Gerência Pró-ativa de Redes de Computadores usando Agentes e Técnicas de Inteligência Artificial

Marco Antonio da Rocha *

Centro Nacional de Supercomputação e Instituto de Informática
Universidade Federal do Rio Grande do Sul
90035-190 Porto Alegre, RS
E-mail: rock@cesup.ufrgs.br

Luis Fernando Nunes Fernandez †

Centro Nacional de Supercomputação e Instituto de Informática
Universidade Federal do Rio Grande do Sul
90035-190 Porto Alegre, RS
E-mail: islu@cesup.ufrgs.br

Carlos Becker Westphall ‡

UFSC-CTC-INE-LRG Laboratório de Redes e Gerência
Caixa Postal 476 88040-970, Florianópolis, SC
E-mail: westphal@lrg.ufsc.br

Resumo

Este trabalho foi desenvolvido na área de Gerência de Redes de Computadores. O trabalho tem por objetivo estabelecer uma estratégia para a implementação de uma gerência pró-ativa no ambiente de gerência disponível, na rede do Centro Nacional de Supercomputação para gerenciamento de redes associado ao uso de agentes.

O trabalho é motivado pela necessidade de explorar o uso de agentes para levantar os sintomas referentes aos problemas que possam acontecer em redes para a gerência pró-ativa e, em especial, reconhecer um problema utilizando técnicas de Inteligência Artificial e tomar as medidas reativas para solucioná-lo, configurando a aplicação de um gerenciamento pró-ativo na prevenção de problemas referentes a redes de computadores.

*Mestrando do CPGCC (Curso de Pós-Graduação em Ciência da Computação) da UFRGS
†Mestre em Ciência da Computação da UFRGS
‡Professor Dr. Titular do Departamento de Informática da UFSC e Coordenador Geral do PLAGERE/PPROTEM-CC/CNPq.

Abstract

This work describes an effort in order to establish a strategy for the implementation of a proactive management for both the "Intituto de Informatica" and the National Supercomputing Center of UFRGS networks. This paper exploits the use of agents for detecting trouble using A.I symptoms and taking some actions in order to avoid them.

1 Introdução

Devido à constante interação entre os diversos tipos de usuários de redes de computadores é sentida, cada vez mais, a necessidade de organização para melhor gerir os recursos fornecidos. O perfil do usuário é diverso e não permite que se aplique soluções padronizadas, é necessário dividir esforços e aplicar soluções diversificadas que atenda aos interesses de cada grupo de trabalho componente da rede. Gerência de rede é uma aplicação distribuída que envolve as trocas de informações entre processos de gerência, com a finalidade de monitorar e controlar os diversos recursos da rede. Os processos envolvidos em uma associação específica assumem dois papéis possíveis: Gerente e/ou Agente. O *gerente* é a parte da aplicação distribuída que gera operações e recebe notificações. O *agente* é parte da aplicação distribuída que gere os objetos a ele associados (respondendo às operações solicitadas pelo gerente e emitindo notificações que refletem o comportamento dos objetos). Admitindo-se que as ferramentas para gerência de redes não abrangem toda a gama de problemas de uma rede e que essas nem sempre são usadas nas organizações que possuem redes, faz-se necessário que outros mecanismos de gerência sejam utilizados para suprir suas carências mais evidenciadas.

Partindo dessas premissas, sentiu-se a necessidade de cobrir esse subconjunto particular das redes de computadores que não são atingidos pelas ferramentas de gerência atualmente disponíveis e que confere a cada organização as particularidades que diferenciam sua rede de outras, bem como tornar mais fácil ao administrador da rede a tarefa de gerência. Assim, convenciona-se dois comportamentos possíveis em uma rede de computadores: um *comportamento reativo*, o qual avisa o gerente dos problemas ocorridos na rede para que ele providencie sua resolução; e um *comportamento pró-ativo*, em que o gerenciamento deve ser capaz de detectar problemas antes que eles aconteçam a fim de poder evitá-los.

Assim, este artigo visa a apresentar uma estratégia adotada para implementar a gerência pró-ativa de redes de computadores. O trabalho que motivou a elaboração deste artigo foi direcionado para a implementação de um agente e a utilização conjunta de ferramentas de monitoração de rede e comandos do sistema operacional UNIX para a gerência em redes de computadores, sendo utilizados para a validação prática os equipamentos da rede do Instituto de Informática e do Centro Nacional de Supercomputação, bem como o ambiente SunNet Manager. Utilizaram-se workstations Sun, sistema operacional Sun OS 4.1 compatível com o sistema UNIX 4.3 BSD, compilador C do sistema Sun OS, o ambiente de janelas Open Windows 2.1 e microcomputadores ligados à rede do Centro Nacional de Supercomputação.

O artigo encontra-se disposto da seguinte maneira: na seção 2, são abordados os aspectos relevantes à gerência de redes de computadores em geral e são apresentados a gerência pró-ativa e como as técnicas de Inteligência Artificial podem ser utilizadas nesse paradigma; na seção 3, é discutida, detalhadamente, a estratégia de implementação do protótipo mínimo de gerência pró-ativa assim como o conjunto de regras utilizado para

detectar sintomas de problemas na rede; na seção 4, tem-se um comentário dos resultados obtidos; e na seção 5 finalmente tem-se a conclusão do artigo .

2 A Gerência de redes e a Gerência Pró-Ativa

O fluxo das informações em um rede de computadores deve ser confiável e rápido, isso implica que os dados sofram uma monitoração constante, de maneira a filtrar ou mesmo detectar problemas que incidiriam em danos. Uma rede pode existir sem mecanismos de gerência, mas seus usuários poderam enfrentar problemas de congestionamento do tráfego, segurança, roteamento e outros.

A gerência está associada ao controle de atividades e ao monitoramento do uso de recursos da rede. As tarefas básicas da gerência em redes, simplificadamente, são: obter informações da rede, tratar essas informações para possibilitar um diagnóstico e encaminhar as soluções dos problemas. Para cumprir esses objetivos, as *funções de gerência* devem ser embutidas nos diversos componentes de uma rede, possibilitando descobrir, prever e reagir a problemas [WES 88].

Para resolver os problemas associados à gerência em redes, a ISO, através do OSI/NM propôs três modelos [WES 91]:

- o Modelo Organizacional, o qual estabelece a hierarquia entre sistemas de gerência em um domínio de gerência, dividindo o ambiente a ser gerenciado em vários domínios.
- o Modelo Informacional, que define os objetos de gerência, as relações e as operações sobre esses objetos. Uma MIB é necessária para armazenar os objetos gerenciados.
- o Modelo Funcional, o qual descreve as funcionalidades de gerência - gerência de falhas, gerência de configuração, gerência de performance, gerência de contabilidade e gerência de segurança.

Assim, a idéia de gerência de uma rede de computadores provê ao administrador da rede as facilidades suficientes para melhor distribuir seus recursos aos seus usuários, mas, devido à quantidade de informações disponíveis, abre-se espaço à gerência pró-ativa.

2.1 A Gerência Pró-Ativa

Tal como citado anteriormente, a idéia de gerenciamento pró-ativo é antecipar possíveis problemas que possam ocorrer em uma rede de computadores para detectá-los antes que eles aconteçam e não apenas avisar quando do seu aparecimento. É fundamental observar um comportamento anormal da rede, coletar seus sintomas e diagnosticar corretamente um problema maior que possa vir a ocorrer ou cadastrar anomalias quando não for possível reunir indícios suficientes para ligar o acontecimento a um problema conhecido. É necessário, também, manter a rede sob constante observação, para que a partir disso, possam colher dados suficientes a fim de selecionar o que poderia ser sintoma e relacionar com um problema conhecido.

Em uma situação ideal, as ferramentas de gerenciamento de uma LAN estabeleceram um framework em que devices como smart hubs monitorarão a atividade da rede e colocarão informação a plataformas de gerênciamento distribuído que, automaticamente, gerarão trouble tickets, custos de operação e relatórios de utilização [JAN 93].

