UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Um Discriminador Inteligente de Eventos de Rede para o Ambiente CINEMA

por

CRISTINA MOREIRA NUNES

Dissertação submetida à avaliação, como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Liane Margarida Rockenbach Tarouco
Orientador

Porto Alegre, maio de 1997.

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Nunes, Cristina Moreira

Um Discriminador Inteligente de Eventos de Rede para o Ambiente CINEMA / Cristina Moreira Nunes. — Porto Alegre: CPGCC da UFRGS, 1996.
152 f.: il.

Dissertação (mestrado) — Universidade Federal do Rio Grande do Sul. Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, BR-RS, 1996. Orientador: Tarouco, Liane M. R.

1. Redes de Computadores. 2. Gerência de Redes de Computadores. I. Tarouco, Liane M.R. II. Título.

Agradecimentos

Gostaria de agradecer em primeiro lugar à profa. Liane Tarouco pela sua orientação. Agradeço também à CAPES pelo auxílio financeiro e aos funcionários do Instituto de Informática, em especial à Eliane, pela simpatia, cafezinhos, docinhos...

Gostaria de agradecer ao Marcos pelo carinho, compreensão e paciência que teve durante esse período. Agradeço também pela força e incentivo que sempre me deu, mesmo estando longe.

Agradeço ao pessoal do POP-RS pela paciência e auxílio prestado em várias ocasiões, às minhas amigas da PUC que estiveram sempre presentes e aos meus colegas do Pós pelo clima amigável que sempre existiu.

Um agradecimento especial aos meus pais e às minhas irmãs por todo o apoio e compreensão que sempre tiveram.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

API	Application Programming Interface
ASN.1	Abstract Syntax Notation One
AUI	Attachment Unit Interfaces
ATM	Agente de Transferência de Mensagens
CINEMA	Cooperative Integrated Network Management
CMIP	Common Management Information Protocol
CMIS	Common Management Information Services
CMOT	Common Management Information Protocol over Transmission Control Protocol - TCP
FCS	Frame Check Sequence
HP	Hewlett-Packard
ICMP	Internet Control Message Protocol
IP	Internet Protocol
ISO	International Organization for Standardization
ISODE	ISO Development Environment
MAD	Módulo de Automatização de Diagnóstico
MAU	Medium Attachment Units
MHS	Message Handling System
MIB	Management Information Base
MTBF	Mean Time Between Failure
MTTR	Mean Time to Repair
NMS	Network Management Station
NOC	Network Operation Center

OSI	Open Systems Interconnection
PDU	Protocol Data Unit
QoS	Quality of Service
RMON	Remote Network MONitoring
SDL	Specification and Description Language
SGMP	Simple Gateway Management Protocol
SMI	Structure of Management Information
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
TDR	Time Domain Reflectometer
TTL	Time to Live
UDP	User Datagram Protocol

Resumo

Com o crescimento das redes de computadores e, principalmente, de sua importância para as organizações, o gerenciamento de redes tornou-se fundamental. Contudo, o gerenciamento de redes é um processo difícil dada sua complexidade e mudanças freqüentes em sua configuração. O ideal seria que sistemas pudessem fazer o trabalho de administradores de redes, reduzindo com isso a sobrecarga de trabalho dos administradores, ou seja, essas pessoas poderiam realizar outras tarefas enquanto o sistema ficaria gerenciando a rede.

O principal objetivo deste trabalho é propor um paradigma que, à medida que seja constatada a ocorrência de algum problema na rede, se tenha um módulo com inteligência suficiente para diagnosticá-lo, determinando porque aquele problema ocorreu. O trabalho descreve um sistema especialista para a gerência de redes que é integrado com um sistema de registro de problemas. Módulos especializados, orientados à análise de aspectos específicos do comportamento da rede, efetuam uma análise das características e do *status* da mesma, filtrando eventos e tentando prover resposta automatizada e/ou recomendações sobre cursos de ação para as anormalidades percebidas. Portanto, com esse objeto, definiu-se um sistema denominado MAD (Módulo de Automatização de Diagnóstico), no qual são definidas regras para diagnosticar os problemas ocorridos e também tentar prover ao usuário a maior qualidade de serviço possível. Este sistema efetua monitorações diárias sobre objetos da MIB II (*Management Information Base*) para tentar localizar os problemas da rede e então gerar alertas ao administrador da rede dependendo de sua gravidade.

Este trabalho é um sub-projeto do projeto CINEMA (*Cooperative Integrated Network Management*). No projeto CINEMA, foram especificados módulos de manuseio de registros de problemas numa base de dados (Sistema de *Trouble Ticket*) e uma plataforma básica para configurar a obtenção de informações sobre a rede - acesso a objetos gerenciados (Sistema de Alertas). O MAD atua como um integrador entre esses dois sistemas, filtrando eventos e gerando alertas ao administrador da rede na forma de um *trouble ticket*, isto é, um registro de problemas.

A validação deste sistema foi realizada através da implementação de um protótipo. O protótipo utiliza o protocolo SNMP (*Simple Network Management Protocol*) para fazer *polling* em objetos da MIB II de determinados componentes da rede. Sua implementação foi útil para que se pudesse fazer o refinamento das regras, tornando-as mais de acordo com a rede monitorada.

Palavras-chave: Redes de Computadores, Gerência de Falhas e Desempenho, Alertas, Protocolo SNMP, Sistemas Especialistas.

Title: "A Network Events Intelligent Discriminator for the CINEMA Environment"

Abstract

With the growth of computer networks and, mainly, of its importance to organizations, the management of networks has become fundamental. However, network management is a hard process given its complexity and its frequent configuration changes. The ideal would be if the systems could play the role of network managers, thus reducing the job overload from the managers. It means that these managers could perform other tasks while the system manages the network itself.

