

Modelo de agente racional para auxiliar na gestão de serviços em redes de computadores

Rational agent model to assist service management of computer networks

José Elias da Silva Justo*
Annabell del Real Tamariz**

A globalização aumentou consideravelmente o volume de dados a serem processados, surgindo a necessidade de interligar computadores em redes. Administrar redes de computadores não é uma tarefa trivial. Agentes inteligentes podem auxiliar na gestão de redes de computadores. O objetivo deste trabalho é desenvolver um protótipo baseado em um agente de *software*, que monitore, gerencie serviços prestados em redes de computadores e intervenha reativamente, reiniciando quaisquer dos serviços com problemas. O agente analisa sua base de conhecimento periodicamente para verificar serviços que foram reiniciados e as possíveis causas dessa ação. O agente mostrou-se eficaz, manteve os serviços em funcionamento e gerou relatórios.

Palavras-chave: Gestão de tecnologia da informação. Redes de computadores. Agente inteligente.

Globalization has greatly increased the volume of data to be processed, resulting in the need of connecting computers in networks. Administration of computer networks is not a trivial task. However, intelligent agents can assist this type of management. The objective of this work is to develop a prototype based on a software agent that monitors, manages services on computer networks, and intervenes reactively, restarting any service problems. The agent analyzes its knowledge base periodically to verify services that were restarted and the possible causes for this. The agent was effective, keeping the services running and generating reports.

Key words: Management of information technology. Computer networks. Intelligent agent.

Introdução

Com a evolução da Tecnologia da Informação (TI), os *hardware* computacionais adquiriram um grande poder de processamento de dados, e os *software* funcionalidades inimagináveis há três décadas, possibilitando a criação e popularização do computador pessoal (PC), a princípio nas médias e pequenas empresas e depois em escritórios e casas. Porém com a globalização e a *internet*, o volume de dados a serem processados cresceu mais rápido que o poder computacional, surgindo a necessidade de interligar os computadores em rede para processar e compartilhar tais dados (KUROSE, 2003).

Uma rede de computadores é um conjunto de computadores autônomos interconectados para compartilhar recursos como *hardware*, *software*, interação e intera-

* Instituto Federal Fluminense - Campos dos Goytacazes/RJ - Brasil

** Universidade Estadual do Norte Fluminense - Campos dos Goytacazes/RJ - Brasil

tividade (SICA, UCHÔA e SIMEONE, 2003), em que há máquinas que assumem os papéis de clientes, servidores e/ou parceiros.

Administrar redes de computadores não é uma tarefa trivial, administrar redes com cerca de mil e quinhentos computadores clientes, trinta servidores dedicados e mais de dois mil usuários é muito complexo, pois exige trabalho conjunto de grandes equipes de administradores de redes (STALLINGS, 2005). Monitorar os serviços dessas redes e agir preventivamente para evitar paradas inesperadas e prejuízos é o anseio das instituições de pesquisa e empresas da área de redes de computadores (NOGUEIRA, 1997); mas com a tecnologia utilizada nos sistemas de monitoramento atuais, apenas mensagens, normalmente *e-mails*, são enviadas para avisar que determinado problema, como por exemplo, o serviço de *proxy* está parado.

Nos últimos anos houve um crescimento das redes e do número de equipamentos, além da diversidade de tecnologias utilizadas nos mesmos, tornando os ambientes de redes muito heterogêneos quanto aos *software* e aos *hardware* utilizados. Juntamente ao crescimento e diversificação dos equipamentos, os custos para montar e principalmente manter uma rede aumentaram significativamente, com pessoal qualificado, serviços especializados, etc. (STALLINGS, 2005).

Redes de computadores são ambientes difíceis de operar, diagnosticar e detectar falhas. Conforme Stallings (2005) e Santos (2004), uma rede de computadores não será bem organizada e gerenciada apenas com a mão de obra humana, é obrigatório o uso de ferramentas automatizadas para auxiliar no monitoramento, controle e gerenciamento da mesma. A partir desse contexto surge a motivação para desenvolver este trabalho.

O objetivo desta pesquisa é desenvolver um protótipo de um sistema de gestão de serviços em servidores de redes de computadores, baseado em um agente inteligente, que monitore serviços de redes e intervenha reativamente, reiniciando quaisquer dos serviços que deveriam estar ativos no servidor e, caso algum desses serviços não seja restabelecido, envie um alerta, por *e-mail*, aos administradores, informando os serviços e os *logs* resultantes da tentativa de iniciá-los, para que a equipe já tenha algumas pistas do que está ocorrendo de errado no servidor. Além dessa atribuição, o sistema deverá analisar sua base de conhecimento, periodicamente, para verificar serviços que tiveram que ser reiniciados constantemente e relatar tal situação, para que a equipe de administradores possa agir proativamente e reconfigurá-los antes que causem alguma parada permanente no sistema.

Revisão Literária

Segundo Russel e Norvig (2004), um agente é qualquer coisa ou ser que é capaz de perceber o ambiente em que está inserido, por meio de sensores e, se necessário, agir sobre ele com seus atuadores.

