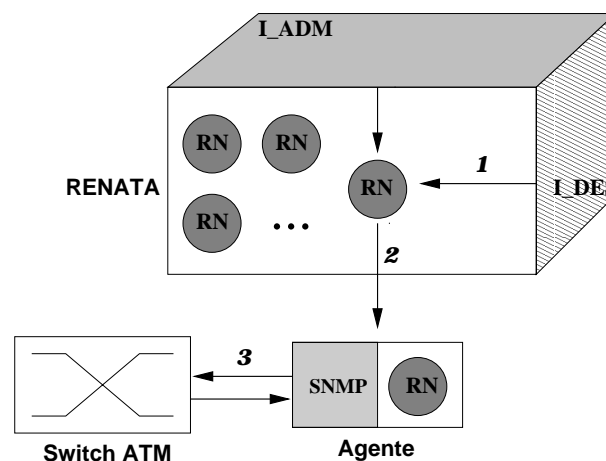


Figura 3: Protótipo da RENATA

A *I\_DES* (Interface do Desenvolvedor) permite acesso às facilidades do Módulo de Treinamento: o Simulador ATM, o MSPD, o Simulador de Redes Neurais e o Documen-

tador de Redes Neurais. A partir da I\_DES, o desenvolvedor cria as redes neurais que, posteriormente, serão executadas a partir de agentes configurados pelo administrador. A I\_ADM (Interface do Administrador) integra mecanismos de gerência ATM com as ferramentas para prototipação dos agentes, isto é, o Módulo de Gerenciamento e o Módulo de Prototipação. O elemento comum entre as interfaces é o Módulo Neural.

A figura 3 também mostra o funcionamento geral do protótipo dividido em três grandes etapas. Na etapa 1, o desenvolvedor cria um Módulo Neural, a partir das ferramentas do Módulo de Treinamento. Na etapa 2, o administrador configura o agente que conterá a rede neural. Já a etapa 3 exemplifica o funcionamento do agente desenvolvido atuando no recurso ATM através do SNMP.

A tabela 1 detalha em passos as etapas do desenvolvimento de um agente na RENATA, relacionando as ferramentas e interfaces utilizadas.

### I\_DES

---

**Passo 1:** Definição do Problema

**Passo 2:** Simulação da Rede ATM

Ferramenta: *Simulador NIST*

**Passo 3:** Preparação e Seleção dos Dados Gerados pelo Simulador ATM

Ferramenta: *MSPD*

**Passo 4:** Projeto, Treinamento e Validação da Rede Neural

Ferramenta: *Simulador SNNS*

**Passo 5:** Documentação da Rede Neural

Ferramenta: *Documentador de Redes Neurais*

### I\_ADM

---

**Passo 6:** Configuração dos Mecanismos de Gerência da Rede

Ferramenta: *RENATA\_SETUP*

**Passo 7:** Prototipação do Agente

Ferramenta: *Módulo de Prototipação*

**Passo 8:** Instalação e Ativação do Agente

**Passo 9:** Monitoramento do Agente

Ferramenta: *Módulo de Gerenciamento*

Tabela 1: Passos e Ferramentas do Desenvolvimento de um Agente na RENATA

## 5 Implementação

### 5.1 Descrição do Protótipo

Um dos objetivos deste trabalho é o desenvolvimento de um ambiente genérico de desenvolvimento de agentes baseados em redes neurais. Todas ferramentas foram implementadas em Java. Foi utilizada a versão 1.1.7 do JDK (*Java Development Kit*), fornecido pela *Sun Microsystems*. Isto garante que RENATA executa sobre qualquer plataforma que suporte a máquina virtual Java. Para o projeto da interface visual das ferramentas, foi utilizada a versão 1.2 do *Borland JBuilder*. O código da interface é modificado para não ficar dependente de nenhuma classe da Borland.

Como é praticamente impossível a criação de um MSPD genérico, este módulo não foi implementado por ser intrinsecamente específico a cada aplicação. Por isso, cabe ao desenvolvedor codificar o MSPD adequado à sua rede neural. Foram implementados o *DocRN*, o *RenataSetup*, o *Módulo de Gerenciamento* e o *Módulo de Prototipação*<sup>1</sup>.

