

# Utilizando Redes Neurais para Predição de Falhas em Links de Redes Ópticas

Carlos Hairon R. Gonçalves<sup>1</sup>, A. Mauro Oliveira<sup>1</sup>,  
Rossana M. C. Andrade<sup>2</sup> e Miguel F. de Castro<sup>3</sup>

<sup>1</sup> Laboratório Multiinstitucional de Redes de Computadores e Sistemas Distribuídos Centro Federal de Educação Tecnológica do Ceará (LAR/CEFET-CE)  
Av. 13 de Maio, 2081, Benfica, Fortaleza, Ceará, Brasil, 60.531-040  
<mailto:{hairon,mauro}@lar.cefet-ce.br>

<sup>2</sup> Universidade Federal do Ceará, Departamento de Computação (DC/UFC)  
Campus do Pici, Bloco 910, Fortaleza, Ceará, Brasil, 60.455-760  
[rossana@ufc.br](mailto:rossana@ufc.br)

<sup>3</sup> Institut National des Télécommunications  
9 rue Charles Fourier 91011 Evry Cedex - France  
[miguel.castro@int-evry.fr](mailto:miguel.castro@int-evry.fr)

## 1 Introdução

No final da década de 90, ATM era visto como a tecnologia que em poucos anos se tornaria um padrão de *facto* para redes globais, metropolitanas e até mesmo para redes locais. Entretanto, segundo [1], há uma forte tendência para o uso de IP + GMPLS (*Generalized Multiple Protocol Label Switching*) sobre redes DWDM (*Dense Wavelength-Division Multiplexing*) em substituição ao seguinte modelo anterior de quatro camadas: IP como camada de aplicação e serviços; ATM como responsável pela engenharia de tráfego; SONET/SDH para o transporte de dados; e DWDM como infra-estrutura física. Atualmente, o ATM ainda resiste como uma solução viável para *backbones* de redes metropolitanas, mas com o surgimento do IPv6 e das tecnologias MPLS (*Multiple Protocol Label Switching*) e DiffServ (*Differentiated Services*), a tendência é cada vez mais o fortalecimento destes modelos, os quais prometem inserir formas de garantias de qualidade de serviço no protocolo IP.

Independentemente da tecnologia de redes utilizada, a gerência Internet baseia-se no paradigma Gerente x Agente e utiliza o protocolo SNMP (*Simple Network Management Protocol*), o qual centraliza no gerente o processamento de informações e, conseqüentemente, o processo de tomada de decisão, deixando os agentes com capacidades computacionais limitadas e praticamente sem autonomia. Uma alternativa para solução deste problema é o uso de Agentes Inteligentes, uma vez que possuem maior autonomia e poder de processamento [4][10]. Agentes Inteligentes podem ser executados próximos ou no próprio objeto gerenciado, fazendo inferências segundo os dados coletados. Desta forma, há uma economia de banda passante, pois a troca de informação entre gerente e agente é diminuída sensivelmente. O uso desta tecnologia traz uma maior flexibilidade se comparada aos tradicionais métodos de gerência de rede. Outro diferencial é a garantia de uma maior escalabilidade, pois parte da carga de tarefas destinadas aos gerentes, pode ser agora dividida entre os agentes. Isto faz com que os gerentes não necessitem realizar tarefas em demasia, diminuindo o custo computacional e operacional.

No caso do uso de agentes pró-ativos, pode-se obter uma economia de banda ainda maior, pois diminuem ainda mais a comunicação gerente/agente. O objetivo deste tipo de agente é detectar situações anormais, comparando o estado atual da rede com um perfil de bom comportamento da mesma. Este perfil deve permitir ao sistema

de gerência evitar problemas. Uma atitude pró-ativa compreende medidas preventivas ou reativas de menor impacto, dentro de uma ação planejada [4]. Técnicas de Inteligência Artificial podem ser usadas para viabilizar o caráter pró-ativo de um agente.