No estágio atual em que se encontra o desenvolvimento dos modelos de inteligência

aplicados à construção de agentes de *software*, as abordagens mais bem-sucedidas estão orientadas à solução de problemas com contornos muito bem delimitados, tais como os que ocorrem durante a execução de atividades rotineiras, complexas e colaborativas nas áreas de engenharia, gerência e manufatura (KSL, 1999).

Agentes inteligentes são utilizados em redes de computadores com vários propósitos, como no ensino a distância (EaD) (MENEZES, FUKS e GARCIA, 1998) (CUNHA, FUKS e LUCENA, 2002); em sistemas distribuídos (JEUKENS, 2003), (ZANELATTO et al., 2010); em comércio eletrônico (GONZAGA, 2006); realizar filtragem colaborativa de informações (PAZZANI e BILLSUS, 2002); na administração de redes (BIANCHINI et al., 2003); na detecção de intrusos e segurança de redes de computadores (LIMA, 2002).

Bodanese e Cuthbert (2000) projetaram e desenvolveram agentes inteligentes com a finalidade de oferecer uma solução eficiente para alocação de recursos distribuídos em redes de telefonia móvel em situações de uso moderado de alto tráfego, dando maior autonomia às estações base de transmissão.

Em Chantaraskul e Cuthbert (2008), é desenvolvido um agente inteligente que utiliza aprendizagem em raciocínio baseado em casos com o objetivo de garantir qualidade de serviço (QoS) para acesso e utilização de redes 3G de telefonia móvel, fornecendo uma política de uso adequada, de acordo com a taxa de transmissão comprada pelo usuário e garantida com o contrato de SLA (*Service Level Agreement*).

Agentes inteligentes também são aplicados na área de gerência de redes de computadores (HOFFNER, 1993) (HOOD e JI, 1998) (EL-DARIEBY, M. e BIESZCZAD, 1999), com a finalidade de ter o controle e poder agir em função de informações coletadas que mostram determinadas situações em que há a necessidade de intervenção para manter o bom funcionamento da rede, com o intuito de maximizar o desempenho; aprovisionar recursos, mesmo ocorrendo alterações de demanda; minimizar as falhas; documentar, manter as configurações e proporcionar a segurança dos elementos da rede.

De acordo com De Franceschi (2003), a aplicação de técnicas de Inteligência Artificial (IA) na gerência de redes ocorre através da extração de conhecimento da própria rede. Desta maneira, é possível projetar sistemas que se adaptem ao meio, possibilitando que a atividade de gerência não dependa exclusivamente da mão de obra especializada e cara do administrador da rede, minimizando o trabalho humano.

Em De Franceschi et al. (2006), agentes inteligentes e base de conhecimento em redes neurais artificiais são utilizados para fazer gerenciamento inteligente pró-ativo de redes de computadores.

Metodologia

Um agente racional executa ações acertadas, consideradas corretas, boas para

realizar as mudanças necessárias no ambiente, obtendo sucesso. Para definir o que é obter sucesso e então completar o conceito de racionalidade, há o que se chama de medida de desempenho, que mede o sucesso do comportamento do agente para saber se fez tudo certo, averiguando se os estados provocados pelas ações do agente, pelos quais o ambiente passa, é a sequência desejável, assim, então, o agente funcionou bem, obteve sucesso, foi racional (RUSSEL e NORVIG, 2004).

O agente em estudo agirá reativamente. As mudanças ocorrem no ambiente, ele as percebe e então reage adaptando-se ao ambiente ou atuando nele para que retorne às condições ideais de funcionamento (CARBONELL e HOOD, 1986).

Um agente racional deve selecionar uma ação que se espera venha a maximizar sua medida de desempenho para cada sequência de percepções possível, dada a evidência fornecida pela mesma e por um conhecimento interno do agente, caso exista. Tal agente é considerado então como agente inteligente (LUGER, 2004).

A seguir apresentam-se os elementos do projeto do agente: o ambiente onde o agente irá trabalhar, sua função, seus objetivos, suas medidas de desempenho, seus sensores e atuadores e a tabulação da função do agente. Tais descrições representam a metodologia adotada para projetar o agente e as considerações sobre o ambiente em que ele agirá, a rede de computadores.

Ambiente de atuação do agente

O agente atua numa pequena rede de computadores separada para testes e acompanhamento do desenvolvimento da pesquisa, composta de 1 computador servidor e 2 clientes, denominados aqui *cliente1* e *cliente2*. O servidor e o *cliente1* possuem sistema operacional *Linux*, o *cliente2* possui sistema operacional *Windows XP*.

São executados no servidor e serão avaliados os seguintes serviços: *samba*, *proxy* e *web*. O primeiro é um servidor de arquivos, o segundo é serviço de *proxy* e *cache* para acesso a internet e o último, é um serviço de páginas *web*, implementado através do servidor *apache*.

Função e objetivos do agente

O protótipo do agente tem a função de auxiliar na administração dos serviços de um computador servidor numa pequena rede de computadores. Ao ser inserido no ambiente, o agente deverá fazer continuamente uma coleta de dados para descobrir se houve mudanças nele, por exemplo, uma máquina que foi desligada, um serviço que parou de funcionar ou não está funcionando adequadamente, etc.