### 5.1.1 DocRN

DocRN é a ferramenta utilizada pelo desenvolvedor para descrever as características e funcionalidades da rede neural criada no Módulo de Treinamento.

Parâmetros importantes como o Intervalo de Predição (período de tempo que o agente deve esperar para ativar a rede neural, em milissegundos), o Modo de Ativação (podendo ser: sempre, quando for pedido ao agente ou dependendo de algum parâmetro na rede) e o Critério de Aceitação (como deve ser avaliado o desempenho da rede neural) são fornecidos no DocRN. Alguns parâmetros do treinamento são armazenados para o caso de ser necessário um retreinamento da rede neural. Tais valores poderiam ser alterados ou utilizados novamente.

Na configuração arquitetural, o desenvolvedor inicialmente fornece o número de entradas e saídas. Para cada entrada (figura 4), o desenvolvedor deve fornecer sua descrição, o tipo do dado original e o tipo da entrada permitida pela rede neural (real ou binária). O tipo do dado original pode ser real ou inteiro (simples, elemento de um vetor ou resultado da operação de outros atributos), string ou uma data. A combinação do tipo original com o tipo da entrada determina como vai ser feita a preparação dos dados. Já para cada saída da rede neural, o desenvolvedor descreve seu significado e uma sugestão de ação em caso de sua ativação.

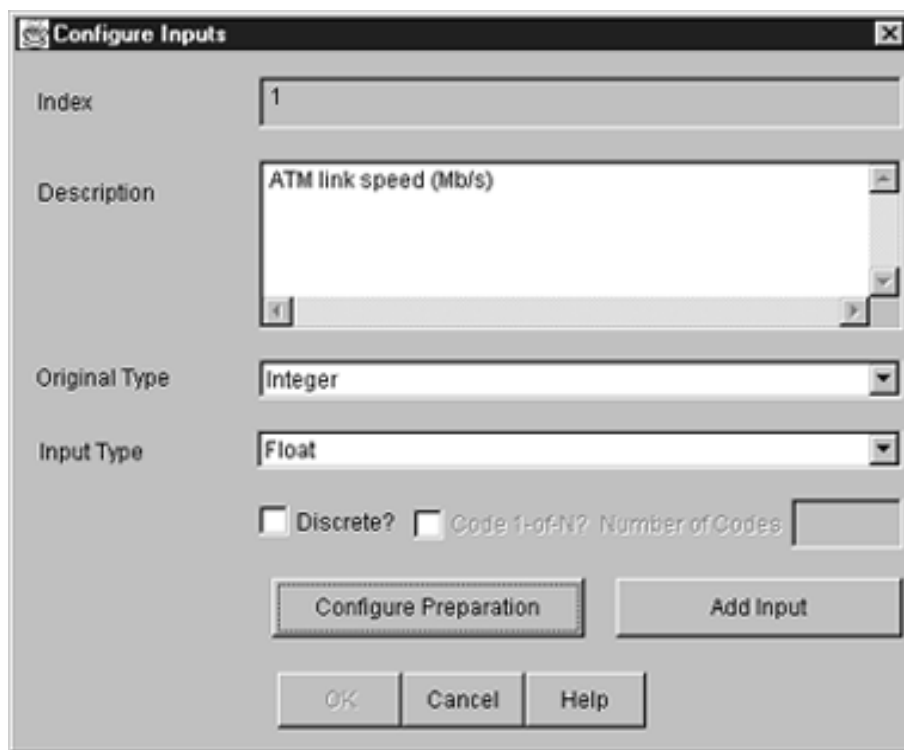


Figura 4: Configuração das Entradas da Rede Neural no DocRN

<sup>1</sup>Maiores detalhes sobre este protótipo podem ser encontrados em <http://www.lia.ufc.br/~blau>.

### 5.1.2 RenataSetup

RenataSetup é ferramenta onde o administrador informa quais são as máquinas que serão gerenciadas pela RENATA e quais são os mecanismos de gerência nelas presentes.