Essas mesmas plataformas deverão providenciar o atendimento através de uma aplicação de gerência com capacidade de configuração e planejamento. No Gerenciamento Pró-ativo, é necessário o concurso de várias ferramentas de gerência para que se estabeleça, no cômputo dos dados, uma relação com a causa a ser inspecionada. Na maioria destas ferramentas trabalha-se com limiares (thresholds) o qual delimitam os limites em que começa a atuar reportando eventos como número de erros, tipos específicos de pacotes e outros parâmetros sobre seleção de intervalos. É importante também a adição de bancos de dados relacionais que torna o uso versátil e facilita o acesso aos dados de gerência, possibilitando a integração com outras ferramentas.

Através de uma medição da atividade normal sobre um tempo determinado e uma identificação de performance baseada em cálculos estatísticos, é possível estabelecer um repositório de dados com os parâmetros normais de funcionamento da rede o qual chamamos baseline. Essa baseline pode ser usada em Gerência Pró-Ativa por um conjunto de funções para estabelecer uma estatística válida da caracterização do comportamento normal da rede sobre um novo período de tempo por um específico intervalo, contabilizando os níveis de tráfego em horas diferentes e em diferentes dias da semana. Desse modo tem-se caracterizado o perfil de cada tipo de rede, e os gerentes de rede podem colocar limiares (Thresholds) significativos para alertar um comportamento anormal.

2.2 A Inteligência Artificial no Gerenciamento Pró-ativo

O gerenciamento de redes, como já se pode ver anteriormente, é uma tarefa complexa, na qual encontra suporte em outras áreas de aplicações, dentre as quais a inteligencia artificial e os sistemas especialistas têm particular destaque. Os principais produtos de gerênciamento já a empregam facilitando as tarefas de gerenciamento. A inteligência artificial pode ser usada para antecipar problemas que deixariam a rede inoperante. Assim, abordar-se-a como a inteligência artificial pode contribuir para um gerenciamento pró-ativo.

A monitoração dos sistemas pode ser utilizada para projetar a performance da rede, comparando-se os dados retirados com uma baseline com vistas à escolha correta de ações corretivas. As tarefas de interpretação e diagnóstico de um mau funcionamento da rede têm seu ponto forte no sistema especialista, que pode usar inferência sobre os dados coletados. Projeto e planejamento de instalações novas podem ser feitos com uma boa base de conhecimento instalada.

O uso de sistemas especialistas,atualmente, tem aumentado principalmente nas áreas que possuem tarefas complexas nas quais não são muitos os especialistas. As redes de computadores enquadram-se neste contexto por sua grande difusão e utilização crescente a cada dia. Os serviços oferecidos por uma rede se tornam indispensáveis; muitas pessoas dependem deles de maneira que o investimento em sua gerência se torna viável. Em uma rápida avaliação custo-benefício podemos citar as seguintes vantagens:

- melhor qualidade de serviços, com a disseminação do especialista por todos os segmentos da rede fica facilitada a tarefa do administrador resultando uma melhor performance;
- maior agilidade, menor custo e maior produtividade na execução dos serviços proporcionada pelo tratamento automático;
- maior confiabilidade, com tomada de decisões mais rápida;

- melhor preparo dos recursos humanos com suporte ao treinamento.

Um sistema especialista divide-se em quatro fases distintas:

- aquisição de conhecimento;
- base de conhecimento;
- máquina de inferência;
- Interface explanatória.

A aquisição de conhecimento é a fase de extração e formalização do conhecimento de um perito para seu uso em um sistema especialista. Nesse processo, trabalham os "engenheiros do conhecimento", técnicos especializados na tarefa de auxiliar os peritos a colocarem seus conhecimentos no sistema especialista através de regras práticas e estruturação do conhecimento. Quando o perito manifesta seu conhecimento empiricamente, o engenheiro de conhecimento representa isso por regras heurísticas que, quando codificadas, dirigem o processamento através da massa de informações, tornando o processo mais eficiente. Assim, obter essas regras é um passo importante na aquisição do conhecimento.

A base de conhecimentos armazena o conhecimento do perito e diferencia-se de uma base de dados convencional por ter um comportamento ativo passível de atualizações conforme o contexto. A estrutura da base de conhecimentos dependerá do tipo de conhecimento representado. Para um conhecimento dedutivo , a base será, normalmente, composta por regras, caso se tivermos modelamento de estruturas físicas, ligações causais ou interrelacionamento entre modelos, a estrutura mais adequada pode ser a rede semântica.

A máquina de inferência tem por função selecionar e aplicar a regra apropriada a cada passo do sistema especialista, manipulando a base de conhecimento. A máquina de inferência pode partir de premissas ou peças elementares de informação e tenta conseguir seu objetivo através da combinação delas. Nesse caso, diz-se que realiza caminhamento para a frente, ou pode partir de um objetivo e verificar as premissas necessárias dos fatos envolvidos chegando a uma conclusão, assim neste diz-se que realiza caminhamento para trás. As máquinas de inferência que trabalham com uma mistura dos dois processos são as que conseguem mais sucesso, pois, em muitos casos, a escolha do processo de inferência é feito reproduzindo a maneira que uma pessoa utilizaria para resolver um problema.

Em um sistema especialista, o conhecimento do domínio do problema é organizado separadamente dos outros conhecimentos do sistema, assim como os procedimentos ou passos para resolução do problema, ou da interação com o usuário representada pela interface explanatória, que define como apresentar os conhecimentos. Essa divisão é propósital uma vez que esses sistemas se dividem em "base de conhecimento" que é o repositório do conhecimento especializado e "máquina de inferência", que reúne os procedimentos de resolução ou passos para solução do problema. O conjunto forma o que se chama de "sistema baseado em conhecimento", a base de conhecimento contém fatos e regras, e a máquina de inferência decide como aplicar essas regras e em que ordem, a fim de deduzir novos conhecimentos. Uma vez que o conhecimento especializado está separado, fica mais fácil ao projetista manipular procedimentos [ROC 94].

3 Implementação do Protótipo Mínimo de Gerência Pró-Ativa

Relatar-se-a agora, a implementação de um protótipo mínimo de gerência pró-ativa para a validação prática do trabalho. Considerando o que foi dito ao longo da trabalho, o objetivo da gerência pró-ativa é desenvolver um comportamento pró-ativo em uma das funcionalidades do modelo funcional de gerência, sendo, no presente trabalho, abordada a gerência de performance, de modo que se deseja previnir os sintomas de queda de performance na rede e, com isso, antecipar-se aos acontecimentos que possam vir a congestioná-la ou que resultem maiores danos a rede, não apenas avisando do ocorrido, mas tomando providências no sentido de evitar que os problemas se tornem maiores e, muitas vezes, críticos.

Para atingir esse objetivo, foi idealizado um processo o qual se mantêm sempre ativo monitorando, periodicamente, sistemas remotos e analisando as respostas e mais outras informações colhidas do sistema e de uma baseline, no qual um gerente recebe gets e traps de agentes e faz a parte de diagnóstico dos sintomas de problemas da rede, tomando ações de acordo com os níveis de thresholds ou apenas comunicando o fato ocorrido, o que configura o comportamento pró-ativo [ROC 94].

3.1 Gerência Pró-ativa abordando a Análise de Performance da Máquina Gateway da Rede

No desenvolvimento deste artigo, será apresentado o detalhamento individual do fluxo de trabalho até agora realizado, mostrando a estratégia adotada para a sua elaboração, mas abstraindo detalhes particulares como a interface com o SNM e o método de desenvolvimento de um agente para o ambiente de gerência [SUN 89].

A tarefa de monitoração e observação da rede, por suas características, é melhor realizada por agentes com ações residentes nas máquinas a serem gerenciadas. Em vista disto, foi criado o agente 6 que tem por objetivo medir o congestionamento de um barramento Ethernet para ter uma medida de qualidade e desempenho dos serviços de comunicação da rede por uma monitoração do volume do tráfego e a verificação da quantidade dos erros. Essas estatísticas serão usadas ,posteriormente, como base de dados para estabelecimento de uma baseline.