Três são os passos seguidos pelo agente para descobrir e monitorar o ambiente. No primeiro, ele descobre quantas e quais das duas máquinas clientes da rede estão

ligadas e ativas, sendo que o computador servidor tem que obrigatoriamente estar ligado para que o agente esteja funcionando, pois o mesmo é instanciado e fica rodando no servidor. Esse procedimento será feito ciclicamente a cada cinco minutos, das 8 h às 18 h, apenas em dias úteis, pois alguma máquina poderá ser desligada e/ou ligada, alterando-se a quantidade de computadores disponíveis na rede. Desta forma o agente aprende a configuração atual do ambiente, que é dinâmico, mesmo tendo o pré-conhecimento das variações que o mesmo pode ter. O intervalo de cinco minutos entre os testes foi definido porque em *software* de monitoração de redes, como o MRTG (*Multi Router Traffic Grapher*), o intervalo de tempo padrão para verificações é de cinco minutos (SILVA, 2005).

O segundo passo é mapear os serviços ativos no servidor e conferir, com o seu conhecimento prévio, definido na base de conhecimento, para saber se todos os serviços que deveriam estar ativos estão e se há algum serviço novo em execução.

Na terceira etapa, o agente já conhece o seu ambiente e começa a monitorá-lo por meio dos seus sensores para perceber possíveis mudanças, como por exemplo, serviços que estavam ativos e passam a estar inativos, analisar as causas das mesmas e tentar restaurar as condições ideais de funcionamento do ambiente, por meio dos seus atuadores. Caso, após várias tentativas, não consiga; enviará um alerta, via *e-mail*, para a pessoa que tem a função de administrador da rede, com o relatório do que está acontecendo no ambiente e o que tentou fazer para restabelecê-lo. O administrador analisará o problema, resolverá e informará ao agente, via base de conhecimento, qual foi o problema e como o solucionou, para que ele aprenda.

Medidas de desempenho

Para saber se o agente está trabalhando bem, se está tendo um bom comportamento, é necessário definir suas medidas de desempenho. Neste artigo foram definidas três medidas. A primeira é a latência entre os computadores da LAN, que deve ser inferior a 1 ms (milissegundo). A latência é uma medida que verifica se a largura de banda da rede é, naquele instante, satisfatória, para que o pacote vá de um ponto a outro da rede, sendo devolvido à origem, sem atrasos, sendo possível o tráfego dos pacotes e, ao mesmo tempo, verifica se a rede está congestionada ou não (KUROSE, 2003). Tal valor, 1 ms, foi escolhido após testes realizados com o comando *ping*, na rede local, durante três dias seguidos, num período em que não houve congestionamentos, estando o valor médio da latência abaixo de 1 ms.

A segunda medida é a latência da LAN para a internet, que deve ser inferior a 190 ms. Conforme Valadão et al. (2010), a latência de até 190 ms é considerada muito boa para a maioria das aplicações disponíveis na internet.

A terceira medida é a verificação dos três serviços prestados pelo servidor. Os três devem estar ativos.

A medida de desempenho premia o agente com 1 ponto para cada uma das três medidas apresentadas acima. Para conceder a pontuação o agente verifica se as latências estão dentro dos limites estabelecidos e contabiliza a quantidade de serviços ativos. Esses testes são realizados de duas formas: a primeira é feita apenas em dias úteis, em intervalos cíclicos de cinco minutos, começando às 8 h, que é o horário de início de atividades dos usuários da pequena rede de computadores e encerra-se às 18 h. Nesse caso tem-se 120 medições, sendo que em cada uma pode-se obter 3 pontos no máximo, somando um total de 360 pontos. Caso o agente alcance um percentual igual ou maior a 99%, considera-se, neste trabalho, que o agente obteve bom comportamento/desempenho e foi racional, pois está gerindo bem os serviços.

A segunda forma de testar a medida de desempenho é das 18 h do dia atual até às 8 h do dia subsequente, nos dias úteis e por 24 h consecutivas, a partir das 8 h da manhã nos sábados, domingos e feriados. Os testes também são realizados a cada cinco minutos, mas não averigam a latência entre as máquinas da rede local, pois só o servidor está ligado. Os outros testes são realizados normalmente, sendo possível alcançar o máximo de 240 pontos. O percentual para alcançar um bom comportamento é o mesmo.

Caso o agente alcance a faixa de 90 a 98,9% da pontuação máxima da medida de desempenho, ele estará com um comportamento razoável, sinalizando que atuou de forma aceitável. Abaixo de 90%, o agente está com uma péssima atuação e informará imediatamente ao administrador da rede que é necessária uma análise profunda da sua base de conhecimento, onde estão registrados os testes e os seus resultados.

Nas duas formas de teste, sempre que o agente não alcançar a porcentagem mínima de sucesso, ele enviará relatório para o administrador da rede, contendo todas as estatísticas dos testes e detalhará os momentos em que não alcançou o máximo de pontuação possível.

Se houver falhas nos equipamentos de rede, como *switch*, cabos, roteador, ou falhas no *hardware* do servidor, o agente terá um mau desempenho, mesmo sem ter culpa alguma, porém nesse contexto a pessoa que administra a rede será informada do mau desempenho e pelo relatório perceberá que há falhas em equipamentos, o que poderia passar horas e até dias, dependendo do contexto, como um feriado prolongado, despercebidos, caso o agente não fornecesse as análises do ambiente.