The main objective of this work is to propose a paradigm so that, as any trouble is found in the network, a module intelligent enough to diagnose this problem will identify the reason why that has occurred. The work describes an intelligent system for network management which is integrated to a trouble ticket system. Specialized modules, directed to the analysis of specific aspects of the network behavior, bring about an analysis of its characteristics and status, filtering events and trying to supply an automated answer and/or advices about action paths to deal with the abnormalities. Therefore, with this goal, a system called MAD (Diagnosis Automatization Module) has been defined, in which rules to diagnose the most common troubles are specified and that tries to supply the user with a quality of services as higher as possible. This system executes daily monitoration on MIB II (Management Information Base) objects trying to locate the troubles in the network and then generating alerts to the network manager depending on its severity.

This work is a sub-project for the project CINEMA (Cooperative Integrated Network Management). The project CINEMA specified modules of trouble ticket handling in a database (Trouble Ticket System) and a basic platform to configure the acquisition of information about the network - access to managed objects (Alerts System). MAD acts as an integrator between these two systems, filtering events and generating alerts to the network administrator in the form of a trouble ticket.

The validation of this system has been done through the implementation of a prototype. The prototype uses the SNMP protocol (Simple Network Management Protocol) to do polling in MIB II objects located in certain network components. Its implementation has been useful to refine the rules, making them fit the network under management.

Keywords: Computer Networks, Fault and Performance Management, Alerts, SNMP Protocol, Expert Systems.

1 Introdução

Gerenciar uma rede de computadores não é uma tarefa fácil e deve ser feita por uma ou mais pessoas muito bem treinadas. Ao gerenciar uma rede é importante analisar todos os seus aspectos, desde cabos até aplicações. Para tanto, existe uma série de ferramentas especializadas em detectar problemas, tanto a nível físico (verificando cabos e conectores), como também analisando os protocolos de níveis superiores.

A gerência em redes de computadores adveio da necessidade de controlar as redes complexas e heterogêneas que estão surgindo com o avanço da tecnologia. Geralmente, organizações investem quantidades significativas de tempo e dinheiro construindo redes complexas que precisam ser gerenciadas. Muitas vezes seria mais rentável se um sistema pudesse vigiá-las e informar ao administrador quando um problema fosse verificado do que dedicar uma ou mais pessoas somente para realizar esta tarefa. Desta forma, o administrador poderia trabalhar em outras tarefas enquanto o sistema ficaria monitorando a rede.

A necessidade de gerenciar o ambiente de redes fez com que fossem criadas ferramentas de gerência para auxiliar no trabalho de administração. A monitoração manual é um processo muito suscetível a erros e que, por consumir muito tempo, diminui a produtividade das pessoas envolvidas com o gerenciamento da rede [MAD 94]. Portanto, é importante ter ferramentas que avaliem seu desempenho de forma ágil. Estas ferramentas devem estar aptas a analisar e gerar relatórios a partir das informações que coletam, permitindo assim verificar as tendências e o perfil de tráfego, possuir um planejamento para futuras expansões e efetuar manutenções preventivas antes que problemas ocorram.

1.1 O cenário de redes atualmente

Hoje em dia é muito comum encontrar redes com tecnologias diferentes trabalhando em conjunto. Os protocolos, ou regras de comunicação, determinam o nível de entendimento entre estações de trabalho.

Anteriormente ao desenvolvimento do Modelo de Referência OSI (*Open Systems Interconnection*) da ISO (*International Organization for Standardization*), a maioria das arquiteturas de rede eram proprietárias, [MIL 91]. Como fornecedores não usavam padrões abertos designados pelos comitês de especialistas internacionais, o software de um fornecedor, tal como IBM não poderia interoperar com o software da DEC, por exemplo, que, na verdade, não poderia se comunicar diretamente com os sistemas da HP (*Hewlett-Packard*). Nesses casos, um *gateway* era necessário para traduzir protocolos de todos os níveis dessas arquiteturas proprietárias.

Um dos objetivos do conceito de sistemas abertos é fornecer interoperabilidade de protocolos entre sistemas de fornecedores diferentes. Teoricamente, se todos os fornecedores seguissem para uma arquitetura comum, os problemas de comunicação entre esses sistemas iriam desaparecer.

Infelizmente, muitos fornecedores continuam adicionando partes proprietárias para ganhar, recuperar ou manter uma fatia do mercado. Portanto, essas múltiplas arquiteturas requerem dispositivos adicionais, tais como *gateways*, para assegurar o nível de interoperabilidade necessário para uma comunicação confiável.

Abaixo estão listados alguns problemas que podem ocorrer em uma rede, [NAS 94]:

- nodos não estão operando perfeitamente em um segmento;
- a rede apresenta uma alta quantidade de erros;
- toda a rede está operando lentamente;
- problemas em acessar o servidor de arquivos da rede;
- um nodo ou periférico não pode ser acessado;
- uma impressora ou impressoras da rede não podem ser usadas ou acessadas;
- problemas em acessar ou usar outros segmentos interconectados através de uma *bridge*, um roteador, um *hub* inteligente ou um repetidor;
- problemas no cabeamento.

Além desses, muitos outros problemas podem ocorrer em uma rede. Para cada problema listado anteriormente há uma forma diferente de verificar qual foi a causa para então corrigi-la. Por exemplo, se um *gateway* não está acessível, o problema pode ser devido a uma configuração inapropriada do *gateway* na rede, mas também pode ser devido a uma possível falha nesta máquina.