Dentre os mecanismos de Inteligência Artificial usados como máquina de inferência para agentes inteligentes, as Redes Neurais aparecem como uma tecnologia viável, motivadora de inúmeras pesquisas [10][13][15][16]. As exigências impostas pelas características dos agentes inteligentes, bem como pelas especificidades impostas pelo tráfego das diferentes tecnologias de redes são requisitos solucionáveis pelas redes neurais, devido as suas características de aprendizado, generalização, adaptabilidade, robustez e tolerância a falhas.

Em [4], um ambiente de apoio a gerência ATM utilizando agentes inteligentes é apresentado comprovando as vantagens da utilização de agentes inteligentes baseados em redes neurais para a gerência pró-ativa de redes ATM. Este ambiente, denominado de RENATA, foi desenvolvido no Laboratório Multiinstitucional de Redes de Computadores e Sistemas Distribuídos do Centro Federal de Educação Tecnológica do Ceará (LAR CEFET-CE).

A proposta apresentada neste resumo estendido tem por objetivo desenvolver uma nova versão deste ambiente, o RENATA 2, cuja evolução compreenderá o suporte a gerência pró-ativa de redes IP, MPLS e GMPLS. Dentre possíveis tarefas realizadas pelos agentes gerados pelo ambiente RENATA 2 estão o controle de tráfego e a alocação de caminhos entre os *hosts* de uma rede de computadores. Na próxima seção apresentaremos mais detalhes sobre esta proposta.

## **2 Uma Proposta para utilização de redes neurais para predição de falhas em links de redes ópticas**

Como citado na seção anterior, o modelo ATM é baseado em quatro camadas. Cada uma dessas camadas possui várias sub-camadas de gerenciamento contendo suas próprias rotinas. Em particular, o gerenciamento de falhas é uma das tarefas mais redundantes, por ser tratado em todas as camadas. Nos equipamentos de transmissão pode-se obter informações como: potência óptica e temperatura do equipamento. Na camada DWDM, obtém-se informações sobre qualidade do sinal óptico, com SNR (*Signal-to-Noise Ratio*) e *crosstalk* através de um equipamento de teste como, por exemplo, um analisador de espectro. Em camadas superiores obtém-se informações mais detalhadas acerca da qualidade do sinal como a taxa de erros de bits BER específica para cada tecnologia de transmissão em uso.

A arquitetura em múltiplas camadas não tem se mostrado como uma solução satisfatória em relação a escalabilidade, geralmente, há uma camada que limita a expansão de toda a rede. Outro fator determinante é o custo, seja de implantação, e/ou de manutenção da rede. Como mencionado anteriormente, uma tendência confirmada por [1] é a simplificação do modelo de quatro camadas, onde IP/GMPLS utilizam o DWDM como infra-estrutura física de transmissão.

Entretanto, o roteamento em redes MPL(*ambda*)S sobre DWDM é um dos problemas que limitam o uso deste protocolo em enlaces DWDM, em virtude da forma lenta oferecida por sua restauração reativa a falhas. Uma solução para tal problema é o uso de mecanismos de proteção 1+1, onde um *link* é utilizado como espelho para prover recuperação imediata de falhas. Tal proposta é bastante onerosa, pois um canal deve ser alocado de forma exclusiva para fornecer a infra-estrutura necessária. O uso de Redes Neurais Artificiais (RNAs) pode prever situações

problemáticas em um *link*, o que permite a tomada de atitudes pró-ativas antes que o problema se agrave. Isto poderá possibilitar que um mecanismo de restauração 1:n possa se aproximar do modelo 1+1, uma vez levantada a possibilidade de ocorrência de falha, o canal de *backup* passaria a funcionar como espelho do *link* supostamente problemático.

