

# Gestão de Serviços em Redes de Computadores com um Agente Inteligente

**José Elias da Silva Justo**  
**sjusto@gmail.com**  
**IFF e UCAM-Campos**

**Annabell del Real Tamariz**  
**annabell@uenf.br**  
**UENF e UCAM-Campos**

**Resumo:** Este artigo apresenta o projeto de um agente inteligente para auxiliar na administração de serviços de redes de computadores. O objetivo deste trabalho é desenvolver um protótipo baseado em um agente de software, que monitore e gerencie serviços prestados numa rede de computadores e intervenha reativamente, reiniciando quaisquer dos serviços com problemas. O protótipo 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 sofram alguma parada permanente. O protótipo desenvolvido na pesquisa mostrou-se eficiente, melhorando o funcionamento diário numa pequena rede de computadores.

**Palavras Chave:** Gestão - rede de computadores - agente inteligente - -



## 1. 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). 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 interatividade (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 grandes, 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. Monitorar os serviços dessas redes e agir preventivamente para evitar paradas inesperadas e prejuízos é uma necessidade ímpar das instituições; mas com a tecnologia utilizada nos sistemas de monitoramento atuais, apenas mensagens, normalmente *e-mails*, são enviadas para avisar que determinado problema ocorreu, como por exemplo, que o serviço de *proxy* parou.

O objetivo deste trabalho é propor e desenvolver um sistema baseado em um agente inteligente, que monitore serviços de redes de computadores e intervenha reativamente, reiniciando quaisquer dos serviços que deveriam estar ativos no servidor e informe aos administradores sobre falhas constantes neles ou tentativas frustradas de reiniciá-los.

### 1.1 MOTIVAÇÃO

Redes de computadores são ambientes difíceis de operar, diagnosticar e detectar falhas. Conforme Stallings (2005) e Santos (2004) uma rede de computadores não dá para ser bem organizada e gerenciada apenas com a mão de obra humana, sendo obrigatório o uso de ferramentas automatizadas para auxiliar no monitoramento, controle e gerenciamento da mesma. A partir dessa informação e da prática de um dos autores, que faz parte da equipe que administra a rede de computadores de uma instituição pública federal de ensino no estado do Rio de Janeiro, em que quando há paradas de serviços, os administradores descobrem que há algum mau funcionamento somente quando os usuários reclamam que não conseguem realizar alguma tarefa, houve a motivação para a realização deste trabalho.

## 2. TRABALHOS RELACIONADOS

Segundo Russel e Norvig (2004) um agente inteligente 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 o mesmo com seus atuadores. A Figura 1 ilustra um modelo genérico de um agente inteligente.

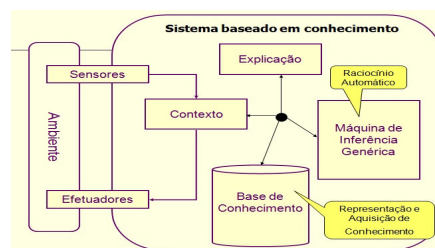


Figura 1: Modelo genérico de um agente inteligente, adaptado de (RUSSEL e NORVIG, 2004).

Um agente inteligente pode ser considerado racional quando executa ações 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).

Neste trabalho propõe-se um agente para trabalhar na área definida como Sistemas e Administração de Redes (BIANCHINI et al, 2003). Esta é uma das áreas de aplicação que usa tecnologia de agente inteligente.

A evolução da computação para cliente-servidor intensificou a complexidade de sistemas que são administrados, especialmente em LANs (*Local Area Network* ou redes locais), e como redes de serviços centralizados estão em plena expansão, essa complexidade aumenta progressivamente. Os usuários, principalmente os operadores e administradores de sistemas, necessitam de uma forma simplificada de gerenciar tais redes.

Arquiteturas de agentes são aplicadas nos sistemas e área de administração de rede há algum tempo, mas estes são, geralmente, apenas agentes móveis, desprovidos de inteligência (LOPES, 2003). Porém, podem ser usados agentes inteligentes para melhorar a performance e eficiência dos *software* de administração de redes de computadores (SANTOS, 2004). Por exemplo, eles podem ajudar a filtrar e a executar ações automáticas a um nível mais alto de abstração e podem até mesmo serem usados para descobrir e reagir a padrões de comportamento de sistema.