Sensores e atuadores

Os sensores do agente são comandos do sistema operacional *Linux*, que está instalado no servidor; eles monitoram os serviços da máquina servidora e a conectividade dela com as demais máquinas da LAN e algumas da *internet*.

Os atuadores são comandos do sistema operacional que gerenciam os serviços disponíveis no servidor, além dos que reiniciam e desligam o computador.

Estados do ambiente – tabulação da função do agente

Para que o agente fosse desenvolvido neste trabalho, era necessário apresentar quais seriam os estados possíveis do ambiente e as ações correspondentes que o agente teria que executar para que o ambiente estivesse em perfeito funcionamento.

No Quadro 1 estão especificadas as sequências de percepções que o agente tem que identificar com os sensores e as respectivas ações que deve executar com os atuadores. Exemplificando, na segunda linha, o agente percebe que o computador servidor está sem acesso à rede local (LAN). Para reverter tal situação, as ações que ele tem que executar são verificar a configuração da placa de rede da máquina, em seguida reiniciar o serviço de rede e por fim, testar se o acesso à LAN foi restabelecido.

Quadro 1 – Estados do ambiente e as ações do agente

Sequência de Percepções	Ação
[Servidor, Desligado]	-- (agente inoperante)
[Servidor, sem acesso à LAN]	Verificar configuração; Reiniciar serviço de rede; Testar comunicação LAN;
[Servidor, sem acesso à WAN (<i>Wide Area Network</i> , no contexto deste trabalho, internet)]	Verificar configuração; Reiniciar serviço de rede; Testar comunicação LAN; Testar equipamento WAN; Testar comunicação WAN;
[Servidor, Serviço de arquivos samba inativo]	Reiniciar serviço de rede; Iniciar serviço;
[Servidor, Serviço de arquivos samba sem aceitar conexões remotas do <i>cliente1</i>]	Verificar porcentagem de ocupação do disco rígido; Verificar configuração; Reiniciar serviço; Verificar <i>firewall</i> do servidor; Verificar <i>firewall</i> do <i>cliente1</i> ;
[Servidor, Serviço de arquivos samba sem aceitar conexões remotas do <i>cliente2</i>]	Verificar porcentagem de ocupação do disco rígido; Verificar configuração; Reiniciar serviço; Verificar <i>firewall</i> do servidor; Verificar <i>firewall</i> do <i>cliente2</i> ;
[Servidor, Serviço de páginas <i>web</i> (apache) inativo]	Reiniciar serviço de rede; Iniciar serviço;
[Servidor, Serviço de páginas <i>web</i> (apache) sem aceitar conexões remotas do <i>cliente1</i>]	Verificar porcentagem de ocupação do disco rígido; Verificar configuração; Reiniciar serviço; Verificar <i>firewall</i> do servidor; Verificar <i>firewall</i> do <i>cliente1</i> ;
[Servidor, Serviço de páginas <i>web</i> (apache) sem aceitar conexões remotas do <i>cliente2</i>]	Verificar porcentagem de ocupação do disco rígido; Verificar configuração; Reiniciar serviço; Verificar <i>firewall</i> do servidor; Verificar <i>firewall</i> do <i>cliente2</i> ;
[Servidor, Serviço de páginas <i>web</i> (apache) sem aceitar conexões remotas da WAN]	Verificar porcentagem de ocupação do disco rígido; Verificar configuração; Reiniciar serviço; Verificar <i>firewall</i> do servidor; Testar equipamento WAN; Testar comunicação WAN;
[Servidor, Serviço de <i>proxy</i> transparente inativo]	Reiniciar serviço de rede; Iniciar serviço;
[Servidor, Serviço de <i>proxy</i> transparente não permitindo a navegação do <i>cliente1</i>]	Verificar porcentagem de ocupação do disco rígido; Verificar configuração; Reiniciar serviço; Verificar <i>firewall</i> do servidor; Verificar <i>firewall</i> do <i>cliente1</i> ; Testar equipamento WAN; Testar comunicação WAN; Testar serviço DNS (<i>Domain Name System</i>) a partir do <i>cliente1</i> ;
[Servidor, Serviço de <i>proxy</i> transparente não permitindo a navegação do <i>cliente2</i>]	Verificar porcentagem de ocupação do disco rígido; Verificar configuração; Reiniciar serviço; Verificar <i>firewall</i> do servidor; Verificar <i>firewall</i> do <i>cliente1</i> ; Testar equipamento WAN; Testar comunicação WAN; Testar serviço DNS a partir do <i>cliente2</i> ;
[<i>Cliente1</i> , Desligado], [Dias: sábado / domingo / feriado]	NoOP (não realizar ação alguma)

[<i>Cliente1</i> , Desligado (ou com rede inativa)], [Dias úteis]	Enviar <i>e-mail</i> informando o fato ao administrador
[<i>Cliente2</i> , Desligado (ou com rede inativa)], [Dias úteis]	Enviar <i>e-mail</i> informando o fato ao administrador
[<i>Cliente2</i> , Desligado], [Dias: sábado / domingo / feriado]	NoOP (não realizar ação alguma)

Implementação do agente

Nesta etapa, trabalha-se a transformação do modelo de agente, desenvolvido na seção 3, em um protótipo que implementa algumas características e funções da arquitetura proposta.