De posse dessas análises iniciais, determinou-se o seguinte fluxo de trabalho:

- rodar o agente6, Hostmem, Hostif e Hostperf em alguns nós das subredes, medindo congestionamento e outras estatísticas em horários diferentes;
- com base nos resultados, estabelecer uma "baseline" ou base com pontos conhecidos como média e desvio padrão das medidas em situações normais;
- implantar o módulo diagnóstico, de modo que se acione regras, cada vez que a medida lida esteja fora dos parâmetros padrões contidos na "baseline" estipulado como:

    1. para cada medida lida dos agentes na monitoração que chegar ao alcance da média da hora em que ocorreu, será disparado o módulo diagnóstico, que passa a verificar se existe problemas de acordo com as regras do módulo;

2. se o módulo diagnóstico verificar que se trata de um problema de queda de performance ou congestionamento, serão usadas regras para estabelecer quais os motivos que causam o evento;
3. em um terceiro momento, após verificados os anteriores,tomam-se atitudes para evitar o problema reportando ao administrador da rede, via e-mail ou pela interface, as anomalias encontradas sugerindo procedimentos corretivos.

3.2 Metodologia do Experimento e Estratégia Utilizadas

Neste experimento, foi utilizada a rede local do CESUP (Centro Nacional de Supercomputação), a qual apresenta uma rede local bastante heterogênea. No CESUP são encontrados microcomputadores IBM-PC compatíveis, microcomputadores Apple Macintosh, estações de trabalho Silicon Graphics, Suns e o supercomputador CRAY YMP2E, mais especificamente, a subrede de estações de trabalho SUN com a topologia mostrada na figura abaixo.

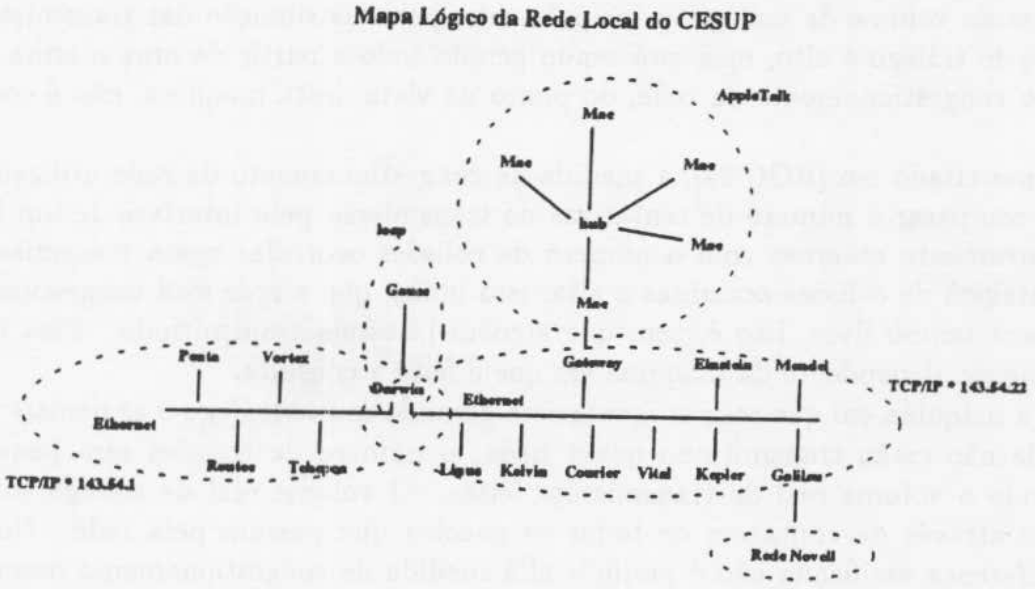

Figure 1: A rede local do CESUP

Os microcomputadores Macintosh comunicam-se utilizando o protocolo AppleTalk. A ligação Física desses dá-se através de um hub, que caracteriza a topologia de estrela para esta sub-rede. A ligação da sub-rede AppleTalk com a rede local do CESUP é feita através de um gateway.

A sub-rede 143.54.22 é um segmento Ethernet e nela encontram-se as estações Silicon Graphics e Suns. Outro segmento Ethernet encontrado é o que compõe a sub-rede 143.54.1. este é o backbone da UFRGS, onde se encontram o roteador Tchepoa (porta de entrada à rede TCHE), o Vortex (servidor de nomes da rede da Universidade), e o Routcc (roteador para o Campus do Centro). O acesso remoto ao CESUP, para os usuários de

outras Instituições, é feito através do Tchepoa e pelo Vortex, o acesso dos usuários da UFRGS é pelo Routcc.

Unindo essas duas sub-redes, encontra-se a estação Darwin, a qual representa um papel-chave na rede local do CESUP. Essa máquina é o unico nodo de acesso ao super-computador Gauss (CRAY YMP2E), tanto para usuários locais como para os remotos. A ligação a Darwin e o Gauss é feita através de um protocolo proprietário da CRAY. A Darwin tem, também, a função de servidora da rede local, serviço de arquivos, correio eletrônico, nomes (DNS) e páginas amarelas (NIS).

3.3 O agente 6

Para a validação prática deste comportamento nesta experiência escolheu-se, também, o problema da performance e congestionamento, tendo em vista que os resultados apurados servirão como um feedback para outros estudos. Objetiva-se encontrar uma região média de queda de performance de uma estação de trabalho e quando for atingido níveis próximos a este, enviar uma mensagem ao administrador da rede informando do início da queda de performance e do congestionamento.

Chama-se congestionamento quando muitos pacotes estão presentes em um trecho de sub-rede e ocasionam a degradação da performance, ou seja, quando o número de pacotes colocados não é proporcional ao número de pacotes distribuídos. O congestionamento difere-se do volume de tráfego, principalmente, pela distribuição das transmissões. Se o volume de tráfego é alto, mas está sendo gerado todo a partir de uma mesma máquina, então o congestionamento da rede, do ponto de vista desta máquina, não é considerado alto.

Como citado em [ROC 94], a medida de congestionamento da rede utilizada baseia-se em comparar o número de tentativas de transmissão pela interface de um host para um barramento ethernet com o número de colisões ocorridas nessa transmissão. Se a porcentagem de colisões ocorridas é alta, isto indica que a rede está congestionada, pois há pouco tempo livre, isto é, sem que alguém já esteja transmitindo. Essa medida é, obviamente, dependente da máquina em que é feita a consulta.

Se a máquina em que roda o agente está gerando muito tráfego e as demais máquinas da rede não estão transmitindo quase nada, o número de colisões será pequeno, não refletindo o volume real de transmissões feitas. O volume real de tráfego só pode ser medido através da contagem de todos os pacotes que passam pela rede. No entanto, essa diferença verificada não é prejudicial à medida de congestionamento desejada, pois se a estação em que roda o agente é a que está transmitindo demais, para ela não há congestionamento. E para as demais estações só haverá congestionamento na medida em que elas tentarem transmitir mais, o que também será refletido na estação do agente, com o aumento do número de colisões. Para obter a porcentagem de colisões, calcula-se a relação número de colisões e o número de tentativas de transmissão ocorridas no intervalo entre o tempo atual e a última consulta [ROC 94].

O agente 6 foi instalado na estação Darwin, que é a servidora da rede local e gateway do CESUP, sendo monitoradas as interfaces ie0 - bus da rede da UFRGS e ie1 - bus da rede local do CESUP. O agente 6 operou em dois modos básicos suportados pelo SNM: monitoração de dados e monitoração de eventos.

Pela monitoração de dados, o agente fica em execução contínua, fazendo uma consulta a cada intervalo de tempo. Ao final de cada consulta, os dados recuperados são enviados ao

gerente (SNM). Esse envio periódico de dados permite a geração de gráficos por parte do SNM e também o armazenamento dos dados recuperados em consultas anteriores no próprio agente, que então pode gerar estatísticas sobre o andamento do sistema de comunicações.

A monitoração de eventos funciona basicamente como a monitoração de dados, porém não tem a finalidade de mostrar ao usuário através de números ou gráficos todos os dados recuperados, mas somente determinados eventos sobre esses dados. Esses eventos são gerados basicamente quando um dado monitorado atinge ou ultrapassa determinado valor ("threshold"). Dessa forma, é possível controlar se um dado recuperado está dentro de uma gama de valores tidos como aceitáveis. Caso esse dado saia fora dessa faixa, o evento informa o administrador, de algum modo, para que ele possa verificar a situação e tomar a medida necessária.

