

# Metodologia para o Desenvolvimento de uma Plataforma de Gerência de Redes Utilizando Agentes Inteligentes

Ricardo Alexandre Reinaldo de Moraes e Analúcia Schiaffino Morales De Franceschi

**Resumo**— O presente trabalho apresenta uma metodologia para o desenvolvimento de uma plataforma de gerência de redes utilizando agentes inteligentes. São descritos experimentos iniciais que comprovaram a aplicação de técnicas de inteligência artificial (IA) na solução de problemas distribuídos. Os experimentos foram realizados com base em um trabalho anterior que investigou o emprego das técnicas de IA na área de gerência de redes. A plataforma está sendo desenvolvida pelo grupo de pesquisa de redes de computadores da UNIPLAC - Universidade do Planalto Catarinense, em Lages, SC.

**Palavras-Chave**—Redes de Computadores, Inteligência Artificial, Agentes Inteligentes.

**Abstract**— The present work presents a methodology for the development of the network management platform employing intelligent agents. There are initial experiments that proved the artificial intelligence (AI) application in the distributed problem-solving. These experiments were based on a earliest work which investigated the employment of the AI techniques in the network management area. The computer network work group has been developing such management platform in the UNIPLAC - Universidade do Planalto Catarinense, in Lages, SC.

**Index Terms**—Computer Networks, Artificial Intelligence, Intelligent Agents.

## INTRODUÇÃO

Segundo Aleksander e Morton [1], os computadores estão presentes no cotidiano de tal forma que se esquece que a sua estrutura é baseada na visão de apenas uma pessoa. Normalmente, é assim que o processo organizacional é realizado. O autor refere-se a arquitetura de John von Neumann que é baseada em aritmética simples e lógica unitária que opera sobre os dados da memória. Isto significa que, a maioria dos problemas são solucionados através de algoritmos baseados em instruções. No entanto, ao considerar os dias atuais, analogamente ao que o autor citou, as redes de computadores também fazem parte da vida das pessoas.

Ricardo Alexandre Reinaldo de Moraes e Analúcia Schiaffino Morales De Franceschi, Departamento de Ciências Exatas e Tecnológicas, Grupo de Pesquisa em Redes de Computadores e Telecomunicações, Universidade do Planalto Catarinense, Lages SC, Brasil, E-mails: ricardo@uniplac.net, analuciadf@uniplac.net.

Estas muitas vezes nem notam que existem milhões de conexões e equipamentos de interconexão de redes, fazendo a ligação de um site a outro. Não faz sentido solucionar problemas complexos que envolvam diversos equipamentos de forma algorítmica. Analisando as técnicas convencionais e as baseadas em inteligência artificial, diversas pesquisas buscam soluções de problemas distribuídos na área de gerência de redes. Através dos métodos convencionais, a tarefa de gerenciar e tomar decisões sobre equipamentos em uma rede ainda continua dependendo exclusivamente do conhecimento do administrador de redes. No desenvolvimento de soluções nos processos de gerência, destaca-se a utilização de agentes inteligentes nos processos de tomadas de decisões em ambientes de gerência de redes. O conceito de agentes em IA surgiu no projeto de compreensão da linguagem HEARSAY-II quando surgiu o primeiro modelo (chamado Quadro Negro) para o problema de coordenação de ações entre várias partes do programa de IA. No entanto, um agente inteligente deve ter como característica a capacidade de assumir a função de outro agente que falhou [18].

Nos últimos anos, o paradigma de agentes tem aumentado significativamente, Hayes [12] atribui este crescimento à características como flexibilidade, modularidade e aplicabilidade na solução de uma variedade de problemas. O autor atribui um aumento parcial aos desenvolvimentos tecnológicos na área de computação distribuída e robótica. Estas áreas necessitam de agentes para modelar soluções adequadas, principalmente para problemas distribuídos. Paralelamente, tem-se um crescimento de paradigmas de programação orientada-a-objetos, em parte devido à importância deste tipo de programação na estruturação de sistemas baseados em agentes.