Para cada dispositivo que RENATA pode gerenciar, o administrador fornece o nome e tipo (comutador, estação ou roteador) do dispositivo, além de seu endereço IP e ATM. Em seguida, os mecanismos de gerência ATM (MIBs SNMP, ILMI e funções OAM) presentes na máquina em questão devem ser fornecidos. Para cada MIB, o administrador deve informar a *community* de leitura e a de escrita, além do número da porta do agente SNMP responsável por esta MIB. Para ILMI, o administrador deve configurar a *community* de cada IME (local e remoto), além da porta de seu agente SNMP. Em relação às células OAM (F4 e F5), o administrador deve informar se estas são permitidas neste dispositivo. Em caso positivo, o *driver* específico para acesso a esse mecanismo deve ser informado.

### 5.1.3 Módulo de Gerenciamento

O Módulo de Gerenciamento é a ferramenta que possibilita acesso direto aos mecanismos e informações de gerência ATM, além de permitir o monitoramento dos agentes gerados pela RENATA.

Para cada dispositivo configurado no RenataSetup, é possível acessar suas MIBs SNMP através de um Navegador de MIBs. Versões adaptadas deste navegador são utilizadas para acesso a ILMI MIB (IME local ou remota) e para permitir o monitoramento dos agentes gerados pela RENATA instalados naquele dispositivo, através de consultas a uma MIB específica que foi desenvolvida para este fim.

### 5.1.4 Módulo de Prototipação

O Módulo de Prototipação é a ferramenta que gera os agentes, a partir das informações de gerência da rede geradas pelo RenataSetup e das informações da rede neural geradas pelo DocRN. Depois de carregar o arquivo gerado pelo DocRN, para cada entrada da rede neural, o administrador configura onde o agente deve coletar essa informação (numa MIB SNMP ou na ILMI MIB, SET na MIB do agente RENATA, via células OAM ou seja um valor fixo e pré-determinado).

De acordo com o significado de cada saída, o administrador determina que ação o agente deve tomar: nenhuma, SET em alguma variável, mandar um *trap* para um gerente ou mandar um e-mail para o administrador comunicando o problema.

## 5.2 Descrição do Agente

Nesta seção, o funcionamento geral do agente é explicado. Na inicialização do agente, duas *threads* são ativadas: a PredictionThread (responsável pela integração deste com a rede neural) e a ReceiverThread (responsável por atender pedidos exteriores de informação). Dependendo do modo de ativação e do intervalo de predição da rede neural, a PredictionThread ativa outras *threads* que são responsáveis pela coleta de informação para as entradas da rede neural. Paralelamente, estas atualizam o vetor de entradas que depois é submetido à rede neural. No caso da origem da entrada ser um GET, foi obtida boa otimização em virtude da abertura do *socket* e da montagem da PDU serem realizadas apenas na inicialização da *thread*. Durante as predições, o tempo de resposta ficou praticamente reduzido ao *round-trip* entre os dois dispositivos.

A rede neural faz sua predição, atualizando o vetor de saídas. Depois de decidida qual saída foi ativada, a ação correspondente é tomada a partir da ativação de outra *thread*, a *ActionThread*. Em seguida, a ação é registrada e depois avaliada. Deste modo, RENATA faz grande uso do mecanismo de *threads* do Java, o que permite grande otimização no funcionamento do agente, facilitando sua atuação num ambiente de alta velocidade.

Para permitir o monitoramento do agente, inclusive por outros sistemas de gerência, foi desenvolvida uma MIB de acordo com a SMIV2 (*Structure of Management Information*) com as informações relativas ao agente e à rede neural. A figura 5 mostra a estrutura da MIB RENATA, que corresponde às informações fornecidas anteriormente mais aquelas relativas às operações executadas pelo agente.