Uma vez que o SNM suporta a geração automática de eventos como "send mail" e permite a especificação de vários tipos de "thresholds", a implementação dessa mmaneira de funcionamento no agente 6 não trouxe nenhum tipo de ônus ao seu desenvolvimento, pelo contrário, foi utilizada essa facilidade para enviar o alerta ao administrador da rede.

A simples monitoração de dados a cada intervalo pequeno de tempo não permite concluir facilmente se a rede está seriamente congestionada ou não, pois rajadas de transmissão acontecem normalmente. Como exemplo, caso se puder obter a medida proposta a cada tentativa de transmissão, os dados recuperados seriam sempre 0 ou 100% de congestionamento, o que seria difícil de ser interpretado por alguém. Essa situação é agravada no modo de monitoração de eventos, pois não interessa de forma alguma a um administrador receber um evento informando que a taxa de colisões foi, por exemplo, de 30% no último segundo. Por outro lado, caso se obtiver uma medida de congestionamento durante todo o período de vida do sistema, essa média seria bastante baixa e não iria refletir bem os problemas ocorridos durante alguns períodos.

Para permitir um controle mais inteligente da situação de congestionamento da rede, é necessário que o agente informe, então, uma média desse dado nos últimos N intervalos de tempo. A escolha desse número de intervalo N deve ser feita com o objetivo de modelar bem a situação do sistema com relação ao tempo real e de acordo com as necessidades de administração.

A partir dessa média é possível então se ter uma visão mais ampla e realmente gerar eventos úteis, informando que o congestionamento da rede nos últimos tempos está crescendo e ultrapassando os limites impostos para um funcionamento correto.

O agente implementado possui uma única tabela chamada "output". Nessa tabela, são recuperados todos os parâmetros correspondentes à tabela de mesmo nome do agente etherif e mais três atributos específicos do agente 6. Estes atributos são "opercol" – percentual de colisões a cada intervalo básico; "oaveper" – média do percentual nos últimos 60 intervalos; "oaveper2" – média do percentual nos últimos 900 intervalos.

O cálculo da média é atualizado a cada intervalo. Esse cálculo não consome muito tempo de CPU porque o algoritmo utilizado mantém na memória do agente a soma dos últimos 900 e dos últimos 60 intervalos, bastando subtrair-se o mais antigo, somar-se o mais recente e efetuar duas divisões (por 900 e por 60) para se obter as novas médias.

O cálculo da média a cada intervalo (a cada consulta) é feito dividindo-se a diferença de colisões pelas tentativas de transmissão no período. Esse número de tentativas de transmissão é calculado pela soma da diferença de tentativas de RE-transmissão com a diferença de pacotes transmitidos com sucesso.

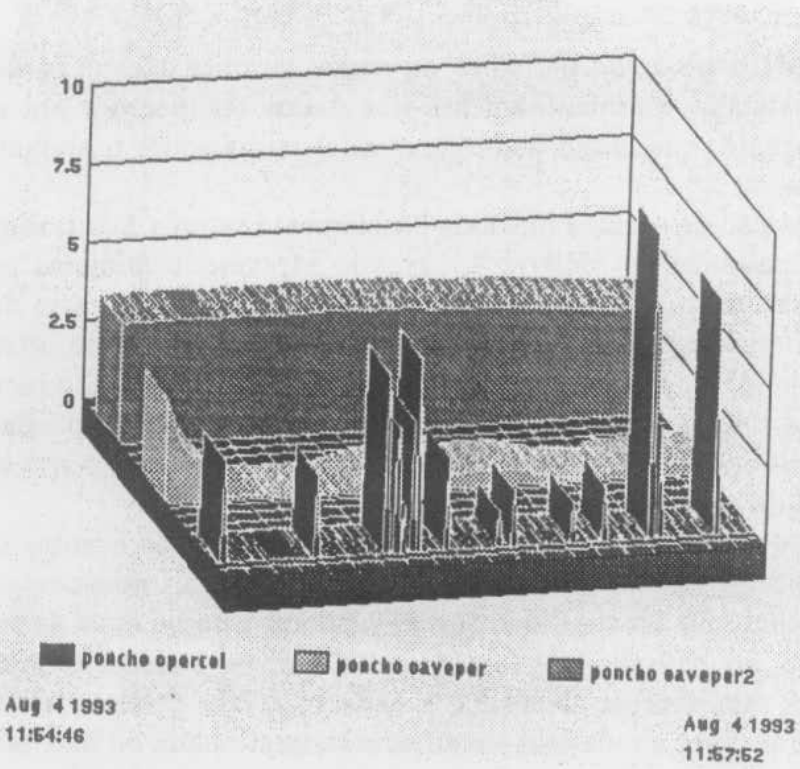

Figure 2: Gráfico com os atributos do agente 6 em uma situação real

Como já se havia observado em experiências passadas, a medida de congestionamento nos intervalos básicos (tempo em segundos colocado na interface do SNM) é muito inconstante, sendo composta de seguidas rajadas refletindo o congestionamento quando alguma aplicação necessita grandes transferências de dados.

Na média calculada com os últimos 60 intervalos básicos, sucessivos picos elevados se refletem como um aumento, razoavelmente grande, no congestionamento das transmissões, indicando que a situação está crítica naquele período.

A média calculada a cada 900 intervalos básicos, no entando, é pouco sensível a alterações causadas por uma ou outra aplicação que necessita grandes transferências. Pelo contrário, sua grande estabilidade reflete mais o estado de utilização da rede em geral, por todos os usuários durante um período mais longo, a partir de 15 minutos.

3.4 A Montagem da Baseline

Esta estação foi monitorada, constantemente, pelo agente 6 por períodos de 24 horas, realizando-se medidas de 10 em 10 e de 15 em 15 minutos, buscando-se o melhor intervalo de monitoração no tempo de uma hora, totalizando de 4 a 6 medidas em uma hora, implicando 96 e 144 durante as 24 horas do dia. Após as medidas, foram feitas avaliações relativas ao andamento e perfil de tráfego medido nesse período e selecionados, respectivamente, os objetos de hostperf, hostmem, hostif e agente 6 que melhor contribuíam para o estabelecimento dos níveis padrões de comportamento, quais sejam:

- hostperf;
- cpu - percentagem de utilização da CPU;

- ipkts - número de pacotes de entrada;
- opkts - número de pacotes de saída;
- ierrs - número de erros de entrada;
- oerrs - número de erros de saída;
- colls - número de colisões;
- hostmem
- mbuf - percentagem de buffers usados;
- memused - número de bytes usados para a rede;
- memfree - número de bytes livres para rede;
- mem - percentagem de bytes usados para rede;
- hostif
- ipkts, opkts, ierrs, oerrs e colls- Mesmo significado para interface ie0,ie1;
- agente 6
- oaveper, opercol, oaveper2 - já citados na seção anterior;
- odrops - número cumulativo de pacotes na fila de saída;
- obuff - número corrente de buffers transmitidos

Cada agente colhe dados da máquina gateway e geram arquivos, de formato ASCII, contendo os valores encontrados em cada um dos parâmetros que eles devem medir. Os objetos selecionados são extraídos desses arquivos através de um programa "parser" específico para cada agente, escrito em linguagem C, que os acumula em outro arquivo intermediário, calculando a média e o desvio-padrão dos valores do objeto para cada hora. Esse arquivo intermediário é passado por outro programa "parser" geral, o qual calcula, também, a média e o desvio-padrão dos valores de cada objeto, reunindo cada dia de monitoração em seu respectivo dia da semana, obtendo, assim, a média e desvio-padrão para cada hora de cada dia da semana, de todo o período monitorado. De posse dessa baseline, tem-se o comportamento médio padrão da rede em cada hora do dia e em cada dia da semana.