No ambiente de gerência de redes existem três conceitos importantes: gerentes, agentes e objetos gerenciados. "...Um gerente pode obter informações atualizadas sobre os objetos gerenciados e controlá-los. Para isso, transmite operações de gerenciamento aos agentes. Um agente executa operações de gerenciamento sobre objetos gerenciados. Pode, ainda, transmitir ao gerente as notificações emitidas pelos objetos gerenciados..." [5]. No entanto, estes gerentes e agentes não possuem nenhuma autonomia, ou seja, não participam do processo de tomada de decisões. A tomada de decisões é normalmente realizada por um especialista conhecido como administrador da rede. Segundo Lesser [14], a autonomia de

um agente está relacionada a capacidade do agente em tomar suas próprias decisões sobre as atividades que devem ser realizadas para alcançar seu objetivo. A autonomia pode ser limitada por políticas embutidas no próprio agente por seu projetista, ou como resultado de interações cooperativas entre outros agentes.

Uma preocupação no desenvolvimento de agentes que utilizam técnicas de IA diz respeito a complexidade de redes neurais. Entenda-se por complexidade a quantidade de recursos necessários para solucionar determinados tipos de problemas. Neste sentido, a primeira grande contribuição foi feita por Minsky e Papert. Eles provaram que uma rede neural direta precisa ter uma camada intermediária entre a camada de entrada e a de saída para solucionar problemas não-linearmente separáveis [15]. Outra contribuição foi a de Barreto que provou ser mais adequado utilizar redes recorrentes do que redes diretas para resolver problemas dinâmicos [2]. A aplicação de uma abordagem estática na solução de problemas dinâmicos causa dois inconvenientes, o primeiro porque não permite cobrir todos os diferentes estados de um sistema dinâmico. E o segundo, porque seria necessário uma rede neural muito extensa para cobrir todos os estados possíveis, e desta forma seria muito difícil convergir a uma solução.

No entanto, é possível encontrar na literatura científica vários trabalhos com abordagens de agentes inteligentes em ambientes de redes de computadores que não se preocupam com características estáticas e dinâmicas dos problemas que estão sendo abordados. Exemplos de trabalhos voltados a criação de agentes para auxiliar na busca por informações na Internet são encontrados em [4][16]. Outros agentes que possuem aplicação na área de gerência de redes são descritos em [6][17]. A maioria destes agentes foram desenvolvidos utilizando regras de produção que podem ser implementadas com o auxílio de uma biblioteca da IBM conhecida como *aglets* ou utilizando o pacote JESS (Java Expert System Shell) para a plataforma JAVA como máquina de inferência. Como trabalhos relacionados, destacam-se [6][8][9] onde apresentou-se uma metodologia baseada na definição do problema e suas características para o desenvolvimento de agentes de software. Os agentes foram divididos em estáticos e dinâmicos. Sendo que agentes estáticos normalmente não necessitam de técnicas de IA, pois eles apenas executam tarefas baseadas em instruções. Porém, para problemas dinâmicos, comprovou-se em [9] a aplicabilidade de redes neurais recorrentes.

Neste trabalho são descritos o paradigma de utilização de agentes na solução de problemas de redes de computadores, utilizando técnicas de IA. Apresenta-se também uma metodologia de desenvolvimento de uma plataforma de gerência de redes.

O presente trabalho está organizado em cinco seções. A próxima seção apresenta agentes como um paradigma para a resolução de problemas de gerência de redes de computadores. Na Seção 3 são descritos os experimentos realizados para testar a metodologia para desenvolvimento de agentes em problemas mal definidos de gerência de redes. A metodologia proposta para o desenvolvimento de uma

plataforma de gerência é descrita da seção 4. Seguem as considerações finais, trabalhos futuros e referências bibliográficas.

## II. AGENTES NA SOLUÇÃO DE PROBLEMAS DE REDES DE COMPUTADORES

Agentes podem ser considerados como um paradigma para o desenvolvimento de sistemas, sejam simples ou complexos. Segundo Roisenberg [18] agentes autônomos são definidos como sistemas computacionais que operam em ambientes dinâmicos e imprevisíveis. No caso de agentes de hardware, eles interpretam dados obtidos pelos sensores que refletem eventos ocorridos no ambiente e executam comandos em motores que produzem efeitos no ambiente. O grau de autonomia de um agente está relacionado à capacidade de decidir por si só como relacionar os dados dos sensores com os comandos dos motores em seus esforços para atingir objetivos, satisfazer motivações etc. Os agentes de software podem ser encontrados em um número muito maior de aplicações. Podem ou não ser desenvolvidos com o auxílio de técnicas de IA. A maioria dos agentes e gerentes de gerência de redes, normalmente, coletam informações dos objetos gerenciados e repassam aos gerentes. Estes por sua vez realizam operações sobre os agentes instruídos por um administrador de redes humano.