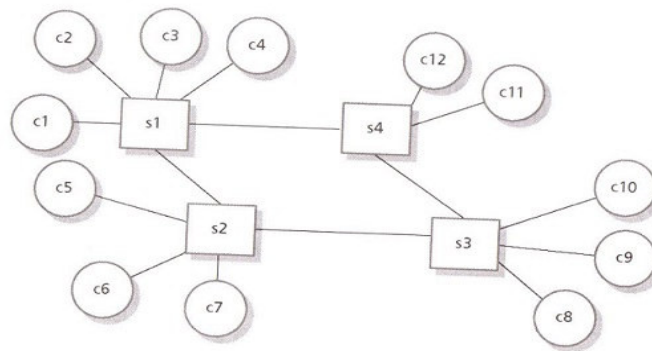
### 3. METODOLOGIA

Nesta seção apresentam-se o escopo do ambiente em que o agente estará inserido, a função e os objetivos dele, os seus sensores e atuadores, o mapeamento dos estados do ambiente, as medidas de desempenho e a forma de representação de conhecimento que é utilizada.

#### 3.1 AMBIENTE

O protótipo do agente projetado está atualmente em uso, para testes e acompanhamento do desenvolvimento da pesquisa, numa pequena rede de computadores, localizada na sala administrativa do Departamento de Tecnologia da Informação da instituição, 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*. A arquitetura dessa rede é cliente-servidor (KUROSE, 2003), de forma que o conjunto de serviços são oferecidos pelo computador que possui a função de servidor e os clientes apenas usufruem desses.

Na Figura 2, há uma representação genérica de redes cliente-servidor, com 4 servidores, representados pelas caixas rotuladas com SI ( $i=1, \dots, 4$ ) e um conjunto de *hosts* clientes rotulados com CI ( $i=1, \dots, 12$ ).



**Figura 2:** Diagrama de uma rede baseada na arquitetura cliente-servidor (KUROSE, 2003).

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

### 3.2 FUNÇÃO DO AGENTE

O agente apresentado neste artigo tem a função de auxiliar na administração dos serviços de um computador servidor numa pequena rede de computadores. A sua principal tarefa é monitorar os serviços de redes e intervir reativamente, reiniciando quaisquer dos serviços inativos. Caso algum deles não seja restabelecido, o agente deve enviar um alerta, via *e-mail* ao administrador, informando o serviço e os *logs* resultantes da tentativa de reiniciá-lo, para que o mesmo 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 o administrador possa agir proativamente, reconfigurando-os antes que causem alguma parada permanente no sistema.

### 3.3 OBJETIVOS DO AGENTE

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. Os passos seguidos pelo agente para descobrir e monitorar o ambiente são: 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.

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

Na terceira e última etapa, o agente já conhece o seu ambiente e começa a monitorá-lo através dos seus sensores para perceber possíveis mudanças, como por exemplo, serviços que estavam ativos e agora, estão inativos e tentar restaurá-los, restabelecendo as condições ideais de funcionamento do ambiente, através dos seus atuadores.

De acordo com De Franceschi, Barreto e Roisenberg (2000), um agente inteligente é autônomo quando consegue se adaptar às mudanças que ocorrem no ambiente. O agente projetado neste trabalho além de racional é autônomo, o que significa que caso o ambiente,

que o agente conhece previamente, conforme informado pelo projetista, seja modificado, como por exemplo, acrescentando-se mais computadores clientes na rede, o agente terá condições de explorá-lo e aprenderá sobre o mesmo, monitorando também as novas máquinas.

### 3.4 SENSORES E ATUADORES

Os sensores do agente são comandos do sistema operacional Linux, como *ping*, *netstat*, *telnet* e *traceroute*, que está instalado no servidor. Eles monitoram os serviços da máquina servidora e a conectividade dela com as demais máquinas da LAN e com algumas da internet. Na Figura 3 é mostrado um trecho do pseudocódigo do procedimento que implementa o sensor do agente para verificar os serviços web e samba, por exemplo.

```

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 3:** Pseudocódigo do sensor do agente.

Os atuadores são comandos do sistema operacional que gerenciam os serviços disponíveis no servidor, além dos que reiniciam e desligam o computador. O procedimento que implementa os efetuadores é ilustrado na Figura 4.

```

Procedimento EfetuadorAgente
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 4:** Pseudocódigo do efetuator do agente.

Os sensores e os efetuadores foram implementados através de um programa implementado em *shell script* do sistema operacional *Linux*.