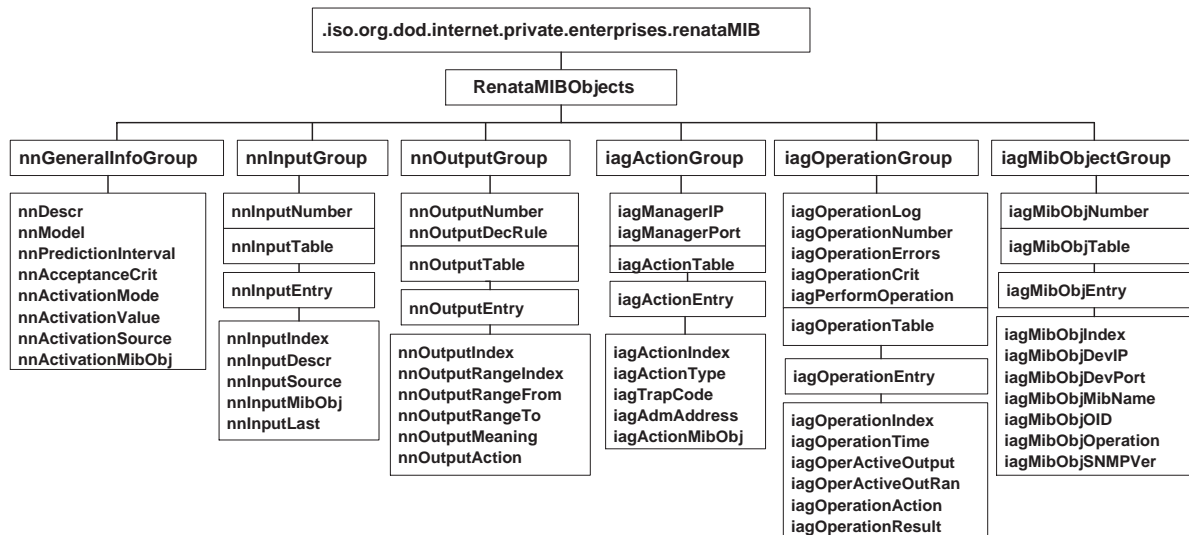


Figura 5: MIB RENATA

### 5.3 Estudo de Caso

Uma extensão do modelo agente-gerente foi proposta em [12], através da introdução de um “subagente adaptativo”. Devido às redes neurais, o subagente desta proposta possui características adaptativas que o permite reagir a possíveis falhas em uma rede local. Este trabalho também propôs um método para a prototipação e treinamento do subagente, através do desenvolvimento de duas bibliotecas C++, uma para suporte ao desenvolvimento de agentes SNMP e outra para controlar o treinamento de redes neurais *feed-forward*. A partir destas classes, o trabalho [13] implementa um agente SNMP que utiliza uma rede neural para realizar CAC de conexões VBR em redes ATM.

Já no protótipo RENATA, esta mesma rede neural foi treinada e validada no SNNS, a partir de simulações no NIST para obtenção dos dados de treinamento. A partir da documentação da rede neural e da configuração da RMAV-Fortaleza (Rede Metropolitana de Alta Velocidade), o agente foi implementado através do Módulo de Prototipação. Este estudo de caso visou apenas a testar a viabilidade e as funcionalidades do protótipo implementado.

A utilização do simulador ATM fornece dados mais próximos da situação real, ao invés da aplicação direta de uma regra na classificação dos padrões de treinamento. Apesar de

acarretar numa relativa perda inicial de desempenho, Java garante portabilidade à RENATA, além de facilitar novas extensões. A possibilidade do agente coletar as informações através dos mecanismos de gerência representa maior flexibilidade e possibilita uma visão maior da rede. A criação de uma MIB específica ao agente desenvolvido na RENATA permite seu monitoramento, inclusive por outros sistemas de gerência. A abstração de apenas fornecer e preparar as entradas e depois interpretar as saídas, torna o protótipo bastante flexível, permitindo que este aceite vários modelos de redes neurais.

## 6 Conclusões

RENATA apresenta-se como um ambiente flexível de experimentação do uso de redes neurais na gerência ATM, permitindo o desenvolvimento de agentes inteligentes. Numa abordagem convencional, o usuário deve se preocupar com todos detalhes da implementação do agente. Já RENATA provê facilidades para a integração da rede neural no contexto de gerência ATM, podendo servir para testar soluções neurais em software.

Diversas MIBs têm sido analisadas, constatando-se uma carência de Objetos Gerenciados que modelem o gerenciamento de tráfego e de congestionamento em redes ATM [14]. Essa limitação se reflete também nesta primeira versão do protótipo. Contudo, as extensões da MIB RMON para ATM e a interação com rotinas de monitoramento de baixo nível devem amenizar essas dificuldades.