### A. Problemas bem definidos

Se o problema é bem definido então a solução é conhecida: tem-se um conjunto de entradas e um conjunto de saídas bem definido. Ou seja, existem heurísticas que podem ser utilizadas para determinar a solução do problema. Uma forma declarativa de resolver uma função, dando as propriedades que devem ser satisfeitas para solucionar o problema, neste caso, é necessário auxílio de um especialista. O especialista que será um administrador de redes, deverá indicar quais os sintomas, para diagnosticar o problema e sugerir as possíveis soluções. Com essas informações pode-se construir uma base de dados (tipo "troubleshoots") para um sistema de diagnóstico de falhas. Este sistema poderá ter características de IA aplicando-se regras de produção (paradigma simbólico). Outra técnica que pode ser aplicada para implementar a entidade cognitiva de agentes para gerência de redes utilizando heurísticas é através de RNAs diretas.

### B. Problemas mal definidos

Neste caso, são necessários exemplos que poderão ser utilizados para alimentar uma RNA recorrente para aproximar uma solução. É dito que é uma solução aproximada por entender que um problema mal definido não existe uma saída e sim uma saída mais próxima à solução. Isto é definido pelo valor do erro de aproximação da rede neural recorrente. Os exemplos formam uma representação do problema, a sua definição é conhecida para um subconjunto de dados possíveis. A partir daí deve-se analisar este subconjunto e verificar se é um problema estático ou dinâmico. Cada tipo de problema possui um método de solução diferente. Um problema estático pode ser resolvido por RNAs diretas. Porém o dinâmico, só é "bem resolvido" se forem utilizadas

RNAs com características dinâmicas. Deseja-se então, conhecer os elementos do conjunto de respostas admissíveis para todos os elementos do conjunto de dados, mesmo aqueles que não estão incluídos na definição da função. Os exemplos, portanto, são utilizados para treinar uma rede neural com algoritmo adaptativo e obter os valores estimados da solução para os outros valores, utilizando a propriedade de generalização.

Segundo Barreto [3], existem diversas formas de definir uma função, dentre elas destacam-se enumeração exaustiva, declarativamente, por um programa ou através de exemplos. Quando um problema não é completamente definido para todo valor de seus dados, em que se conhece apenas a definição do problema para um subconjunto dos dados possíveis. Neste caso, a solução não é única: todas as funções que forem iguais dentro da região em que o problema é definido são válidas. Neste caso, é melhor ter uma solução aproximada do que os dados para definir a função. Este trabalho defende que resolver o problema será então encontrar um modo de implementar a função ou aproximá-la com as ferramentas disponíveis. Ou seja, treinar a rede através de exemplos e obter valores estimados da solução para os outros valores, utilizando a propriedade de generalização das RNAs.

### III. DESENVOLVIMENTO DE BASELINES PARA A GERÊNCIA PRÓ-ATIVA

No sentido de testar a metodologia para desenvolvimento de agentes de gerência de redes em problemas mal definidos, selecionou-se um problema para o aprendizado através de exemplos por uma rede neural recorrente. O problema escolhido foi o desenvolvimento de um perfil (baseline) de utilização da rede. A utilização de um recurso é medida através da razão entre o tempo em que o recurso é ocupado, realizando o serviço, e o tempo total decorrido em um determinado período. O período em que o recurso não está sendo utilizado é chamado de período ocioso. Balancear a carga do sistema de forma que nenhum recurso seja super ou sub-utilizado é uma das características mais importantes desta métrica, esta atividade é conhecida como planejamento de capacidade. A taxa de utilização é fornecida em porcentagem.