3.5 O Módulo Diagnóstico

O módulo diagnóstico foi elaborado com todas as características de um sistema especialista, sendo utilizadas as quatro fases desse tipo de sistema, as quais se passará a apresentar agora.

Para o processo de aquisição de conhecimento, foram utilizadas as metodologias propostas por [LEA 95] de construção de árvores de conhecimento. De forma complementar, foi realizada uma revisão bibliográfica do material disponível e, logo após, foram realizadas

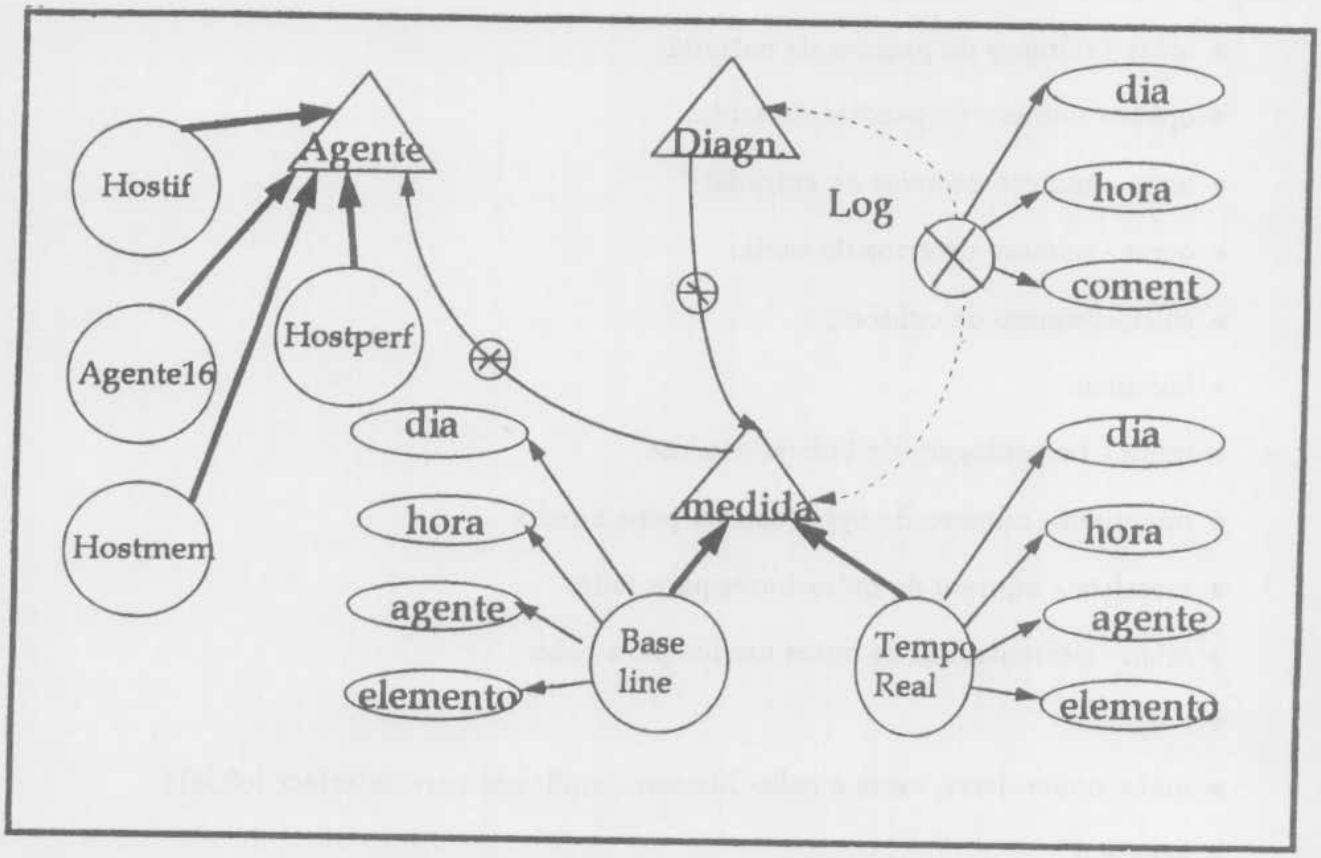

Figure 3: Modelagem utilizando o GSM

as primeiras abordagens à modelagem do sistema, representada através de GSM (Generic Semantic Model). Na figura abaixo, está a modelagem final do sistema:

A metodologia de construção de árvores de conhecimento, também proposta por [LEA 95], associa diagnósticos a seus fatores determinantes, que são ordenados conforme a importância. Através desse processo, foi construído um conjunto de árvores que representam a base do conhecimento do sistema. Dessas árvores foram derivadas as regras que compõem a base de regras do módulo diagnósticos e identificou-se que os diagnósticos são realizados em quatro níveis. Na Figura 4 pode-se identificar três deles: o inferior é o nível de parâmetros, onde é identificado o estado de cada parâmetro, sendo que este é avaliado em função de seu valor e da média e desvio correspondentes ao dia/hora atuais; o nível de diagnóstico parcial contém diagnósticos que são realizados através de diagnósticos de parâmetros, mas não são importantes para o usuário; o nível mais acima, de diagnósticos finais, é o que é apresentado para o usuário. Adicionalmente aos níveis da árvore que aparecem no diagrama, foi implementado, ainda, o nível de sugestão, onde, em função dos diagnósticos finais, são apresentadas sugestões para o administrador da rede.

O conhecimento está representado por fatos e regras. Os fatos são gerados a partir dos arquivos de monitoração e baseline. A implementação desse sistema foi feita em linguagem Prolog.

Os arquivos que chegam pelo sistema tem seu formato convertido para a forma de fatos Prolog. Os fatos possuem o seguinte formato:

- agente(nome-atributo,valor);
- anterior(nome-atributo,valor);
- agente(interface,nome-atributo,valor);

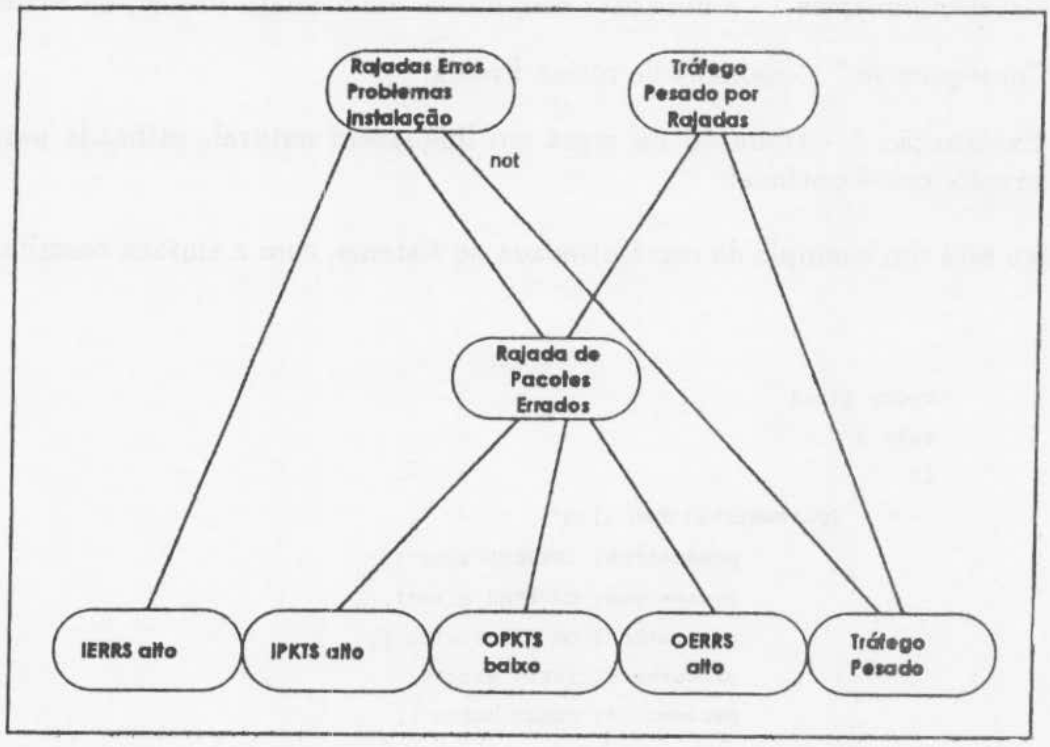

Figure 4: Representação parcial da árvore de conhecimento

- anterior(interface,nome-atributo,valor).