### 3.5 ESTADOS DO AMBIENTE – TABULAÇÃO DA FUNÇÃO DO AGENTE

Para o desenvolvimento do agente deste trabalho, é necessário mapear quais são os estados possíveis do ambiente e as ações correspondentes que o agente tem que executar para mantê-lo em perfeito funcionamento.

Na Tabela 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.



**Tabela 1:** Estados do ambiente e as ações do agente.

Seqüência de Percepções	Ação
[Servidor, Desligado]	-- (agente inoperante)
[Servidor, sem acesso à WAN (internet)]	Verificar configuração; Reiniciar serviço de rede; Testar comunicação LAN; Testar equipamento WAN; Testar comunicação WAN;
[Servidor, Serviço de arquivos samba sem aceitar conexões remotas do <i>cliente1</i> ou <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> ou <i>cliente2</i> respectivamente;
[Servidor, Serviço de páginas web (apache) inativo]	Reiniciar serviço de rede; Iniciar serviço;
[Servidor, Serviço de páginas web (apache) sem aceitar conexões locais]	Verificar porcentagem de ocupação do disco rígido; Reiniciar serviço de rede; Verificar configuração apache; Reiniciar serviço apache; Verificar <i>firewall</i> do servidor;
[Servidor, Serviço de proxy transparente não permitindo a navegação local]	Verificar porcentagem de ocupação do disco rígido; Reiniciar serviço de rede; Verificar configuração <i>proxy</i> ; Reiniciar serviço <i>proxy</i> ; Verificar <i>firewall</i> do servidor; Testar equipamento WAN; Testar comunicação WAN; Testar serviço dns;
[ <i>Cliente1</i> ou <i>Cliente2</i> , Desligado], [Dias: sábado / domingo / feriado]	NoOP (não realizar ação alguma)
[ <i>Cliente1</i> ou <i>Cliente2</i> , Desligado (ou com rede inativa)], [Dias úteis]	Enviar e-mail informando o fato ao administrador

### 3.6 MEDIDAS DE DESEMPENHO

Para que os projetistas saibam se o agente está sendo racional e útil é necessário definir medidas de desempenho mensuráveis e acompanhá-las durante o funcionamento do *software*.

O agente tem um bom desempenho se a latência entre os computadores da LAN for inferior a 1ms (milissegundo) e a da LAN para a internet for até 300ms. Outra medida adotada é a de estar em estado ativo e pleno funcionamento os três serviços prestados pelo servidor.

A medida de desempenho premia o agente com 1 ponto para cada uma das três formas de avaliação apresentadas acima. Para conceder a pontuação o agente verifica se as latências, durante intervalos de tempo determinados, 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 8h, que é o horário de início de atividades dos usuários da pequena rede de computadores e encerra-se às 18h. 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 95%, considera-se, neste trabalho, que obteve bom comportamento/desempenho e foi racional, pois está monitorando e gerindo bem o ambiente.

A segunda forma de testar as medidas de desempenho é das 18h do dia atual até às 8h do dia subsequente, nos dias úteis. Os testes são realizados, também a cada cinco minutos, mas, não averigua a latência entre as máquinas da rede local, LAN, pois só o servidor está ligado. Os outros testes são realizados normalmente, podendo alcançar a pontuação máxima 336 pontos.

A ultima forma de testar é por 24h consecutivas, a partir das 8h da manhã nos sábados, domingos e feriados. Assim como a segunda, não verifica a latência da LAN, tendo a possibilidade de alcançar 576 pontos.





O percentual para obter um bom comportamento, tanto da segunda como da terceira metodologias de testes, é o mesmo da primeira. Esses dados e os da primeira forma de medir estão dispostos na Tabela 2.

**Tabela 2:** Valores relativos às medidas de desempenho, de acordo com dia da semana e horários

Percentual	Tipo de comportamento	Tempo	Intervalo cíclico	Quantidade de medições	Pontuação máxima
Dias úteis	Das 8h às 18h	10h	5 min	120	360
Dias úteis	Das 18h às 8h	14h	5 min	168	336
Fins de semana e feriados	Das 8h às 7h	24h	5 min	288	576

Caso o agente alcance a faixa de 80 a 94% da pontuação máxima da medida de desempenho, ele estará com um comportamento razoável, sinalizando que atuou de forma aceitável. Mesmo assim ele deverá analisar o seu histórico de ações para averiguar o porquê não alcançou os seus objetivos de manter os serviços disponibilizados pela rede de forma funcional e tentará aprender com a sua atuação mediana.