A coleta dos dados foi feita através de consultas a MIB (*Management Information Base*) para obtenção de informações de tráfego. Neste teste foram consultadas as entradas da MIB ifInOctets contendo o valor do contador de tráfego de entrada e ifOutOctets contendo o valor do contador de tráfego de saída, os dados foram coletados de três interfaces de um roteador CISCO AS-2511 da rede da Universidade do Estado de Santa Catarina, especificamente, no Centro de Ciências Agroveterinárias. Foram feitas 288 amostras a cada dia, intercaladas por cinco minutos entre os dias 06/12/02 e 20/12/02. As entradas da MIB que foram consultadas são apresentadas na Tabela 1.

TABELA 1. ENTRADA DA MIB CONSULTADAS NOS TESTE

Identificador (OID)	Descrição	Tipo de Dado (ASN.1)
1.3.6.1.2.1.2.2.1.10	IfInOctets	Counter 32
1.3.6.1.2.1.2.2.1.16	IfOutOctets	Counter 32

As coletas foram realizadas através de um MIB Browser desenvolvido em JAVA. O MIB Browser faz parte da plataforma que está sendo desenvolvida, ou seja, consiste em um dos módulos que a compõe. Salienta-se que este módulo consiste em um agente estático que possui apenas a tarefa de coletar dados através do protocolo SNMP (*Simple Network Management Protocol*) e armazenar as informações em uma base de dados ou repassá-los a outros agentes.

#### A. Rede neural recorrente

Para comprovar o aprendizado através de exemplos foi utilizada uma rede recorrente com a retroação partindo da camada de saída para a camada de entrada. Foram utilizadas funções e objetos pré-definidos para a construção de redes neurais personalizadas da ferramenta MatLab [11]. Como no problema abordado, decidiu-se criar uma baseline da taxa de utilização de uma interface de rede qualquer. E, a taxa de utilização comporta-se como uma função sigmoial (Figura 1) e o tempo, como uma função linear, optou-se por alimentar as RNAs através dos exemplos das taxas de utilização e representar o tempo através das seqüências dos dados.

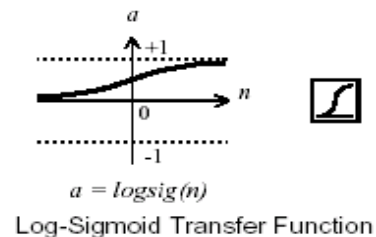


Fig. 1. Função de transferência utilizada

Foram desenvolvidas e testadas diversas RNAs [9], neste trabalho é descrita apenas uma delas. A RNA contém três entradas e três camadas intermediárias. Cada camada intermediária possui 500 neurônios. A camada de saída possui quatro recursões. Uma para ela mesma e outras três que realimentam as camadas intermediárias. Para que a RNA seja uma rede recorrente são necessárias linhas de atrasos TDLS (*Tapped Delay Line*). A Figura 2 ilustra a RNA descrita.

#### B. Experimentos e Resultados

No processo de treinamento da RNA (Figura 2) foram fornecidos conjuntos de 120 amostras de taxas de utilização obtidas no dia 17, 18 e 19 de dezembro. Em seguida, a RNA foi simulada com os mesmos valores de treinamento para provar o aprendizado através de um conjunto de exemplos. A Figura 3 ilustra as saídas das funções treinada e simulada sobrepostas.