Os agentes hostif e agente 6 são dependentes das interfaces de cada máquina, por esse motivo eles armazenam, também, esse valor. O valor anterior de cada parâmetro deve ser guardado para o caso de medidas que são cumulativas, e só a diferença entre o anterior e o atual interessa.

Como o protótipo foi implementado em linguagem Prolog [CLO 84], para definir uma sintaxe para as regras que representam o conhecimento do domínio foram utilizados operadores. Definidos esses operadores, as regras podem ser escritas em uma sintaxe mais cômoda e adequada para o especialista e incluída no interpretador facilitando a implementação da máquina de inferência.

- teory "Teoria";
- rule " N ";
- if " ListaDeCondições ";
- then " Consequência ";
- because " Explanação ";

onde:

- "Teoria" - identifica o ruleset. Dessa forma a máquina de inferência pode ser disparada para um conjunto de regra apenas, ao invês de pesquisar toda a base de regras;

- " ListaDeCondições " - é uma lista conjuntiva, em formato Prolog, de condições;
- " Consequência " - chamada de rotina Prolog;
- " Explanação " - tradução da regra em linguagem natural, utilizada para a explanação, que é opcional;

Abaixo está um exemplo de regra utilizada no sistema, com a sintaxe descrita acima:

```
teory final
rule 6 :
if
        [parametros('CPU alto'),
                parametros('OPERCOL alto'),
                parametros('OAVEPER alto'),
                parametros('OAVEPER2 alto'),
                parametros('IPKTS alto'),
                parametros('OERRS baixo')]
then
                goal(final('Baixa performance'))
because
'foi detectado uso de CPU acima do normal e indicativos OAVEPER e OAVEPER2
maiores que zero, juntamente com quantidade de pacotes que entra e sai da
maquina  maior que o normal. Isto indica que a maquina esta carregada, pois
existe grande numero de colisoes no barramento e uma taxa de erros grande'.
```

Figure 5: Regra de Baixa Performance

O primeiro conjunto de regras serve de parâmetro que identifica situações anormais com relação aos parâmetros dados pelos agentes. Para tanto, o valor obtido no arquivo de monitoração gerado pelos agentes é comparado com o valor médio da baseline, levando em conta que a baseline representa o padrão "normal" da rede, para cada dia da semana e hora do dia.

O segundo conjunto é chamado parcial e detecta situações de alerta que devem ser investigadas pelo sistema, mas que não são apresentadas ao usuário.

O conjunto final leva em conta os diagnósticos de parâmetros e final para identificar situações que devem ser apresentadas ao administrador da rede.

As regras do conjunto sugestão levam em conta os diagnósticos finais e fornecem sugestões ao administrador para evitar ou contornar o problema. O mapeamento entre os níveis da árvore de conhecimento e os conjuntos de regras é direto. A organização da base de regras em "rulesets" torna mais fácil de implementar o mecânismo de inferência, bem como torna a busca mais eficiente.

A máquina de inferência foi implementada sobre o mecanismo de resolução do próprio Prolog, sobre o qual o protótipo foi implementado . Dado um conjunto de regras, o

```
diag(Teoria):-
        (teory Teoria rule N : if LCond then Cons;
                teory Teoria rule N : if LCond then Cons because _),
                testa_cond(LCond),
                call(Cons),
                goal(marca_trace(Teoria,N)),
                fail.
diag(_).

testa_cond([]).
testa_cond([C|R]):-
                call(C),
                testa_cond(R).
```

Figure 6: Algoritmo Prolog para Máquina de Inferência

mecanismo toma cada uma das regras daquele conjunto e testa suas premissas, encadeando para trás. O algoritmo Prolog é o que segue abaixo:

O algoritmo torna uma regra, seja ela com ou sem uma explanação (because), testa o conjunto de condições, representado por uma lista, executa a ação da regra, marca aquela regra como disparada [goal(marca-trace(Teoria, N)]. O registro das regras que foram disparadas é utilizado posteriormente pelo mecanismo de explanação. Depois de testada uma regra, é forçada uma falha para que, por "backtracking", as demais regras sejam testadas.

No protótipo, esse mecanismo de inferência é invocado para cada conjunto de regras, de forma seqüencial. Essa estratégia, bastante procedimental, torna a inferência mais fácil de ser implementada.

A interface explanatória busca os diagnósticos no topo da árvore (sugestão) que foram detectados, monta o caminho que levou a cada diagnóstico e monta um texto utilizando as sentenças "because" de cada regra disparada no caminho. A seguir segue uma representação desse algoritmo:

- seja T o conjunto de teorias na base de regras;
- t é a teoria no topo da hierarquia de teorias T;
- identificar o conjunto R de regras que pertencem à teoria t que foram disparadas;
- para cada regra r de R;
    1. montar o caminho C de regras disparadas que levaram ao disparo de r;

2. percorrer o caminho C, do topo para a base da árvore de conhecimento, tomando as sentenças "because".

O mecanismo de explanação monta toda a explanação de todos os caminhos de uma única vez, respondendo porque é dada uma sugestão pelo sistema de gerência pró-ativa, porque foi dado cada diagnóstico final, e assim por diante.

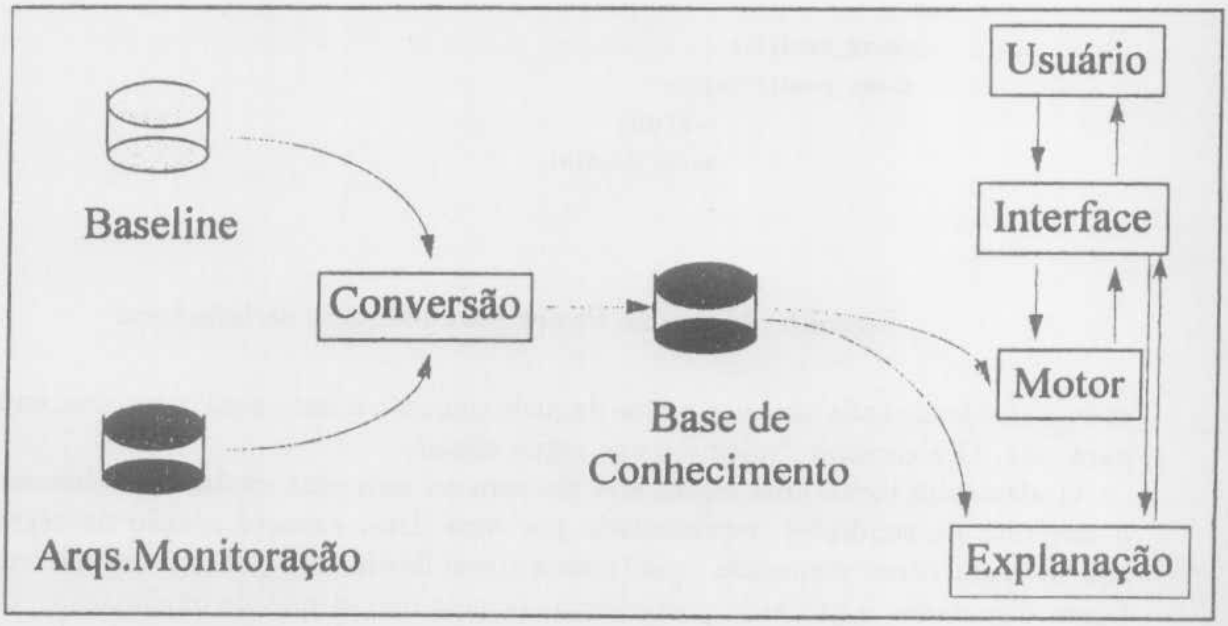

Figure 7: Arquitetura do Sistema

Assim, o módulo possui:

- blocos de conversão;
- base de conhecimento;
- motor de inferência;
- explanação;
- interface.