1.2 Como pode ser feito o diagnóstico atualmente

Dada uma mistura de fatores, tais como arquiteturas centralizadas versus distribuídas, facilidades de transmissão em banda larga, múltiplos protocolos, novos métodos para conectar LAN/WAN, e sistemas de gerenciamento de redes diferentes, é importante saber como analisar uma rede. A rede ainda pode usar uma combinação de tipos de cabos distintos, incluindo par trançado, cabo coaxial e fibra óptica. O hardware de rede local pode incluir métodos de acesso, tais como CSMA/CD (IEEE 802.3) e *token passing* (IEEE 802.5). Dispositivos de conectividade podem usar diferentes algoritmos, tais como *spanning tree* (IEEE 802.3) ou *source routing* (IEEE 802.5). Para gerenciar todas essas diferenças é necessário uma linha básica para saber por onde começar, [MIL 91]:

- 1° deve-se definir o problema;
- 2° desenvolver uma solução;
- 3° documentar o trabalho;
- 4° disseminar os resultados.

Atualmente, existem protocolos especializados no gerenciamento de redes. O padrão ISO, conhecido como CMIP (*Common Management Information Protocol*), está descrito na ISO 9595 e na ISO 9596. Uma variante desta solução é o CMOT (*Common Management Information Protocol over Transmission Control Protocol - TCP*). Contudo, o protocolo de gerenciamento de rede mais popular é o SNMP (*Simple Network Management Protocol*), que foi desenvolvido pela comunidade Internet (TCP/IP). Sua popularidade é devido à sua simplicidade.

Para auxiliar o administrador de uma rede a encontrar a causa de um problema, existem ferramentas de monitoração e analisadores de protocolos que podem ser utilizados. A maioria desses analisadores utilizam o protocolo SNMP para gerenciar as redes, manipulam o gerenciamento de eventos e empregam inteligência artificial para reduzir o tempo gasto para resolver problemas. Segundo [JAN 96], a maioria dos fornecedores de softwares de gerenciamento de redes estão replanejando suas plataformas para suportar o ambiente distribuído e suas aplicações estão emergindo para traduzir informações do mundo real dentro de procedimentos pró-ativos.

1.2.1 Ferramentas de diagnóstico existentes

Quando vai-se analisar as ferramentas de diagnóstico existentes hoje em dia constata-se que existe uma variedade de softwares de diagnósticos produzidos por diferentes fabricantes. Embora as capacidades operacionais de tais softwares variem consideravelmente, alguns deles possuem características semelhantes, tais como monitorar a linha de comunicação, observar o estado dos condutores na interface física ou transmitir padrões de dados e observar a resposta do equipamento.

Para diagnosticar problemas de nível físico, tais como problemas de cabos, pode-se utilizar equipamentos como TDR (*Time Domain Reflectometer*) ou algum outro testador de cabo, [NUN 95]. Problemas de cabo contribuem para uma percentagem muito alta de falhas nas redes e por isso, possuir equipamentos próprios que verifiquem tais problemas, é muito importante.

Existem também ferramentas cujo trabalho é simplesmente converter, decodificar a informação de controle que vem com o protocolo e mostrá-la ao usuário em um formato que ele entenda. Essas ferramentas são chamadas de analisadores de protocolos. Um analisador de protocolo monitora a rede em tempo real. Eles são conectados a uma rede e, então, capturam os dados que estão trafegando por ela para posterior decodificação e análise.

Um analisador de protocolo permite monitorar uma rede transparentemente, sem interferir em nenhuma transmissão. Ele pode ser usado para uma variedade de propósitos incluindo encontrar falhas de software e hardware, otimizar a rede e isolar cabos com problemas, [MIL 91]. Muitos dos analisadores de protocolos incluem funções de testadores de cabos, mas diferem de ferramentas que fornecem somente visões instantâneas da rede, tais como multímetros ou TDRs. Um analisador de protocolo pode ficar observando a rede vinte e quatro horas por dia e pode indicar problemas que são suspeitos mais difíceis de serem localizados sem uma supervisão contínua sobre a rede.

Devido as redes atualmente estarem conectadas com diferentes tecnologias e múltiplos protocolos, existem algumas características que são desejáveis em tais analisadores:

- utilização de sistemas especialistas;
- permitir executar em computadores pessoais (PC's);
- possuir uma interface gráfica com o usuário;
- analisar combinações de LAN/WAN;
- possuir características de multiporta e multiprocessamento;
- suportar protocolos de MAN/WAN;
- suportar protocolos de gerenciamento de rede;
- analisar ou coletar dados remotamente;
- ter integração com ferramentas de simulação e modelagem de rede.

É claro que nem todos os analisadores de protocolos possuem todas essas características, mas cada vez mais elas vêm sendo incorporadas por eles, que tentam acompanhar a evolução permanente das redes.

Abaixo estão listados alguns analisadores de protocolos com seus respectivos fabricantes, [NUN 95] [NAS 94]:

- Chameleon 100 - Tekelec Inc.
- DA 30 - Wandel & Goltermann Technologies Inc.
- Dataglance - IBM
- Dolphin ESP (*Expert System Protocol*) e Dolphin ESP Plus - Dolphin Networks
- Expert Sniffer - Network General Corp.
- Fireberd 500 - Telecommunications Techniques Corporation
- Framethrower - LANquest Labs Inc.

- IBM Network Manager - IBM
- LANalyzer - Network Communication Corporation
- LANdecoder -Triticom
- LANpharaoh - Azure Technologies Inc.
- LANvista - Digilog
- LANwatch - FTP Inc.
- Network Advisor - Hewlett Packard Network Test Division
- Powerbits 3201 - Alantec

Devido à carência de recursos humanos qualificados para gerenciar uma rede, torna-se difícil, muitas vezes, para a pessoa que está administrando a rede, interpretar os resultados que são mostrados por essas ferramentas. Além disso, existem ferramentas de diagnóstico que dão resultados muito incompletos ou superficiais, tornando difícil integrá-los para auxiliar na solução do problema.

Para usar um analisador de protocolos de forma eficiente, a pessoa responsável deve ter uma boa percepção dos seguintes domínios:

- conhecimento da arquitetura da rede;
- consciência da metodologia de análise dos protocolos;
- entendimento dos modos operacionais básicos do analisador de protocolo usado.