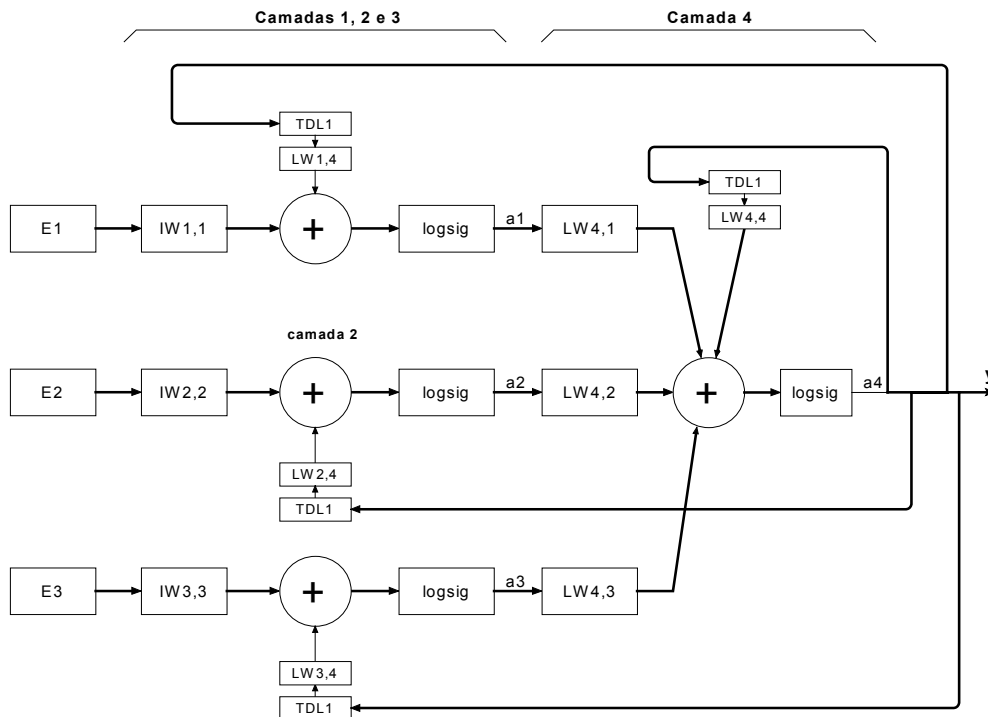


Fig. 2. RNA com três entradas, quatro camadas e quatro recursões

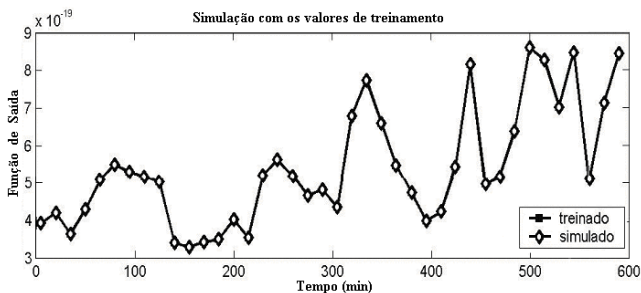


Fig. 3. Treinamento da rede com três entradas

Portanto, através do gráfico comprova-se que a rede foi capaz de aprender o conjunto de exemplos fornecidos. Posteriormente, a rede foi simulada com amostras de dados de dias diferentes do treinamento, observando que houve a generalização do aprendizado. A Figura 4 ilustra o resultado da simulação de um dos dias.

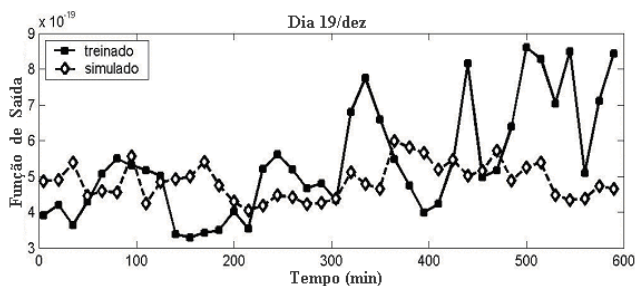


Fig. 4. Simulação com amostras do dia 19/dez

Nesta Figura observa-se que as funções treinada e simulada possuem um comportamento semelhante. Todos os resultados são detalhados em [9]. Desta forma, comprovou-se que quando um problema não é completamente definido para todo valor de seus dados, em que se conhece apenas a definição do problema para um subconjunto dos dados possíveis. A solução não é única, todas as funções que forem iguais dentro da região em que o problema é definido são válidas. Portanto, é melhor ter uma solução aproximada do que os dados para definir a função [3].