O aspecto de comunicação inter-agentes RENATA é uma possível extensão deste trabalho pois, neste primeiro momento, não há comunicação entre os agentes. Uma primeira idéia seria possibilitar essa comunicação, seja através de alguma linguagem de comunicação inter-agentes como KQML (*Knowledge and Query Manipulation Language* ou através do CORBA (*Common Object Request Broker Architecture*), que garante transparência de acesso e interoperabilidade. Deste modo, os agentes da RENATA poderiam ser objetos CORBA, que usariam essa infra-estrutura para comunicação com outros objetos (agentes ou gerentes) e o SNMP para acesso aos recursos ATM. Entretanto, para algumas aplicações deve ser avaliada a sobrecarga causada pelo CORBA, que poderia comprometer a eficácia da solução, que pretende ser pró-ativa.

Dentre outras possibilidades, destaca-se o desenvolvimento de novos agentes a partir do protótipo. RENATA possibilitará uma gama de novas aplicações de redes neurais na gerência ATM, a partir da criação de agentes baseados deste de sua concepção nas propostas e funcionalidades deste trabalho.

RENATA nasceu no contexto do projeto NEURALTEL (*Neural Networks and Telecommunications*). O NEURALTEL é financiado pelo programa ALFA (*Amérique Latine - Formation Académique*) da Comunidade Européia e visa a promover a interação entre pesquisadores na América Latina e na Europa, através da facilitação da mobilidade de professores, de alunos de graduação e pós-graduação entre os países envolvidos.

O trabalho desenvolvido também serviu de base para a concepção do livro “*Introdução à Gerência de Redes ATM*” [1], livro-texto para o curso de mesmo nome apresentado durante o XVI Simpósio Brasileiro de Redes de Computadores, realizado em maio de 1998 no Rio de Janeiro.

## Referências

- [1] M. Oliveira, M. Franklin, A. Nascimento, and M. Vasconcelos, *Introdução à Gerência de Redes ATM*. Editora CEFET-CE, segunda ed., 1998.
- [2] The ATM Forum, af-ilmi-0065.000, *Integrated Local Management Interface (ILMI) Specification Version 4.0*, September 1996.
- [3] J. P. Bigus, *Data Mining with Neural Networks*. McGraw-Hill, 1996.
- [4] S. Naughton, “A Review of the Resource Management Task in ATM Networks,” Tech. Rep. TCD-CS-95-06, Department of Computer Science Trinity College, September 1995.
- [5] A. Hiramatsu, *Handbook of Neural Computing*, ch. ATM Network Control by Neural Network. Institute of Physics Publishing and Oxford University Publishing, 1997.
- [6] S. A. Youssef, I. W. Habib, and T. N. Saadawi, “A Neurocomputing Controller for Bandwidth Allocation in ATM Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 191–199, February 1997.
- [7] A. Faragó, J. Biró, T. Henk, and M. Boda, “Analog Neural Optimization for ATM Resource Management,” *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 156–164, February 1997.
- [8] T. Magedanz, K. Roethermel, and S. Krause, “Intelligent agents: An emerging technology for next generation telecommunications?,” in *INFOCOM’96*, March 1996.
- [9] F. A. da Silva Cruz, M. S. M. A. Notare, F. Gauthier, B. G. Riso, and C. B. Westphall, “Uso de Inteligência Artificial na Implementação de um Sistema de Gerência Proativo para Redes ATM,” in *XXIII Conferencia Latinoamericana de Informática*, (Valparaíso, Chile), November 1997.
- [10] U. of Stuttgart, *SNNS - Stuttgart Neural Network Simulator - User Manual, Version 4.1*, 1995.
- [11] N. Golmie, A. Koenig, and D. Su, *The NIST ATM Network Simulator - Operation and Programming, Version 1.0*, August 1995.
- [12] E. M. Vieira, “Métodos para Desenvolver Agentes Adaptativos em Gerência de Redes Usando Redes Neurais,” Master’s thesis, Federal University of Santa Catarina (UFSC), March 1997.
- [13] J. A. Arantes, E. M. Vieira, and C. B. Westphall, “Construção de um Agente SNMP para a Gerência de Redes ATM Utilizando Redes Neurais,” in *XVI Simpósio Brasileiro de Redes de Computadores*, pp. 179–191, May 1998.
- [14] L. M. R. Tarouco and M. de Fátima Webber do Prado Lima, “Análise do Gerenciamento dos Mecanismos de Policiamento de Tráfego em Redes ATM Através de Objetos Gerenciados,” in *XVI Simpósio Brasileiro de Redes de Computadores*, pp. 319–338, May 1998.