O bloco de conversão é responsável por receber os arquivos de monitoração e da baseline e convertê-los para o formato de fatos Prolog. Esses fatos irão formar a base de conhecimento.

O motor de inferência é utilizado para, a partir da base de conhecimento, analisar os valores dos parâmetros e inferir um diagnóstico. Esse diagnóstico é apoiado pela explanação que indica os motivos que levaram a tal situação problema, além de sugerir possíveis formas de resolução desses problemas.

Pela interface, o administrador da rede recebe as informações do sistema bem como as sugestões e tem a possibilidade de expor sua opinião, de concordar, ou não, com

o diagnóstico encontrado. Além disso, ele deve, também, em qualquer das situações, descrever qual dos rumos ele seguiu na tentativa de solucionar o problema.

4 Resultados

Por motivos de segurança, o protótipo atual roda em um microcomputador compatível IBM/PC conectado á rede e foi implementado em Arity Prolog. Para seus testes, os arquivos de monitoração gerados pelos agentes na máquina Darwin tiveram que ser copiados diretamente para o diretório do protótipo, sendo que, neste período de testes, se verificou apenas um tipo de problema.

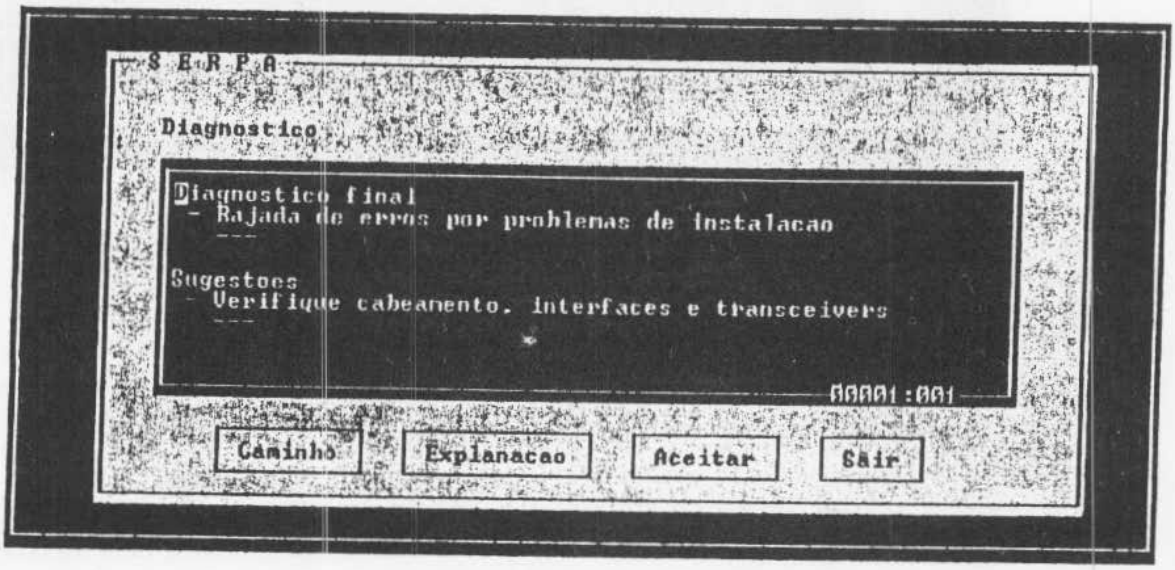

Figure 8: Tela de Exibição do Diagnóstico

Ao ser executado o protótipo, automaticamente, converteu os dados de monitoração e consultou as informações estatísticas da baseline, tomando como base a hora e data do sistema, informou ao usuário o diagnóstico final e as sugestões de ações a serem tomadas pelo administrador da rede.

Desse ponto, foi possível solicitar uma explanação:

A interface do sistema é composta por:

- Janela Central, o qual é utilizada para mostrar os diagnósticos ;
- Opção "Caminho", seria uma simplificação da explanação, mostrando quais as regras que foram disparadas, mas sem montá-las;
- Opção "Explanação", explica as regras utilizadas pelo sistema;
- Opção "Aceitar", registra no arquivo log;

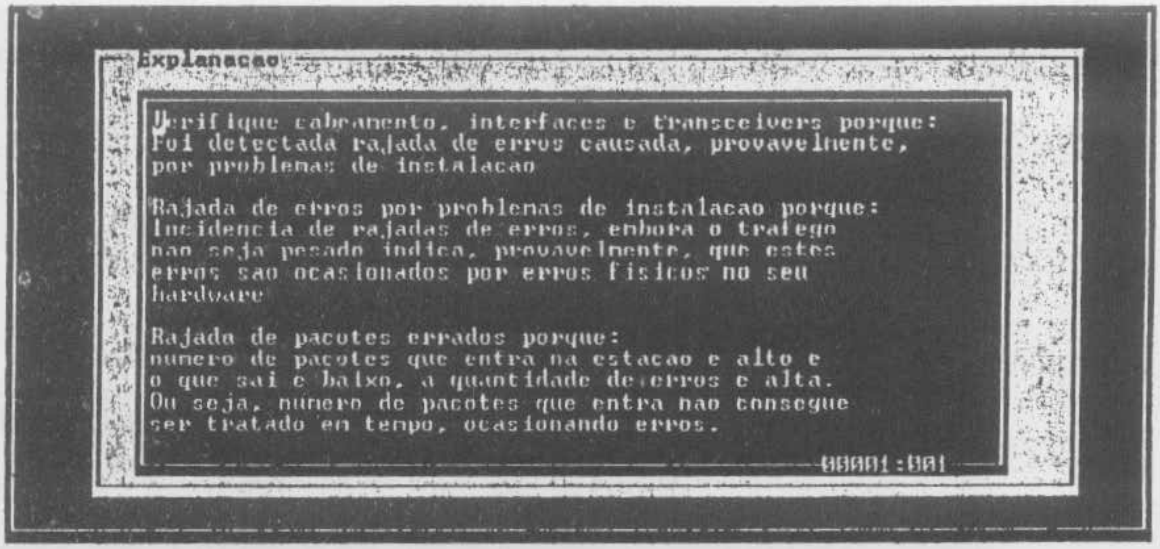

Figure 9: Tela de Explanação

- Opção "Sair", sai do sistema.

Caso a explanação tenha um número de linhas superior ao disponível no monitor de vídeo, ultrapassando o tamanho da janela, como ocorreu no exemplo, essa pode ser rolada para baixo:

A opção "Aceitar" registra em um arquivo de log, que o administrador aceitou as sugestões do sistema; se ele sair do sistema sem aceitá-las também fica registrado no arquivo de log a sua atitude. O objetivo do log é a validação do próprio sistema, podendo, posteriormente, serem investigadas as situações em que o administrador não aceitou as sugestões, de maneira a criar um histórico que possa ser utilizado pela máquina de inferência para construir os diagnósticos. A automatização desse processo é muito importante, porque na rotina de trabalho diária o administrador de uma rede, normalmente, não tem tempo de procurar manualmente em um arquivo de log, principalmente devido ao tamanho desses arquivos.

O protótipo comprovou a utilidade, pois o sintoma "rajada de pacotes errados" , dado a sua característica muito sutil, demandaria muito tempo para que se detectasse o problema, e, com a utilização dessa ferramenta, foi possível ao administrador tomar as providências de maneira rápida, visto que já havia recebido a notificação do evento antes que esse se desse por completo.

5 Conclusão

Com o objetivo de validar a utilização da gerência pró-ativa em redes de computadores, este trabalho propôs uma estratégia para a implementação da gerência pró-ativa utilizando inteligência artificial, mais especificamente, sistemas baseados em conhecimento.