#### IV. METODOLOGIA PARA O DESENVOLVIMENTO DA PLATAFORMA DE GERÊNCIA

A implementação dos agentes utilizando a metodologia proposta está sendo desenvolvida através da criação de uma plataforma de gerência em JAVA. O ambiente desta linguagem é naturalmente distribuído e integra de forma natural os aspectos de segurança [13]. Pretende-se ainda, aplicar heurísticas e regras de produção para classificar os valores obtidos nas baselines. Pois, na Figura 4 pode-se observar que a RNA proposta foi capaz de representar o perfil de utilização da interface de rede. Em [18], é descrito o desenvolvimento de baselines para a taxa de utilização empregando técnicas e ferramentas de simulação. Porém, como se trata de um problema dinâmico, a utilização de RNAs é mais adequado. Ou seja, o perfil de utilização da rede varia freqüentemente, conforme a taxa de ocupação e o número de equipamentos em uma rede. No caso de simulação, qualquer alteração na topologia da rede

implicará em uma nova modelagem, e conseqüentemente uma nova simulação para a obtenção dos novos valores. No caso das RNAs recorrentes, não é necessário porque elas são capazes de aprender, implicitamente, a função de transição dos possíveis estados da rede para o conjunto de exemplos fornecidos durante o treinamento.

Nas atividades de gerência de redes, em muitos casos, verifica-se na ocorrência de uma falha, a sua observação e correção somente horas depois da ocorrência. Pois, diversos sistemas de gerência, simplesmente geram alarmes que são disponibilizados para consulta em uma estação de gerência ou através da Web. Este tipo de solução é ineficiente quando os administradores de redes não se encontram no ambiente de trabalho.

O grupo de pesquisa de redes de computadores da UNIPLAC está desenvolvendo uma plataforma de gerência de redes. Pretende-se utilizar a plataforma em empresas de pequeno e médio porte da região de Santa Catarina, demonstrando as vantagens do uso da gerência de redes na qualidade final dos serviços oferecidos.

Neste artigo é apresentada a estrutura da plataforma de gerência de redes. Esta plataforma deverá auxiliar no processo de tomada de decisões de gerência. Futuramente, poderá corrigir as falhas.

Pode-se observar na Figura 5 que a estação de gerência de redes (NMS), além de estar conectada a rede local, podendo armazenar dados e alarmes para consulta em um servidor HTTP, está também conectada a rede telefônica pública de voz através de um MODEM. O que possibilita que os agentes possam contactar os administradores de redes para informar problemas críticos da rede, como por exemplo, a queda do enlace com a Internet. Neste tipo de evento, a conexão da empresa com a Internet fica interrompida, impossibilitando a comunicação da estação de gerência com o ambiente externo. Neste exemplo, entrará em ação os agentes inteligentes. A princípio os agentes podem tentar resolver o problema, como por exemplo, reiniciando as interfaces dos equipamentos e verificando se o estado da rede passará para ativo. Porém, em caso de insucesso, há a possibilidade do agente realizar uma chamada telefônica para o administrador de rede de plantão, executando um arquivo de áudio com a descrição do problema ocorrido. Este requisito foi definido através de observações nas atividades de gerência em universidades e empresas de pequeno e médio porte. Na Figura 6 é apresentada a arquitetura de desenvolvimento da plataforma.