Funcionamento Básico do Agente

Todas as vezes que o agente for executado, ele funcionará, basicamente, da seguinte maneira, como ilustrado na Figura 1:

1. procura serviços em execução, no servidor e os registra em sua base de conhecimento;
2. inicializa a variável medida de desempenho do agente, atribuindo-lhe o valor zero (0);
3. analisa sua base de conhecimento para descobrir se na última atuação dos sensores havia algum serviço que não estava em execução; caso haja, utiliza seus atuadores para ativá-lo, senão, atribui um ponto à sua medida de desempenho, pois não há serviços inativos;
4. se for dia útil no intervalo entre 8 h e 18 h:
 - a. verifica quais computadores clientes estão ligados e atualiza sua base de conhecimento;
 - b. averigua a latência da LAN. Se ela estiver abaixo de 1 milissegundo, atribui mais um ponto à sua medida de desempenho.
5. testa a latência do servidor para a WAN (*internet*) e, caso esteja abaixo de 190 milissegundos, soma um ponto à sua medida de desempenho;
6. registra a pontuação obtida pela sua medida de desempenho, durante a execução atual, na sua base de conhecimento.

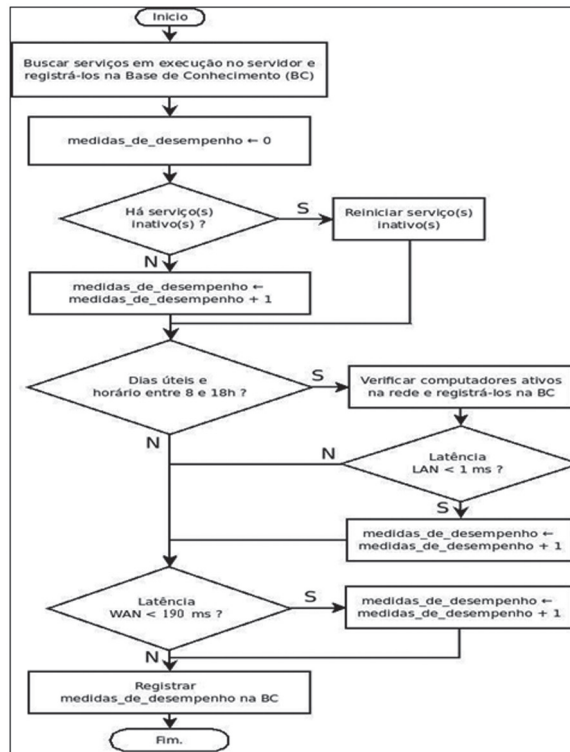


Figura 1 – Base de conhecimento do agente

Base de conhecimento

A base de conhecimento do agente está organizada na forma de cálculo de predicados, implementada em *Prolog* (PALAZZO, 1997), uma linguagem de programação utilizada em projetos de inteligência artificial.

O agente tem um conhecimento prévio do ambiente, chamado de conhecimento inicial, fornecido pelo projetista e será sempre utilizado como ponto de partida, com informações básicas sobre o ambiente, tais como, quantidade de máquinas na rede, neste caso são 3; endereço IP (*Internet Protocol*) de cada computador; qual computador é o servidor, *cliente1* e *cliente2*, o *status* de cada computador, considerando-se inicialmente que o servidor está ligado; a máscara da rede e os serviços que devem estar disponíveis no servidor. Tais informações são ilustradas na Figura 2.

```

quantidade(3,computadores).
computador(servidor,ip(192.168.5.1),status(online)).
computador(cliente1,ip(192.168.5.2),status(offline)).
computador(cliente2,ip(192.168.5.3),status(offline)).
mascara_rede(255.255.255.0).
servico_rede(web,porta(80,tcp)).
servico_rede(samba,porta(139,tcp)).
servico_rede(proxy,porta(3128,tcp)).
  
```

Figura 2 – Base de conhecimento do agente

A partir da base de conhecimento inicial, o agente verifica o ambiente em busca de mais serviços de rede no servidor, verifica se os clientes estão ligados e se há mais computadores na rede. Conforme o projeto, o agente fará verificações a cada cinco minutos para acompanhar os estados do ambiente e tais informações são acrescentadas na base de conhecimento. Também são colocadas na base as informações sobre os problemas ocorridos na rede e o processamento realizado pelo agente para tentar resolvê-los. Essa base é também utilizada para gerar os relatórios sobre o funcionamento dos serviços na rede. A Figura 3 ilustra tais conhecimentos, mostra o estado atual dos serviços, ativo ou inativo, a data das observações do ambiente, o dia da semana, o horário e também há uma cláusula que representa as medidas de desempenho do agente.

```
situacao(ativo(web),data(23,03,2011,quarta),horario(11,23)).
situacao(ativo(samba),data(23,03,2011,quarta),horario(11,23)).
situacao(ativo(proxy),data(23,03,2011,quarta),horario(11,23)).
situacao(ativo(web),data(23,03,2011,quarta),horario(11,28)).
situacao(ativo(samba),data(23,03,2011,quarta),horario(11,28)).
situacao(ativo(proxy),data(23,03,2011,quarta),horario(11,28)).
situacao(ativo(web),data(23,03,2011,quarta),horario(11,33)).
situacao(inativo(samba),data(23,03,2011,quarta),horario(11,33)).
situacao(ativo(proxy),data(23,03,2011,quarta),horario(11,33)).
situacao(ativo(web),data(24,03,2011,quinta),horario(08,42)).
situacao(ativo(samba),data(24,03,2011,quinta),horario(08,42)).
situacao(inativo(proxy),data(24,03,2011,quinta),horario(08,42)).
situacao(inativo(web),data(24,03,2011,quinta),horario(08,47)).
situacao(ativo(samba),data(24,03,2011,quinta),horario(08,47)).
situacao(ativo(proxy),data(24,03,2011,quinta),horario(08,47)).
situacao(inativo(web),data(25,03,2011,sexta),horario(10,53)).
situacao(ativo(samba),data(25,03,2011,sexta),horario(10,53)).
situacao(inativo(proxy),data(25,03,2011,sexta),horario(10,53)).
medidas_desempenho(data(25,03,2011,sexta),pontos(3,dia)).
```