O propósito da RNA é prever um estado de falha em redes IP/GMPLS sobre DWDM. Tal rede neural servirá de máquina de inferência para agentes inteligentes que servirão de monitores do tráfego e sinalizadores de eventuais problemas. De acordo com [8], há duas classes de componentes de redes que podem sinalizar possíveis problemas, componentes de hardware, cujas falhas têm de ser identificadas, uma vez que degradam ou interrompem o sinal de transmissão e equipamentos de monitoramento que estão presentes em várias camadas para proverem informações adicionais sobre a qualidade do sinal.

Nesta proposta, inicialmente, faremos um estudo dos componentes de hardware e equipamentos de monitoramento com seus respectivos parâmetros, os quais possam ser candidatos a variáveis de entrada de uma RNA.

Em seguida, trabalharemos com o simulador escolhido para a versão 2 do RENATA, o NS (*Network Simulator*) [12]. Tal escolha foi feita por que o NS é uma ferramenta aberta, de amplo uso na comunidade científica e trabalha com diversos tipos de tecnologias de redes.

Apesar do NS oferecer facilidades ao trabalho proposto, o mesmo não oferece um tratamento de erros apropriado à resolução do problema de se gerar agentes inteligentes baseados em redes neurais aplicados ao gerenciamento de falhas. O tratamento de erros proposto pelo NS é imperativo, por exemplo, para simular a queda de um *link* entre os LSRs 4 e 6 no instante 4,5s da simulação de uma rede MPLS, deve-se executar o seguinte comando: `$ns rtmovel-at 4.5 down $LSR6 $LSR4`. Neste exemplo, nenhum fator relacionado com a simulação contribui para a queda do link, o comando em questão reflete apenas a vontade do usuário, ou seja, de forma imperativa tal comando é passado ao NS, que por sua vez o executa sem nenhuma restrição e sem nenhum relacionamento com o mundo real. Para se treinar uma rede neural é necessário uma base de conhecimento que se aproxime ao máximo possível do mundo real ou, de modo ideal, o reflita fielmente. Portanto, como o tratamento de erros proposto pelo NS não é adequado para a geração de um banco de exemplos essenciais ao aprendizado das redes neurais propostas por este trabalho, é necessário o desenvolvimento de uma ferramenta que supra este problema, tal software está em desenvolvimento e está sendo denominado de Gerador de Perturbações (GDP).

O GDP simula o comportamento dos principais dispositivos de uma rede de fibra ótica, levando em consideração as variáveis físicas que ditam o comportamento de tais dispositivos. Os equipamentos simulados pelo Gerador de Perturbações são Transmissor Óptico, Fibra Óptica e Receptor Óptico. Esses equipamentos são ditos principais por estarem obrigatoriamente presentes em redes de fibra ótica sejam diretamente no próprio equipamento, ou indiretamente, por exemplo, um transmissor que faz parte de um multiplexador DWDM. O gerador de perturbações está sendo desenvolvido em código Java puro o que lhe permite ser multiplataforma.

Finalmente, uma solução para o problema de re-roteamento rápido após uma falha ocorrida em um link transmissão do MPLS, quando este é usado sobre redes ópticas (MPLS ou GMPLS), será implementada com agentes inteligentes. Estes agentes são baseados em redes neurais e podem prever falhas em links IP+GMPLS sobre DWDM e desta forma garantir agilidade no re-roteamento destas redes.

### 3 Conclusão

Um dos problemas atuais do MPLS, quando este é usado sobre redes ópticas (MPλS ou GMPLS), é o re-roteamento rápido após uma falha ocorrida em um link transmissão. Esta proposta tem por objetivo aplicar agentes inteligentes baseados em redes neurais para prever essas falhas e, desta forma, garantir agilidade no re-roteamento destas redes.

Este trabalho, inicialmente, está se concentrando no mecanismo de inferência do agente inteligente que monitorará uma rede em particular. A proposta tem sido desenvolvida em parceria LAR/CEFET-CE e MCC/UFC. Um estudante de mestrado e estudantes de iniciação científica estão envolvidos atualmente nesta proposta.