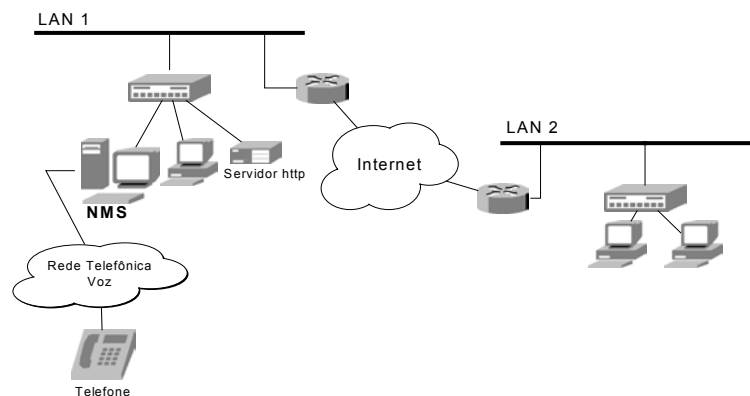


Fig. 5. Estrutura da plataforma

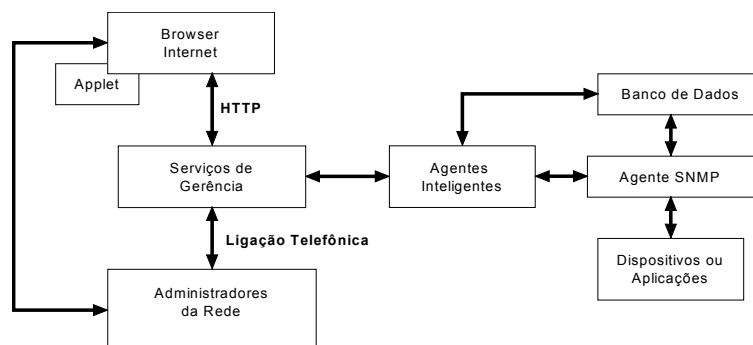


Fig. 6. Metodologia de desenvolvimento da plataforma.

O desenvolvimento da plataforma será constituído de diversos módulos. Pretende-se utilizá-la nas cinco áreas funcionais de gerência: desempenho, contabilização,

segurança, configuração e falhas. O módulo de coleta de dados já está parcialmente desenvolvido. Este módulo consiste em um agente passivo, que apenas coleta dados

através do protocolo SNMP e os repassa para outros agentes ou armazena-os em um banco de dados. As técnicas de IA serão utilizadas no desenvolvimento de agentes inteligentes dentro das cinco áreas de gerência. Um dos agentes que está em desenvolvimento é o responsável pela verificação da taxa de utilização de interfaces de rede. O perfil de utilização será definido periodicamente através do uso de uma RNA recorrente, conforme apresentado da seção 3. Aplicando-se a metodologia para o desenvolvimento de agentes utilizando IA [9], estão sendo desenvolvidos *frameworks* para cada técnica. Desta forma, os agentes responsáveis por classificar os valores de utilização obtidos na *baseline* serão construídos com base em regras de produção ou outro tipo de técnica para solução de problemas estáticos (bem definidos). Os demais *frameworks* serão com base em redes neurais recorrentes, para problemas dinâmicos (mal definidos).

#### V. CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho apresentou uma metodologia para utilização de técnicas de IA que será aplicada no desenvolvimento de agentes em uma plataforma de gerência de redes de computadores. Parte-se da definição do problema, quando o problema a ser solucionado é bem definido então existe um conjunto de entradas e saídas bem definidas. Neste caso, pode-se aplicar técnicas mais simples como regras de produção e RNAs diretas para a construção dos agentes. No entanto, se o problema não for bem definido, conhecendo-se apenas as entradas, ou quando houver uma grande variação no conjunto de saídas. Será possível construir agentes utilizando técnicas capazes de aprender a partir de um conjunto de exemplos, como é o caso das RNAs recorrentes.

Como principais contribuições deste trabalho destacam-se: A metodologia foi aplicada no desenvolvimento de *baselines* para a gerência pró-ativa de redes. No método tradicional de se obter uma função *baseline* é necessário realizar uma atividade de amostragem durante vários períodos de tempo, identificando o desempenho normal da rede através de médias e cálculos estatísticos. Neste trabalho foi demonstrada uma outra maneira de se obter uma função de *baseline* que é através de técnicas de IA. No sentido de contribuir com processo de automação da gerência de redes, apresentou-se experimentos que comprovam a utilização da metodologia no desenvolvimento de agentes na área de gerência de redes empregando técnicas de IA.

Em seguida, como trabalhos futuros poderão ser pesquisadas outras técnicas IA, como é o caso de algoritmos genéticos; que poderão ser integrados como novos *frameworks* para agentes da plataforma de gerência. Pretende-se ainda, desenvolver um pacote em JAVA para a aplicação de técnicas de IA no desenvolvimento de agentes. A plataforma deverá ser utilizada em empresas de pequeno e médio porte, onde foram observadas carências nas atividades de gerência de redes. Devido ao custo elevado das plataformas existentes e em alguns casos, falta

de informação sobre a importância destas atividades dentro da empresa.