Figura 3 – Base de conhecimento do agente

A partir da atuação do sensor, como por exemplo, na Figura 3, às 10h53min, do dia 25/03/2011, o agente faz uma pergunta à sua base, como ilustrado na Figura 4, para saber se havia algum serviço inativo e, constatado que os serviços *web* e *proxy* estão inativos, o atuador verifica tal situação, como a mais atual da base de conhecimento e tenta inicializar tais serviços.

```
?-situacao(inativo(X),data(25,03,2011,sexta),horario(10,53)),write(X),nl,fail.
```

Figura 4 – Consulta feita pelo *prolog* à base de conhecimento do agente

Sensores

Os sensores do agente foram implementados através de comandos do sistema operacional *Linux*, tais como: *ping*, *netstat*, *telnet* e *traceroute*. Na Figura 5 é mostrado um trecho do pseudocódigo do procedimento que implementa o sensor do agente.

```

Procedimento SensorAgente
1  verificar status serviço web;
2  verificar status serviço samba;
3  verificar status serviço proxy;
4  atualizar base de conhecimento;
fim SensorAgente;

```

Figura 5 – Pseudocódigo do sensor do agente

Atuadores

Os atuadores também foram implementados utilizando-se comandos do sistema operacional, como por exemplo: *reboot*, *halt*, *service <nomeServico> [start/restart]*. O procedimento que implementa o

```

Procedimento AtuadorAgente
1  verificar serviços com status de inativo na base de
   conhecimento;
2  servicosInativos ← vetor de serviços inativos;
3  tamVetor ← comprimento(servicosInativos);
4  i ← 0;
5  enquanto (i <= tamVetor) faça
6     se (inicializar(servicosInativos[i]) = falso) então
7         ArquivoRelatorio ← logs inicialização serviço;
8     fim-se;
9  fim-enquanto;
10 enviar e-mail com conteúdo arquivoRelatorio;
fim AtuadorAgente;

```

Figura 6 – Pseudocódigo do atuador do agente

Os sensores e os atuadores foram implementados em shell script do sistema operacional Linux.

Resultados

O protótipo desenvolvido na pesquisa foi acompanhado durante quatro meses; foram colhidos e mensurados os dados dispostos na sua base de conhecimento.

A Tabela 1 mostra a quantidade mensal de vezes que cada serviço esteve no estado inativo; a quantidade total, por mês, de verificações feitas no ambiente; as porcentagens mensais de inatividade de cada serviço e as totais, para demonstrar o somatório mensal de inatividade.

Ressalta-se que o ambiente real do agente possui uma mixagem de características determinísticas e estocásticas, o que possibilita averiguar que a quantidade de vezes que cada serviço esteve no estado inativo, nos quatro meses de atuação do agente no ambiente, variou sem seguir nenhuma lógica predeterminada, sendo as condições de uso do ambiente, da rede, que determinavam se um serviço estaria sendo mais requisitado do que os outros, o que faz que haja sobrecarga nele, podendo causar a mudança de estado ativo para inativo.

Também é possível visualizar, na Tabela 1, que nos meses de maio e junho, os estados de inatividade dos serviços, comparados aos de março, foram poucos, sendo

o serviço proxy, no mês de maio, o que esteve mais vezes inativo, percentualmente, 0,07% das 26.784 medições. Nesse mesmo mês, todos os três serviços somaram somente 0,17% de inatividade, valor aquém dos percentuais de março e abril, respectivamente 0,91% e 0,25%.

Optou-se por utilizar a soma das porcentagens das inatividades, conforme apresentadas na Tabela 1, para ter-se uma visão mais generalista das falhas que ocorreram mensalmente no ambiente, já que podem ser implantados novos serviços e/ou desativados serviços existentes.