Uma forma comum de verificar quando algum problema está ocorrendo na rede é através de uma indicação na forma de um alerta. Em um determinado momento, vários alertas podem estar sendo gerados e analisando-se cada um deles pode-se descobrir a origem de muitos problemas. Alertas estão freqüentemente ligados a congestionamentos, mas quando ocorre uma falha eles podem ser colocados em serviço de forma manual ou automática, [MUL 90].

Uma estudo mais aprofundado sobre ferramentas para diagnóstico de nível físico e analisadores de protocolos está descrito em [NUN 95].

Além dos analisadores de protocolos existem sistemas criados especialmente para gerenciar uma rede. Geralmente sistemas de gerenciamento de rede possuem capacidades de controle e diagnóstico e, assim, conseguem gerar alertas quando os componentes da rede não estão trabalhando adequadamente. Quando um monitor local detecta um problema, ele coloca um alerta na tela do operador. Muitos tipos de *modens*, *switches* e multiplexadores emitem alertas negativos, isto é, algumas luzes que estão acesas nos seus painéis são apagadas quando um problema for detectado. Esse tipo de alerta pode não ser notado pelo operador.

Sistemas de gerenciamento de rede, por outro lado, fornecem um sinal de alerta positivo, tal como uma mensagem no terminal do operador. Este sinal não indica apenas qual a falha que ocorreu, mas normalmente fornece uma informação sobre a natureza da falha e a sua localização. Um *display* colorido, por exemplo, torna mais fácil para o operador notar um problema e verificar sua severidade. Portanto, pode-se utilizar indicadores de alertas multinível nos quais a cor vermelha usualmente indica uma falha, verde indica operação normal, enquanto amarelo é usado para indicar uma condição de possível falha. Além disso, alertas podem aparecer piscando na tela do operador ou emitindo um sinal sonoro para indicar que algo de errado está acontecendo com a rede.

Tais ferramentas são discutidas com maior detalhe e comparadas em [MED 96], apresentando como características básicas:

- interface gráfica, tal como figura 1.1;
- especificação de filtros simples com geração de alertas.

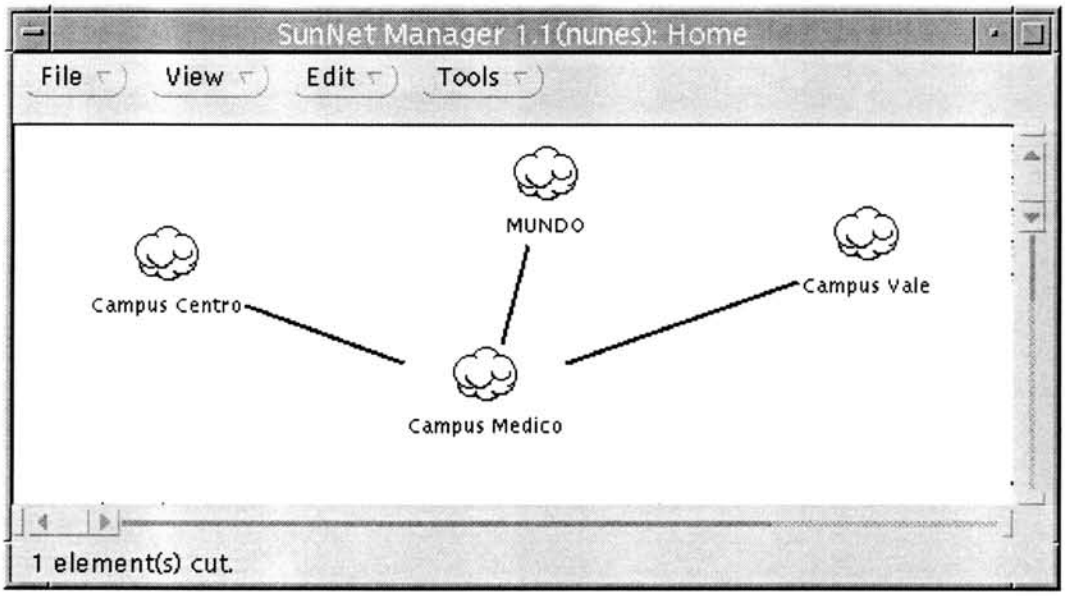

FIGURA 1.1 - Exemplo de uma ferramenta com interface gráfica.

Todavia, elas são bastante limitadas no que tange à correlação de alertas e na integração com sistemas de registro de problemas.

Assim, o sistema objeto deste trabalho se diferencia dos produtos existentes no mercado pelo grau de inteligência agregado ao processo de análise dos eventos percebidos na rede e sua avaliação contextualizada, isto é, um alerta pode ser ignorado tendo em vista o histórico recente da rede (já haver registro sobre aquele tipo de problema ou sobre um outro fator causador já conhecido).

1.3 Objetivos

Pelo fato de não haver um número adequado de pessoas capacitadas para gerenciar uma rede e algumas ferramentas fornecerem resultados difíceis de serem analisados, o principal objetivo deste trabalho é definir um paradigma para automatizar o diagnóstico de problemas em redes. Para tanto foi projetado um módulo chamado Módulo de Automatização de Diagnóstico (MAD) que é um sub-projeto do projeto CINEMA (*Cooperative Integrated Network Management*) [MAD 93].

O projeto CINEMA é composto por dois sistemas, o Sistema de *Trouble Ticket* e o Sistema de Alertas. No primeiro é feito o manuseio de registros de problemas, no qual estes possuem a história completa do problema. O segundo foi definido para fazer a monitoração contínua sobre a rede e comparar os valores coletados com "limites" que podem ser fornecidos pelo usuário ou calculados automaticamente pelo sistema. Quando esses "limites" forem ultrapassados, eventos serão gerados pelo sistema.