#### REFERÊNCIAS

- [1] ALEKSANDER, I.; MORTON, H. *An Introduction to Neural Computing*. Chapman and Hall: 2a. edição, Grã-Bretanha, 1991.
- [2] BARRETO, J.M. *Conexionismo e Resolução de Problemas. Concurso de Professor Titular*, Universidade Federal de Santa Catarina, Departamento de Informática e Estatística, Florianópolis, SC, 1996.
- [3] BARRETO, J.M. *Inteligência Artificial No limiar do Século XXI Abordagem Híbrida Simbólica, Conexionista e Evolucionária*. 2a edição. Florianópolis, 2000.
- [4] BOGONIKOLOS, N.; FRAGOUDIS, D.; LIKOTHANASSIS, S. "ARCHIMEDES: An Intelligent Agent for Adaptive – Personalized Navigation within a WEB Server". In: *Proceedings of the 32nd Hawaii International Conference on System Sciences*, IEEE Press, 1999.
- [5] BRISA. *Gerenciamento de Redes – Uma abordagem de sistemas abertos*. São Paulo: Makron Books do Brasil, 1993.
- [6] COVACI, S.; ZHANG T.; BUSSE, I. "Java-based Intelligent Mobile Agents for Open System Management". In: *Proceedings of the 9th International Conference on Tools with Artificial Intelligence, ICTAI'97*, IEEE Press, 1997.
- [7] DE FRANCESCHI A.S.M., BARRETO J.M., ROISENBERG M. "Desenvolvendo agentes de software para gerência de redes utilizando técnicas de inteligência artificial", In: *II Congresso Brasileiro de Computação – CBComp 2002*, Itajai-SC, Brasil. ISBN 1677-2822, Agosto/2002.
- [8] DE FRANCESCHI, A.S.M. *Desenvolvimento de Agentes Autônomos para Gerência de Redes de Computadores. Exame de Qualificação do Curso de Doutorado* (PPGEEL, UFSC), Florianópolis, 2000.
- [9] DE FRANCESCHI, A.S.M.; BARRETO, J.M.; ROISENBERG, M. *Desenvolvimento de Agentes Autônomos em Gerência de Redes de Computadores*. In: *VIII Simpósio Brasileiro de Telecomunicações, 2000*, Gramado, RS, setembro, 2000.
- [10] DE FRANCESCHI, MORAES, R.A.R.; A.S.M.; BARRETO, J.M.; ROISENBERG, M. *Employing Recurrent Artificial Neural Networks for Developing Baselines for Proactive Network Management*. In: *IATED International Conference Neural Networks and Computational Intelligence - NCI 2003*, 2003, Cancun México, maio, 2003.
- [11] Demuth, H.; Beale, M. *Neural Network Toolbox For Use with MATLAB® - Users Guide Version 4*. (ISBN 0-534-95049-3). USA : The Math Works, 2001.
- [12] HAYES, C.C. Agents in A Nutshell – A very Brief Introduction. *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 1, Jan./Fev., 1999. Pp.127-132.
- [13] Java Dynamic Management Kit. Reference Guide. Sun Microsystems. [http://www.sun.com/software/java\\_dynamic](http://www.sun.com/software/java_dynamic).
- [14] LESSER, V.R. Cooperative Multiagent Systems: A Personal View of the State of the Art. *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 1, Jan./Fev., 1999. Pp.133-142.
- [15] MINSKY, M.L., PAPERT, S. A. *Perceptrons: an introduction to computational geometry*, MIT Press, 1988.
- [16] NISSEN, M.E. An Intelligent Agent for Web-based Process Redesign. In: *Proceedings of the 32nd Hawaii International Conference on System Sciences*, IEEE Press, 1999.
- [17] PINHEIRO, R.; POYLISHER, A.; CALDWELL, H. Mobile Agents Aggregation of Network Management Data. In: *Proceedings of First International Symposium on Agent Systems and Applications – Third International Symposium on Mobile Agents*, 1999.
- [18] ROISENBERG, M. *Emergência da Inteligência em Agentes Autônomos através de Modelos Inspirados na Natureza. Tese de Doutorado* (PPGEEL, UFSC), Florianópolis, 1998.
- [19] SCHWEITZER, C. M. "Desenvolvimento de Baselines com o Uso de Simulação para Automação da Gerência de Redes", Projeto de Conclusão de Curso (INE, UFSC), Florianópolis, SC, 1996.