Tabela 1 – Relação de inatividade dos serviços e suas porcentagens

Mês	Serviço	Quantas vezes o serviço esteve inativo no mês	Total de verificações mensais, por serviço	Porcentagem de inatividade	Porcentagem total das inatividades por mês
março	proxy	72	26.784	0,27%	0,91%
	samba	85		0,32%	
	web	86		0,32%	
abril	proxy	23	25.920	0,09%	0,25%
	samba	23		0,09%	
	web	18		0,07%	
maio	proxy	19	26.784	0,07%	0,17%
	samba	14		0,05%	
	web	13		0,05%	
junho	proxy	7	25.920	0,03%	0,11%
	samba	8		0,03%	
	web	13		0,05%	

Avaliando-se o agente por meio das suas próprias medidas de desempenho, percebem-se duas semelhanças com o que foi apresentado na Tabela 1. Uma é o fato da melhora ocorrida no ambiente após o agente ter sido inserido nele, causando uma maior disponibilidade dos serviços para os usuários, mantendo-os na condição de ativos em mais de 99,8% do tempo transcorrido nos meses de maio e junho.

A outra, é que à medida que os aplicativos de rede estão em melhores condições de uso, apresentando maior estabilidade, permanecendo mais tempo como ativos, o agente se autoavalia como tendo bom comportamento. A Tabela 2 demonstra os dados que comprovam tais afirmações, onde se apresentam os valores gerais e as médias mensais das medidas de desempenho averiguadas durante os dias úteis, às noites e fins de semana, para determinar o tipo de comportamento do agente.

As medidas de desempenho foram implementadas no projeto a partir do início do mês de abril, por isso é que não há na tabela valores referentes a março.

No primeiro mês, as medidas de desempenho do agente alcançaram 98,77% da pontuação total, ele se autoavaliou com comportamento razoável. É importante

destacar que a disponibilidade dos serviços, dados mostrados na Tabela 1, é apenas um dos três itens avaliados pelas medidas de desempenho do agente.

Nos meses de maio e junho, o agente avaliou-se como tendo bom comportamento, alcançando mais de 99% do total dos pontos das medidas de desempenho, demonstrando que ele está realmente mantendo o ambiente em ótimas condições de uso para os usuários da rede.

Tabela 2 – Tipo de comportamento do agente

Mês	Média mensal da medida de desempenho (Dia) e %	Média mensal da medida de desempenho (Noite) e %	Média mensal da medida de desempenho (Fins de semana) e %	Médias das porcentagens	Tipo de comportamento do agente
abril	357 – 99,09%	330 – 98,23%	570 – 98,98%	98,77%	Razoável
maio	357 – 99,22%	333 – 99,05%	574 – 99,61%	99,29%	Bom comportamento
junho	357 – 99,24%	333 – 99,23%	574 – 99,59%	99,35%	Bom comportamento

Conclusões

Atualmente as redes de computadores estão em plena expansão, sendo que os equipamentos em uso nelas são cada vez mais heterogêneos. Devido a esses fatos, a monitoração, o controle, enfim a administração das redes está cada vez mais complexa, o que inviabiliza uma boa administração sem a utilização de ferramentas automatizadas e, se possível, dotadas de técnicas de inteligência artificial.

Os benefícios que um sistema inteligente que auxilia no monitoramento de serviços numa rede trazem são de extrema valia para um administrador de redes. Eles reduzem o tempo que é necessário para investigar e corrigir anomalias nos aplicativos de rede e contribuem para a sua estabilidade e desempenho.

Neste trabalho apresentou-se o modelo de um protótipo de agente racional reativo que auxilia na administração de alguns serviços numa rede de computadores. Até o momento, conseguiu-se implementar no protótipo do agente a identificação de serviços inativos, reinicializando-os e fornecendo relatórios das ações realizadas para os administradores.

O agente mostrou-se eficaz dentro dos padrões usados para medir seu desempenho, mantendo os serviços em funcionamento, gerando os relatórios e auxiliando na administração da rede.

O trabalho contribui com a área de administração de redes pelo fato de o agente além de identificar a falha, ou seja, um serviço inativo, tentar reativar o serviço, e no caso de não conseguir, informar ao administrador que o serviço está com anomalias e que não conseguiu ativá-lo, além disso, anexa os *logs* resultantes da tentativa de reativar tal serviço. A maioria das ferramentas automatizadas de administração de redes apenas

informam que houve falha, algumas poucas reagem ao evento, mas não fornecem *feedback* das ações realizadas.

Referências

BIANCHINI, C. B.; ALMEIDA, E. D.; FONTES, D. S.; ANDRADE, R. M. C. Um Padrão para Gerenciamento de Redes. In: CONFERÊNCIA LATINO-AMERICANA EM LINGUAGENS DE PADRÕES PARA PROGRAMAÇÃO, 3., 2003, Porto de Galinhas. *Anais...* Porto de Galinhas, 2003.

BODANESE, E.L.; CUTHBERT, L.G. Application of intelligent agents in channel allocation strategies for mobile networks. In: PROCEEDINGS OF THE ICC 2000, New Orleans, LA, pp. 181–185, 2000.

CHANTARASKUL, S.; CUTHBER, L. An intelligent-agent approach for congestion management in 3G networks. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 21, p. 619-632, 2008.

CARBONELL, J.G.; HOOD, G. The World Modelers Project: learning in a reactive environment. In: *Machine Learning: A Guide to Current Research*. New York: Kluwer Academic Press, 1986. p. 29-34.

CUNHA, L.M.; FUKS, H.; LUCENA, C.J.P. Formação de Grupos no Ambiente AulaNet Utilizando Agentes de Software. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – SBIE, 13., 2002.