O MAD atuará como um integrador entre esses dois sistemas. Ele fará o papel de um Processador de Eventos inteligente, processando os eventos recebidos do Sistema de Alertas e transformando-os em alertas quando eles forem considerados críticos. Para cada alerta gerado pelo MAD, um registro de problemas será aberto para ele. Este módulo trabalhará como um sistema especialista, atuando nas áreas de gerenciamento de desempenho e de falhas. Para que problemas sejam descobertos, são efetuadas monitorações em determinados objetos da MIB II (Management Information Base) através de operações do protocolo SNMP (Simple Network Management Protocol). Após cada monitoração os valores coletados podem ser transformados em eventos. Sempre que isso ocorre, os eventos são passados por uma base de regras, a qual foi definida neste trabalho, e que verifica se há necessidade de ser gerado um alerta para o administrador da rede.

Os capítulos subseqüentes deste trabalho estão a seguir indicados. O capítulo 2 apresenta uma revisão de alguns problemas típicos que podem ocorrer em uma rede. O capítulo 3 descreve resumidamente o funcionamento do modelo de gerência Internet, bem como a facilidade de gerenciar equipamentos utilizando para tanto o protocolo SNMP. O capítulo 4 trata sobre a qualidade de serviço que a rede deve prestar ao usuário; além de apresentar as características ideias de um Sistema de *Trouble Ticket* e descrever o Ambiente CINEMA. O capítulo 5 apresenta as características do paradigma proposto. Nele estão especificadas as redes semânticas utilizadas para montar a base de regras do MAD, bem como os detalhes do seu funcionamento. O capítulo 6 descreve aspectos de especificação e implementação do protótipo desenvolvido, e faz uma avaliação dos resultados obtidos durante a fase de testes do protótipo. Por fim, o capítulo 7 apresenta as conclusões sobre o trabalho.

2 Revisão dos Problemas Típicos

Em uma rede de computadores ocorre freqüentemente uma série de problemas por motivos diferentes. Alguns desses problemas podem ser verificados quando o desempenho da rede se degrada. Muitos dos problemas encontrados em uma rede podem ser devido a alguma falha em seus componentes. Cada componente de rede quando está com algum problema gera um sintoma que pode ser percebido pelo administrador. Métodos distintos podem ser aplicados pelos administradores para detectar tais problemas.

Muitas vezes, quando se analisa um sintoma de algum problema na rede, esta análise pode indicar que seu causador é um determinado componente da rede, outras vezes ela indica uma rede local específica como sendo sua originadora. O primeiro passo quando se vai corrigir um problema é tentar isolar a falha, até se chegar ao componente específico que a causou, o que às vezes requer uma intervenção manual. Geralmente, esse processo de detecção requer do administrador que uma série de passos sejam aplicados. Uma boa forma para se começar a detecção de problemas em qualquer rede é partindo-se do nível físico. Para tanto, pode-se verificar se a infra-estrutura de cabeamento está funcionando adequadamente. Além disso, é muito importante avaliar seu *layout* e configuração.

Algumas vezes, redes podem ter altos níveis de tráfego ou sobrecargas gerais devido a problemas de comunicação de protocolos. Se o protocolo tem um erro, ele pode causar um alto nível de tráfego *broadcast*. Incompatibilidades na versão de protocolos ou na própria configuração do protocolo podem causar altas taxas de retransmissões na rede. Isso torna-se evidente quando *drivers* de rede são de diferentes versões ou quando sistemas operacionais de rede são incompatíveis.

A seguir são descritos alguns dentro os mais comuns problemas que podem ocorrer em uma rede de computadores.

2.1 Problemas em *Bridges*, *Switches* e Roteadores

Um *switch* é um dispositivo projetado para solucionar problemas de desempenho de redes locais resultantes da falta de largura de banda e congestionamento na rede. Como uma *bridge*, um *switch* toma uma decisão relativamente simples para seguir quadros adiante baseada no endereço MAC contido em cada quadro, mas ao contrário da *bridge*, um *switch* pode enviar dados com retardo muito baixo, fornecendo um desempenho maior. Esta tecnologia permite que a largura de banda seja aumentada ou reduzida em segmentos de redes locais para poder reduzir o congestionamento.

Um dos mais freqüentes problemas que ocorrem em *bridges* e *switches* é sua configuração incorreta e suas tabelas de endereços corrompidas. Uma forma de analisar se uma *bridge* está direcionando quadros corretamente é fazer uma análise fim-a-fim. Para tanto, deve-se verificar os quadros que são transmitidos de um lado para outro. Cada quadro pode ser decodificado por ambos os lados, e a pessoa que está analisando pode identificar se a *bridge* envolvida na transferência está direcionando e transferindo o quadro através da rota correta. A figura 2.1 mostra um exemplo conceitual da transferência de uma rota incorreta causada por uma *bridge*.

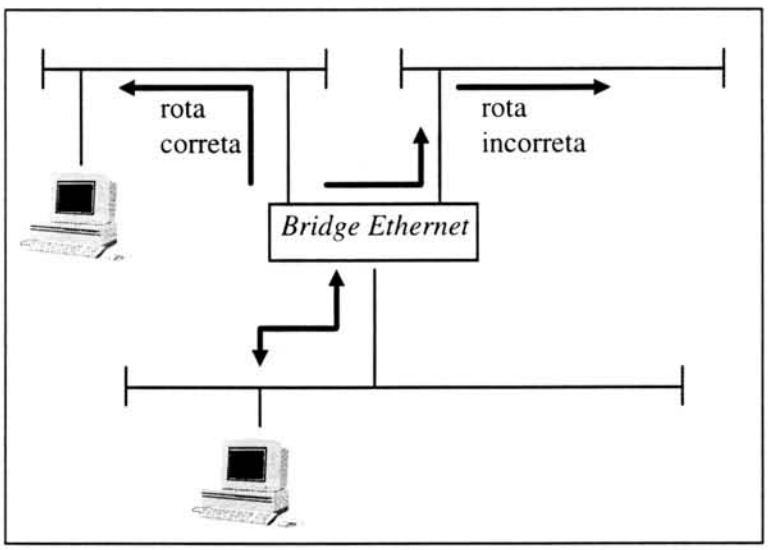

FIGURA 2.1 - *Bridge* Ethernet causando rota incorreta.