A forma como o agente deve sinalizar e interagir com o sistema de comunicação e o processo de criação do mesmo não estão sendo contemplados nesta primeira fase de desenvolvimento já em andamento e deverão ser concretizados em trabalhos futuros.

### 4 Bibliografia

- [1] BANERJEE, Ayan et al., “Generalized Multiprotocol Label Switching: An Overview of Routing and Management Enhancements”. *IEEE Communications Magazine*, volume 39, número 1, p. 145-150. Janeiro de 2001.
- [2] BIGUS, Joseph P, BIGUS, Jennifer, “Constructing Intelligent Agents with Java – A Programmer’s Guide to Smarter Applications”. Página 146, Wiley Computer Publishing, 1998.
- [3] DOVERSPIKE, Robert, YATES, Jennifer. “Challenges for MPLS in Optical Network Restoration”. *IEEE Communications Magazine*, volume 39, número 2, p. 89-96. Fev. de 2001.
- [4] FRANKLIN, Miguel. “Redes Neurais na Estimativa da Capacidade Requerida em Comutadores ATM”. Dissertação de Mestrado, Universidade Federal do Ceará, Abril de 1999, 138 f.
- [5] KARTALOPOULOS, S.V. *Fault Detectability in DWDM – Toward Higher Signal Quality & System Reliability*. New York: IEEE Press, 2001. 165p.
- [6] LAWRENCE, Jeremy. “Designing Multiprotocol Label Switching Networks”. *IEEE Communications Magazine*, volume 39, número 7, p. 134 – 142. Julho de 2001.
- [7] MARQUES, A.L.; SILVA, H.J.A. **Circuito de comando para laser semiconductor – modulação FSK directa**, Coimbra: Instituto de Telecomunicações. Disponível em <<http://www.it.uc.pt/oc/ocpub/am97cp02.pdf>>. Acesso em: 11 de jan. de 2003.
- [8] MAS, Carmen, THIRAN, Patrick. “An Efficient Algorithm for Locating Soft and Hard Failures in WDM Networks”. *IEEE Journal*, volume 18, número 10, p. 1900 – 1991, Outubro de 2000.
- [9] N. Golmie, F. Mouveaux, L. Hester, Y. Saintllan, A. Koeing e D. 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.
- [10] NASCIMENTO, A.S. Desenvolvendo Agentes Inteligentes para a Gerência Pró-Ativa de Redes ATM. 1999. 166f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal do Ceará, Fortaleza.
- [11] NAUGHTON, S.; SOMERS, F.; *Asynchronous Transfer Mode (ATM) Source Frame Prediction Using Neural Networks*. In: INNC - IRISH NEURAL NETWORKS CONFERENCE. 1995.
- [12] PARC (Palo Alto Research Center), LBNL (Lawrence Berkeley National Laboratory) e UCB (University of California). NS (*Network Simulator*). <http://www.isi.edu/nsnam/ns>. Acesso em 10 dez. 2002.
- [13] SOTO, C.P. **Redes Neurais Temporais para o tratamento de Sistemas Variantes no Tempo**. 1999. 112f. Dissertação (Mestrado em Engenharia Elétrica) – PUC/RJ, Rio de Janeiro.
- [14] STEVENS, W. Richard. “TCP/IP Illustrated Volume 1, The Protocols”, Editora Addison Wesley, Julho de 1997.
- [15] TAFNER, M.A.; RODRIGUES I.W. **Redes Neurais Artificiais, Introdução e Princípios de Neurocomputação**. ed.1. Blumenau: EKO, 1996. 199p.
- [16] The MPLS Resource Center, [www.mplsresource.com/mplsfaq](http://www.mplsresource.com/mplsfaq). Acesso em 06 jun. 2002.
- [17] University of Stuttgart – Institute for Parallel and Distributed High Performance Systems (IPVR). SNNS (*The Stuttgart Neural Network Simulator*). Version 4.0, 1995.