DE FRANCESCHI, A.S.M. *Aplicação de Técnicas de Inteligência Artificial no Desenvolvimento de Agentes para Gerência de Redes*. Tese (Doutorado) - Universidade Federal de Santa Catarina. Florianópolis, 2003.

DE FRANCESCHI, A. S. M.; BORGES, K. S. B.; MORAES, R.; VASQUES, F. Artificial Neural Networks Applied to the Development of Agents for Network Management: Simulation Results. In: WSEAS INTERNATIONAL CONFERENCE ON COMMUNICATIONS, 10., July 10-12, 2006, Vouliagmeni, Athens, Greece.

GONZAGA, C. S. M. *Um Agente para Leilões Eletrônicos Simultâneos Heterogêneos com Bens inter-relacionados*. Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação. São Paulo: 2003.

HOFFNER, Y. The Management of Monitoring in Object-based Distributed Systems. In: IFIP/ISINM'93 – INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT, 1993. *Proceedings...* USA: IEEE Press, 1993.

HOOD, C; JI, C. Intelligent Agents for Proactive Fault Detection. *IEEE Internet Computing*, p. 65-72, mar/apr, 1998.

JANSON, S. *Intelligent Software Agents*. [on-line] 1998. Disponível em: <www.sics.se/isl/abc/survey.html> . Acesso em: 20 jul. 2010.

JEUKENS, A. *Tolerância a Falhas em Sistemas de Agentes Móveis*. Dissertação (Mestrado) - Universidade Estadual de Campinas, Instituto de Computação. São Paulo, 2003.

KUROSE, J. F. *Redes de Computadores e a Internet: uma nova abordagem*. São Paulo:

Addison Wesley, 2003.

LIMA, Christiane Ferreira Lemos. *Agentes Inteligentes para Detecção de Intrusos em Redes de Computadores*. Dissertação (Mestrado) - Universidade Federal do Maranhão. São Luiz, 2002.

LUGER, G. F. *Inteligência Artificial: estruturas e estratégias para a solução de problemas complexos*. Tradução Paulo Martins Engel. 4ª ed.. Porto Alegre: Bookmann, 2004.

MENEZES, R.; FUKS, H.; GARCIA, A. C. B. Utilizando Agentes no Suporte à Avaliação Informal no Ambiente de Instrução Baseada na Web – AulaNet. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 9., nov. 1998, Fortaleza.

NOGUEIRA, J.M.S. O Estado Atual da Pesquisa e Desenvolvimento em Gerenciamento de Redes no Brasil. 2º Seminário Franco Brasileiro de Sistemas Informáticos Distribuídos. In: SFBSID'97 – ARQUITETURAS MULTIMÍDIAS PARA AS TELECOMUNICAÇÕES, 1997, Fortaleza, *Anais...*

PALAZZO, L. A. M. *Introdução a Programação Prolog*. Pelotas: UCPEL, 1997.

PAZZANI, M. J.; BILLSUS, D. *Adaptive Web Site Agents*. New York: Kluwer Academic Publishers. Manufactured in The Netherlands. Autonomous Agents and Multi-Agent Systems, 2002. v.5, p. 205-218.

RUSSEL, S. J.; NORVIG, P. *Inteligência Artificial*. Tradução Publicare Consultoria. 2ª edição. Rio de Janeiro: Elsevier, 2004.

SANTOS, F. J. J. *Sistema de Gerenciamento de Redes baseado em Conhecimento*. Monografia (Graduação) - Universidade Federal de Lavras, Lavras, 2004.

SEGAL, R.; KEPHART, J. MailCat: an Intelligent Assistant for Organizing E-mail. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS, 3., 1999. Disponível em: <www.research.ibm.com/swiftfile>. Acesso em: 20 jan. 2011.

SICA, F. C.; UCHÔA, J. Q.; SIMEONE, L. E. *Administração de Redes Linux*. Lavras, MG: Editora UFLA/FAEPE, 2003.

SILVA, J. M. A. *Construção de Agentes SNMP em Ambientes Linux*. Monografia (Graduação) - Universidade Federal de Lavras, Lavras, 2005.

STALLINGS, W. *Redes e Sistemas de Comunicação de Dados*. Tradução da 5ª edição. São Paulo: Editora Elsevier, 2005.

STANFORD KNOWLEDGE SYSTEMS LABORATORY – KSL. *How Things Work Demonstrations*. [on-line] 1999. Disponível em: <www.ksl.stanford.edu/htw/htw-demos.html>. Acesso em: 15 set. 2010.

STARR, B.; ACKERMAN, M.; PAZZANI, M. Do-I-Care: A Collaborative Web Agent. In: ACM CHI'96 CONFERENCE, 1996.

VALADÃO, E.; GEÇARY, A.; SILVA, T.; GUEDES, D.; DUARTE, R. Medição, Análise e Modelagem de Tempos de ida-e-volta na Internet. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 28., 2010, Gramado.

ZANELATTO, A. W.; MARTINS, P. J.; MATTOS, M. C.; SIMÕES, P. W. T. A.
Desenvolvimento de Um Sistema de Apoio a Gerência de Falhas Baseado em Casos –
Antifail. In: SULCOMP, 5., 2010.

Artigo recebido em: 29 abr. 2011

Artigo aceito para publicação em: 8 maio 2012