Analisadores de protocolos são muito úteis quando utilizados para inspecionar roteadores Ethernet, porque quadros transferidos podem ser capturados e então decodificados. Protocolos de roteamento são usados entre roteadores para atualizar múltiplas rotas em uma localização. Segundo [NAS 94], certos protocolos de roteamento podem ser decodificados para mostrar o número de redes que foram atravessadas durante uma transferência. Pegando-se como exemplo a família de protocolos TCP/IP, o nível IP inclui o campo *time to live* (TTL), que quando capturado nos níveis mais baixos, pode mostrar um *loop* de roteamento. Um *loop* de roteamento pode ser definido como uma condição que ocorre quando um quadro roda continuamente em um segmento da rede, em rota circular, até que ele seja descartado pelo roteador.

Os roteadores na Internet utilizam o esquema de endereçamento IP para transferir e rotear pacotes. Quando há uma falha na configuração do roteador, pode ocorrer um problema conhecido como *loop* de roteamento. Uma forma de diagnosticar esse problema é capturando os pacotes que trafegam por alguns roteadores. Se a maioria dos pacotes capturados possuírem um valor de TTL muito baixo, dois a um, pode-se dizer que a causa do *loop* de roteamento é devida a algum roteador. Para descobrir qual

roteador está causando o problema, deve-se analisar todos os roteadores por onde os pacotes estão trafegando.

Uma outra causa de problemas em roteadores é a configuração de um endereço Internet incorreto. A configuração de um sistema de endereços para roteamento IP é uma tarefa complexa. Esta tarefa requer uma atenção cuidadosa no momento de documentar e coordenar o projeto. Quando um dispositivo IP não pode ser alcançável em uma rede, um roteador pode ser a causa.

Para testar conectividade de um dispositivo e um roteador, pode-se invocar um teste de ***ping*** para alguma máquina. Este teste consulta o roteador e nodo através do protocolo ICMP (*Internet Control Message Protocol*). Se o equipamento não pode ser alcançado, deve-se verificar sua configuração e operação.

A maioria dos problemas em um roteador são causados por configuração incorreta. Outro exemplo de configuração incorreta pode ser verificada quando um roteador está passando somente um tipo de protocolo. O problema é comumente uma configuração incorreta no filtro do protocolo, o qual deixaria passar apenas um protocolo particular.

A maioria das *bridges*, *switches* e roteadores no ambiente Ethernet tem um grupo de LED's para diagnóstico que mostram informações sobre transferência de pacotes e atividades da porta. O estado operacional de uma porta de uma *bridge* ou de um roteador pode às vezes ser determinado pelo estado desse LED. Os manuais do fabricante, com recomendações específicas, precisam ser consultados antes que qualquer ação seja tomada quando alguma indicação de estado anormal ocorrer.

O processo de detecção de problemas para aqueles que aparecem como sendo da operação de uma *bridge*, *switch* ou roteador em uma rede é o seguinte:

1. verificar a configuração e endereços da *bridge* ou roteador;
2. verificar os diagnósticos disponíveis para localizar a causa;
3. por último, e provavelmente o mais eficiente, utilizar um analisador de protocolos.

Um módulo definido como MAR [HAR 97] será integrado ao Projeto CINEMA e tratará exclusivamente de problemas de roteamento.

A tabela abaixo, tabela 2.1, especifica alguns tipos de problemas que ocorrem em *bridges* e fornece possíveis causas, bem como sugestões de ação.

TABELA 2.1 - Problemas típicos que ocorrem em *bridges*.

Sintoma	**Causa possível**	**Sugestões de ação**
meio saturado, muitas retransmissões e sessões com *timeout*	- pacotes em *loop* e *broadcast storm*	-Examinar a topologia da rede. - Procurar possíveis *loops* e eliminá-los.

		Se *broadcast storm* e *loops* persistem, fazer uma pesquisa binária por segmentos de rede para isolá-los. - Reprojetar a rede para eliminar os qualquer *loop*. - Implementar o algoritmo de *spanning tree* para prevenir *loops*.
incapacidade de fazer conexões	- má configuração de filtros;	- Remover os filtros de *bridge* de interfaces suspeitas. Verificar se a conectividade retorna. Se não retornar, aquele filtro não é o problema, um ou mais filtros estão causando o problema de conectividade.
	- problema com a conexão física;	- Verificar o estado da linha de comunicação. Se o estado está *down*, checar a conexão física entre a interface e a rede. Se estiver *up*, verificar o meio e a conectividade de outras máquinas.
	- filas de entrada e saída cheias, devido ao tráfego excessivo de *broadcast*.	- Reduzir a quantidade de tráfego na rede implementando filtros ou segmentando a rede. Se a conexão for um enlace serial, aumentar a largura de banda, aumentar o tamanho da fila, ou modificar o tamanho do *buffer*.
sessões terminando abruptamente	- o tempo final da sessão está muito baixo	- Usar um analisador de rede para procurar retransmissões. Se retransmissões forem encontradas, aumentar o tempo de transmissão da máquina. Usar o analisador novamente para verificar se as retransmissões diminuíram.
	- retardo excessivo sobre enlace serial lento	- Aumentar a largura de banda, aplicar prioridade nas filas, aumentar o tamanho da fila ou modificar o tamanho do *buffer*.
usuários não conseguem conectar *bridges* ou roteadores	- projeto de rede ruim; endereços de rede mal configurados	- Modificar endereços que estejam incorretos.
	- roteador mal configurado	- Examinar a configuração de todas as *bridges* e roteadores da rede.
conectividade bloqueada para certas porções da rede	- endereços de rede mal configurados	- Verificar endereços de rede nas interfaces suspeitas. Ter certeza que todas as *bridges* estão no mesmo grupo ou domínio de *bridge*.
	- cabo desconectado	- Verificar as conexões físicas de todas as redes afetadas.