Abaixo de 80%, o agente está com uma péssima atuação e ele não fará coisa alguma, apenas informará ao administrador da rede que é necessário uma intervenção humana após uma análise profunda da sua base de conhecimento inicial e da base de conhecimento construída durante a sua atuação na rede, onde estão registrados os testes e os resultados dos mesmos. Tais percentuais estão na Tabela 3.

Tabela 3: percentuais de avaliação do comportamento do agente.

Período	Tipo de comportamento
Maior ou igual a 95%	Bom
De 80 a 94 %	Razoável
Menor que 80%	Péssimo

Nas três formas de testar, 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 neste contexto o desempenho abaixo do mínimo esperado é algo bom, pois a administração da rede será informada do mau desempenho e pelo relatório perceberá que há falhas em equipamentos.

### 3.7 REPRESENTAÇÃO DO CONHECIMENTO

Em computação, quando não se aplica recursos e/ou técnicas de Inteligência Artificial (IA) em desenvolvimento de *software*, o sistema projetado é capaz de responder corretamente só àquelas solicitações previamente planejadas. Quando o sistema se depara com algum tipo de dado e/ou situação para o qual não foi preparado previamente, simplesmente não consegue processá-los e dar continuidade às suas rotinas, causando insatisfação aos usuários. A solução para tal problema é que o *software* seja reativo e/ou pró-ativo, podendo compreender melhor as opções de processamento a fim de fazer suas próprias escolhas. Surgem assim as pesquisas sobre agentes baseados em conhecimento, como um ser altamente reativo, capaz de responder às decisões do usuário (RUSSEL e NORVIG, 2004). O componente central deste tipo de agente é sua base de conhecimento, a qual é formada por um conjunto de sentenças expressas numa linguagem de representação de conhecimento, que figura alguma asserção sobre o ambiente ao qual pertence.



A função de qualquer esquema de representação é capturar as características essenciais de um domínio de problema e fazer com que esta informação se torne disponível e entendível para um procedimento que através de inferência tentará resolver o problema.

Segundo Luger (2004) a base da hipótese do sistema simbólico físico é formada por padrões simbólicos que representam aspectos significativos de um domínio de problema. Operações sobre esses padrões geram soluções potenciais, sendo que a seguir, busca-se selecionar a melhor solução e sustenta que a atividade inteligente é alcançada pelo uso destas proposições. A hipótese implicitamente distingue os padrões formados por um arranjo de símbolos e o meio usado para implementá-los. Então, se a inteligência advém apenas da estrutura de um sistema simbólico, qualquer meio que implemente com sucesso os padrões e os processos corretos alcançará a inteligência, independentemente de ele ser formado por neurônios, circuitos lógicos ou blocos de montar.

Tem-se então que definir as estruturas e operações simbólicas necessárias para a resolução inteligente de problemas e desenvolver estratégias para buscar, eficiente e corretamente, as soluções potenciais geradas por estas estruturas e operações; formando o que se conhece por representação do conhecimento e busca.

Neste trabalho usa-se o cálculo de predicado como uma linguagem representacional para a inteligência, pois possui uma semântica formal bem definida, regras de inferência consistentes e completas. A capacidade de inferir novas expressões corretas de um conjunto de asserções verdadeiras é uma característica importante do cálculo de predicados. As novas expressões são corretas na medida em que são consistentes com todas as interpretações prévias do conjunto original de expressões (RUSSEL e NORVIG, 2004).

A base de conhecimento do agente está organizada na forma de cálculo de predicados, implementada em *prolog*, uma linguagem de programação muito utilizada em projetos de inteligência artificial (PALAZZO, 1997). Ela é uma linguagem orientada à programação lógica (RUSSEL e NORVIG, 2004), sendo uma tecnologia que se aproxima bastante do ideal declarativo, onde os sistemas devem ser construídos expressando-se o conhecimento numa linguagem formal e os problemas devem ser resolvidos executando-se processos de inferência sobre esse saber.

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* e os serviços que devem estar disponíveis no servidor.