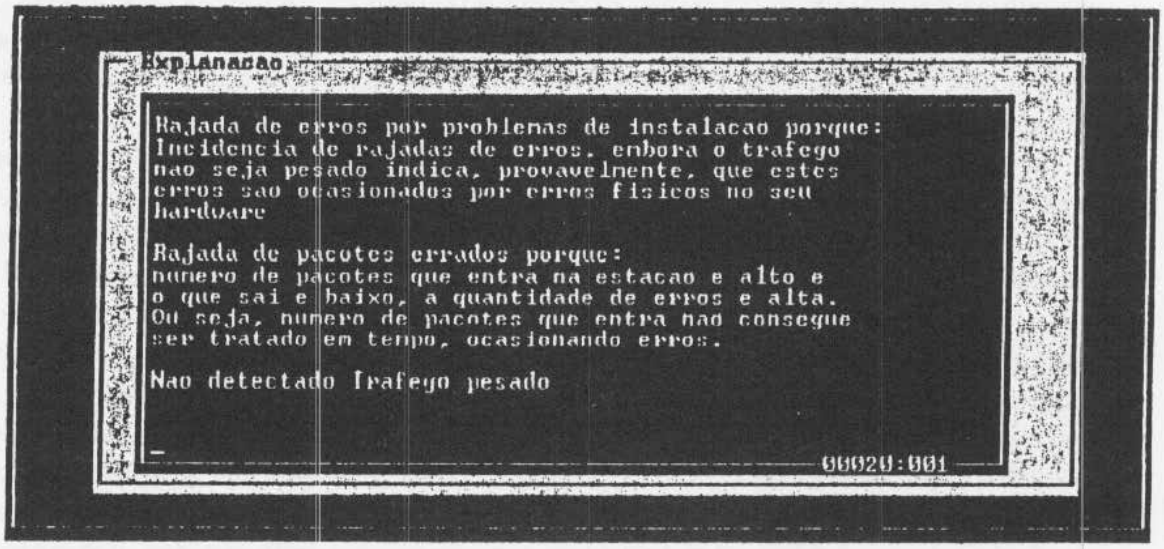

Figure 10: Rolagem da Tela de Explanação.

A área de gerência de redes escolhida para este experimento foi a gerência de desempenho analisando o problema de degradação do desempenho em uma máquina "gateway" (servidora da rede) observando como seu comportamento pode influenciar na rede.

Foram realizadas atividades teóricas e práticas que verificaram ser viável a utilização de agentes para a monitoração da rede e, de posse destes dados, construir uma base de dados contendo o estado normal da rede a "baseline". Assim, foi possível verificar o comportamento da rede e avaliar o desempenho da máquina "gateway" (servidora da rede). Para isto foram realizadas as seguintes tarefas: foi desenvolvido o agente 6, responsável pela monitoração da rede e análise de desempenho do barramento Ethernet; foi desenvolvido programas "parser" em linguagem C para colocar os arquivos de monitoração no formato dos arquivos componentes da baseline; foi montada uma baseline com os resultados do agente 6 mais os três agentes proprietários hostmem, hostperf, hostif; foi elaborado o módulo de diagnóstico com as características de um sistema especialista para averiguar se existe problema de performance, e em caso positivo, estabelecer os motivos que causam a degradação do desempenho. Ao ser executado o protótipo, automaticamente, converteu os dados de monitoração e consultou as informaçẽs estatísticas da baseline, tomando como base a hora e data do sistema, informou ao usuário o diagnóstico final e as sugestões de ações a serem tomadas pelo administrador da rede.

O protótipo comprovou a utilidade, pois o diagnóstico "rajada de pacotes errados" , dado a sua característica muito sutl, demandaria muito tempo para que se detecta-se o problema, e com a utilização desta ferramenta foi possível ao administrador tomar as providencias de maneira rápida, visto que já havia recebido a notificação do evento antes que este se desse por completo. Seria interessante ampliar o escopo deste trabalho e fazer uma pesquisa mais ampla de modo a ampliar o módulo de diagnóstico e aperfeioar este

protótipo. Além disso podemos citar os seguintes itens a serem melhorados: interligar o módulo diagnóstico com os arquivos de monitoração de maneira automática; melhorar a interface explanatória para emitir sugestões aos usuários por e-mail de modo automático; implementar outras áreas de gerência de redes, como gerência de configuração, falhas. Como novas tendências podemos citar os seguintes itens os quai acreditamos seja um seqüência natural deste trabalho: extender o trabalho para outras tecnologias como FDDI, ATM, etc; desenvolver o módulo de diagnóstico com um sistema conexionista; utilização do módulo de diagnóstico utilizando redes neurais; desenvolver a gerência pró-ativa usando uma combinação de inteligência artificial com simulação.

O problema de gerência de redes é por demais complexo e necessita ser abordado com o devido cuidado para não criar novos problemas ao tentar solucionar outros, ainda mais quando se fala, também, em gerência de performance , tópicos tradicionalmente tratados em áreas distintas. Em seu novo enfoque, a gerência pró-ativa, todos os cuidados devem ser redobrados, pois devemos antecipar o acontecimento de problemas e não induzir os administradores a buscar soluções para situações virtuais. Este artigo, além de apresentar uma estratégia de implementação, mostra resultados do modelo idealizado que serviram para mostrar que o comportamento pode ser alcançado. O experimento demonstrou que o CESUP possui um rede bem montada, de comportamento e performance bem estável, o qual não nos possibilitou um maior número de problemas a serem inspecionados, porém o problema que ocorreu foi detectado e reportado ao administrador, comprovando o comportamento pró-ativo. Seus fundamentos e príncipios podem ser utilizados como uma base a novos trabalhos a serem realizados com o modelo proposto.

References

[COM 91] COMER, D. E. **Internetworking with TCP/IP: Principles, Protocols, and Architeture**. Volume 1. Seg. Edição. Prentice-Hall, Englewood Cliffs, NJ, 1991.

[JAN 93] JANDER, Mary. **Proactive LAN Management: Tools that Look for Trouble to Keep LANs Out of Danger.** Data Communications. Março de 1993.

[POS 81] POSTEL, J.B. **Internet Protocol.** Request for Comments 791, DDN Network Information Center, SRI International, Setembro, 1981, 45pp.

[POS 81a] POSTEL, J.B. **"Internet Control Message Protocol - DARPA Internet Program Specification"**. Request for Comments 792, 1981.

[SUN 89] Sun Microsystem Inc. **SunNet Manager Tutorial - How Write an Agent**, 1989.

[SUN 89a] Sun Microsystem Inc. **Network Programming Guide**, 1989.

[STE 90] STEVENS, W.R. **UNIX Network Programming**.Prentice-Hall Inc. Englewood Cliffs, NJ, 1990.

[SUN 90] Sun Microsystems Inc. **SunOS Reference Manual**. Vol I, 1990.

[WES 88] WESTPHALL, C. B. **Proposição de Funções em Gerência de Comunicação de Dados**. *Dissertação de Mestrado*, Porto Alegre, UFRGS-CPGCC, Maio 05, 1988.

[WES 91] WESTPHALL, C. B. **Conception et développement de l'architecture d'administration d'un réseau métropolitain**. *Thèse de Doctorat nouveau régime*. Université Paul Sabatier. Toulouse, 16 Juillet 1991.

[WES 92] WESTPHALL, C. B. & ASSUOL, S. **Management Architecture for Networks of the Future**. IFIP/IEEE International Workshop on Distributed Systems: Operations & Management. October 12-13, 1992. Munich, Germany.

[CLO 84] CLOCKSIN, W. F. **Programming in Prolog**. Second Edition.Cambridge, Springer-Verlang, England, UK, 1984.

[LEA 95] LEÃO, Beatriz. **Notas de Aula da Disciplina COMP02**. CPGCC/UFRGS. Porto Alegre, 1995.

[HUL 87] HULL, R., KING, Roger. **Semantic Database Modeling: Survey Applications and Research Issues**. ACM Computing Surveys. v.19, n.3 Setembro, 1987.

[ROC 94] ROCHA, M. A. **Uma Estratégia Para Implementar Gerência Pró-Ativa em Redes de Computadores**. *Trabalho Individual I*, Porto Alegre, UFRGS-CPGCC, Março 05, 1994.

[ROC 94] ROCHA, M.A. & WESTPHALL, C. B. **Gerência de Redes de Computadores Através de Novos Agentes**. *Anais do XII Simpósio Brasileiro de Redes de Computadores*, p.113-133, Curitiba, PR, Brasil 1994.