2.2 Detecção de erros no nível físico

Um outro problema que pode ocorrer com a rede refere-se às altas taxas de erros e colisões no seu nível físico. Se o nível físico da rede não está bom e causa problemas excessivos de erros e colisões, a comunicação com o protocolo de nível superior será prejudicada e toda a rede não poderá operar normalmente. Por exemplo, se um segmento possuir um alto número de colisões, a largura de banda será consumida e o tempo de operação normalmente usado para transferência de *frames* será usado para transmitir erros de nível mais baixo.

As seguintes condições de erros de nível físico podem causar problemas numa rede Ethernet:

- colisões;
- erros de CRC e alinhamento;
- quadros longos e curtos.

Quando analisa-se o percentual de erros do nível físico e obtém-se níveis acima de dois a três por cento, significa que algo de anormal está ocorrendo e então os quadros deste nível deveriam ser filtrados e analisados.

Embora qualquer um dos erros listados acima podem estar presentes em uma rede, estudos mostram que colisão é o que mais ocorre. Uma taxa de colisões normal deveria ser menor do que um por cento do tráfego normal da rede, e a taxa de erros global incluindo todas as condições não deveria exceder dois por cento da largura de banda disponível, [NAS 94].

Qualquer segmento de rede Ethernet terá colisões, devido às características do método de controle de acesso CSMA/CD. Se problemas de hardware estão presentes no esquema de cabeamento, a taxa de colisões pode exceder a quantidade normal, podendo prejudicar o tráfego da rede.

Abaixo são analisadas cada uma dessas condições de erro, separadamente.

2.2.1 Colisões

Uma colisão em uma rede Ethernet pode ser definida como o resultado de duas ou mais estações tentando transmitir em um mesmo meio Ethernet ao mesmo tempo. Quando a rede se torna sobrecarregada, a probabilidade de ocorrer uma colisão aumenta. Quando uma colisão ocorre, as estações que estavam transmitindo abortam suas transmissões, esperam um intervalo de tempo aleatório e tentam transmitir novamente, assumindo que nenhuma outra estação tenha começado a transmitir naquele intervalo, [TAN 89]. Colisões são eventos normais em uma rede a menos que seu nível cresça degradando o desempenho da rede. O nível aceitável de colisões depende da aplicação e protocolo que se estiver utilizando.

Colisões em uma rede podem ser de três tipos: locais, remotas ou tardias. Colisões locais acontecem quando quadros são menores que 64 bytes. Este tipo de colisão ocorre no segmento local. Colisões remotas são também quadros menores que 64 bytes, mas que são passados de outros segmentos. Isto pode indicar que uma interface de rede específica ou um *transceiver* no outro lado do repetidor está com um problema. Com a utilização de *bridges* ou roteadores colisões remotas não acontecem já que todas as colisões são isoladas em seu segmento local respectivo.

A maioria dos analisadores de protocolos podem detectar quando uma colisão não é local se os campos de endereços são endereços locais válidos. Para localizar especificamente o segmento e nodo que está gerando essa taxa é preciso verificar todos os segmentos ligados ao repetidor. Quando uma colisão local é encontrada em um segmento específico, o próximo passo é localizar o nodo ruim e então substituí-lo e reanalisá-lo.

O outro tipo de colisão é a colisão "tardia". Este tipo de colisão ocorre com quadros maiores que 64 bytes que possuem um valor corrompido no campo de CRC. Este quadro estará em um segmento local. Se uma colisão ocorre com menos de 64 bytes normais de transmissão gerados por uma estação particular, significa que há uma ocorrência de colisão normal sobre a transmissão. Contudo, se a colisão ocorre com mais de 64 bytes de dados transmitidos, ela é considerada "tardia" porque o transmissor do quadro não detectará a colisão. Colisões tardias indicam que o tempo para propagar o sinal de uma extremidade à outra da rede (ida e volta) está maior do que o tempo necessário para a transmissão do preâmbulo do quadro Ethernet (64 bytes), pois, dois componentes que causam a colisão tardia nunca detectam que outra estação está enviando dados se isto ocorre após transmitirem o preâmbulo. A causa desse tipo de colisão pode ser porque a rede está temporariamente mais longa do que é permitido pela especificação IEEE 802.3, o comprimento dos segmentos de cabos pode estar maior do que o comprimento máximo permitido para o tipo de cabo ou há um número excessivo de repetidores entre componentes de rede. Ainda outra causa de colisões tardias é provocada por *transceivers* ou placas de rede defeituosos (que transmitem em modo surdo).

Colisões excessivas em uma rede podem consumir largura de banda e causar um problema de transmissão através de toda a rede. Em tal caso, é importante que o problema de colisão seja detectado e a causa corrigida. Altas taxas de colisões na rede podem ser originadas de vários fatores, como por exemplo, rede mal configurada ou algum componente da rede transmitindo quando ele não deveria estar. A maioria das colisões são resultado de reflexões ou outro problema com a parte física da rede, tal como um *transceiver* com problema. Além disso, colisões também podem ocorrer se algum cabo estiver com um problema ou uma estação não estiver conectada corretamente na rede. A taxa de colisões está relacionada com a utilização da rede, sempre que se tem uma carga excessiva na rede a taxa de colisões aumenta.

2.2.2 Erros de CRC e alinhamento

O quadro Ethernet é responsável por conferir sua transferência de estação a estação através da verificação do campo CRC. O campo CRC é um campo de algoritmo matemático usado para a verificação de integridade do conteúdo do quadro Ethernet durante a comunicação de uma estação com outra, servindo para garantir que erros de transmissão sejam detectados. Quando uma estação recebe o quadro Ethernet, ela recalcula o conteúdo do quadro para conferir se ele combina com o valor de CRC recebido.