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 com tais informações, criará uma base de conhecimento dinâmica, diferente da base inicial que é estática. A base atual guarda as informações dos estados do ambiente, dos problemas ocorridos na rede e do processamento realizado pelo agente para tentar resolvê-los. Esta base é também utilizada para gerar os relatórios sobre o funcionamento dos serviços na rede. Na Figura 5 ilustra-se uma sentença atômica, composta de termos complexos, que determina a condição ativa do serviço samba, no dia 09/07/2011 às 2h e 42 min. Também há a informação a respeito do serviço *proxy* estar inoperante.

```
situacao(ativo(samba),data(09,07,2011,sábado),horario(02,42))  
situacao(inativo(proxy),data(09,07,2011,sábado),horario(02,42))
```



**Figura 5:** Registro de conhecimento sobre a situação dos serviços samba e proxy num dado momento.

A partir da informação contida na Figura 5, relativa ao estado inativo do serviço *proxy*, o atuador do agente tentará reiniciar o serviço.

#### 4. RESULTADOS

O protótipo desenvolvido na pesquisa mostrou-se eficiente, melhorando o funcionamento diário da pequena rede de computadores. Para obter tais resultados foram analisados dois cenários: o primeiro antes da existência do agente, em que o funcionamento da rede foi acompanhado por um período de cinco meses e o segundo com a utilização dele auxiliando na administração da rede.

Em ambos os cenários são analisados parâmetros como porcentagens da quantidade de ordens de serviço em que a causa foi o não funcionamento de serviços de redes, tempo médio para resolução dos problemas que geraram as ordens de serviços e o nível de satisfação dos usuários, após a conclusão do serviço solicitado, no momento da finalização da ordem de serviço.

Todos os parâmetros, porcentagens, tempos e níveis de satisfação estão relacionados na Tabela 4. Os níveis e satisfação dos usuários são: muito satisfeito (MS), satisfeito (S), nem satisfeito nem insatisfeito (R), insatisfeito ( I ) e muito insatisfeito (MI).

**Tabela 4:** Descrição dos cenários simulados.

Cenários	Parâmetros		
	Porcentagem de ordens de serviços sobre serviços de redes parados	Tempo médio para a resolução dos problemas	Nível médio de satisfação dos usuários
1 – sem agente	11,90 %	15 minutos	R
2 – com agente	2,14 %	5 minutos	S

Os resultados obtidos deixam claro que, de acordo com o cenário 1, condição anterior da rede de computadores, o tempo médio para a execução completa das ordens de serviços é três vezes maior que no cenário 2, e os usuários não estão nem satisfeitos nem insatisfeitos com as resoluções dos problemas, pois a impressão que eles possuem é que os mesmos problemas estão sempre ocorrendo: as paradas dos serviços da rede. No cenário 2, situação atual da rede, a porcentagem de problemas relacionados a serviços de rede parados é quase seis vezes inferior à do cenário 1 e os usuários estão satisfeitos com as resoluções dos problemas.

Também é importante ressaltar que o tempo de resolução dos problemas é menor porque o agente envia relatórios para o administrador informando os prováveis motivos do não funcionamento dos serviços, logo, no ato da reação a equipe possui conhecimento do que está acontecendo, agindo mais rapidamente e às vezes, proativamente. A quantidade de ordens de serviço diminuiu significativamente, pois o agente inteligente reinicia serviços com problemas automaticamente, de modo que são resolvidos antes que os usuários percebam.

#### 5. CONCLUSÕES

Neste trabalho apresentou-se o modelo de um 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, reiniciando-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 e gerando os relatórios, auxiliando na administração da rede

## **6. REFERÊNCIAS**

**BIANCHINI, C. B.; ALMEIDA, E. D.; FONTES, D. S. & ANDRADE, R. M. C.** Um Padrão para Gerenciamento de Redes. III Conferência Latino-Americana em Linguagens de Padrões para Programação, Porto de Galinhas. 2003.

**DE FRANCESCHI, A.S.M.; BARRETO, J.M. & ROISENBERG, M.** Desenvolvimento de Agentes Autônomos em Gerência de Redes de Computadores. XVIII Simpósio Brasileiro de Telecomunicações, Gramado, 2000.

**KUROSE, J. F.** Redes de Computadores e a Internet: uma nova abordagem. Addison Wesley, São Paulo, 2003.

**LOPES, R. F.** Desenvolvimento de um Agente SNMP Multi-Plataforma, Monografia, Universidade Federal do Maranhão, São Luiz, 2003.

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

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

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

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

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

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