Às vezes, erros de CRC e alinhamento podem ser gravados como o mesmo tipo de erro. Isto acontece porque ambos indicam problema de alinhamento de bytes nas transmissões. Taxas de erros de CRC e alinhamento abaixo de um por cento do tráfego global da rede são considerados aceitáveis. Quando a taxa de colisões aumenta, é possível que o grupo de estações envolvidas em uma transferência esteja com algum problema, por isso, todos os endereços dos nodos que transmitem antes e depois do quadro com erro de CRC devem ser capturados. É possível ainda que um *transceiver*, um repetidor ou uma porta de *hub* tenham uma falha.

Quando o número de erros de CRC está alto em um segmento específico, conexões de cabos podem ter problemas. O cabeamento e conexões deveriam ser verificados primeiro. Se o problema continuar presente, a placa de rede deveria ser substituída. O conector BNC-T, o conector *tap*, e *transceivers* também podem ser a causa principal desse tipo de problema. Depois de verificar a infra-estrutura de cabeamento e não havendo problemas, deve-se substituir a placa de rede. Feito isso, deve-se reanalisar a rede para verificar se os erros continuam ocorrendo. Se o problema ainda estiver presente, repetidores deveriam também ser verificados como causa de erros de CRC. É possível atribuir esse problema a uma estação particular na rede.

Um quadro tem um erro de alinhamento se o número de bits recebidos não for múltiplo de oito; desde que há oito bits em um byte, quadros deveriam ser recebidos no limite do byte. Quadros com erros de alinhamento também têm um erro de FCS (*Frame Check Sequence*). A causa desse tipo de erro inclui problemas em instalações elétricas, em placas de rede e cabos fora da especificação.

2.2.3 Quadros longos e curtos

Todos os quadros Ethernet possuem regras quanto a seu comprimento. Dependendo do tipo de quadro eles podem ter de 64 a 1.518 bytes de comprimento. Analisando-se os quadros que trafegam pela rede pode-se distinguir dois tipos: curtos e longos. Um quadro curto é aquele com menos de 64 bytes, já o longo possui mais de 1.518 bytes. Esses tipos de quadros são considerados problemas em uma rede. Sua causa pode ser devido a problemas na placa de rede, *transceivers* ou mesmo um *driver* que controle a rede local. Quadros longos e curtos não incluem campos de endereços confiáveis. Para resolver o problema deve-se ir removendo da rede os nodos suspeitos até que se encontre o causador desses tipos de falha.

Segundo [ART 96], os quadros curtos são causados por:

- colisões, pois elas ocorrem nos primeiros 64 bytes transmitidos (uma colisão tardia causa quadros curtos se uma estação não está transmitindo de acordo com as especificações);
- rede maior que as especificações e o atraso de retorno é insuficiente para que o mecanismo CSMA/CD identifique as colisões apropriadamente;
- estação transmitiu propositadamente um quadro curto.

Os quadros longos podem ser causados por:

- uma estação não estar transmitindo de acordo com as especificações;
- *transceivers* defeituosos.

2.3 Alto nível de *broadcast*

Pacotes *broadcast* são pacotes enviados por qualquer máquina da rede e recebidos por todas as outras. Um campo de endereço no pacote especifica o destinatário (que pode ser para uma determinada sub-rede ou para toda a rede, por exemplo). Ao receber um pacote, cada máquina verifica o campo de endereço, se o pacote for destinado a outra máquina, ele é descartado, [TAN 89]. O Endereço *broadcast* é aquele onde todos os bits são 1. A maioria dos algoritmos de roteamento utilizam um endereço de *broadcast* para divulgação de rotas.

Toda a rede possui um nível de pacotes *broadcast* que é normal, necessário para manter a rede em operação, contudo alguns *broadcasts* podem causar problemas, especialmente se eles são passados de outras redes, mas não são requeridos na rede onde foram capturados. Neste caso, o software pode repassar ou enviar uma mensagem de erro quando receber pacotes não destinados a essa rede. Segundo [MAD 94], considerando-se o repasse dos pacotes recebidos deve-se levar em conta dois fatores. Primeiro, algumas implementações TCP/IP permitem que as máquinas, que operam como roteadoras, façam o repasse dos pacotes recebidos que não são destinados a eles. Segundo, mensagens com endereços de *broadcast* devem ser tratadas pelos computadores que as recebem como se fossem direcionadas para si próprios. Contudo, algumas implementações de protocolos podem não estar em conformância com os padrões estabelecidos e, portanto, não reconhecem pacotes *broadcast*. Quando isso ocorre, a estação que recebe o pacote *broadcast*, por não conhecer tal endereço, acaba repassando o pacote ou enviando uma mensagem de erro.

Em conseqüência do envio de cada mensagem *broadcast* que o originador gera, ele deve receber as respostas de cada um dos computadores que reconhece aquela mensagem como sendo um *broadcast*. Caso haja um computador que replique aquela mensagem, lançando-a novamente no segmento de rede, possivelmente um novo

conjunto de respostas daqueles computadores, que agem conforme os padrões de endereçamento, chegará artificialmente até o computador originador da mensagem. Se o segmento de rede contiver um número grande de nodos, este segundo conjunto de respostas pode saturar a rede e/ou sobrecarregar o computador que enviou a mensagem originalmente. Este fenômeno é definido como *broadcast storm*. Segundo [ART 96], existem três causas principais que provocam um *broadcast storm*: problemas nos protocolos, má configuração e implementações com defeito.

Segundo [NAS 94], como a largura de banda em qualquer rede é limitada, é aconselhável que o nível de *broadcast* seja mantido em um valor mínimo. Devido a medidas com analisadores de protocolo constatou-se que na maioria das redes Ethernet a quantidade normal desse tipo de pacote é entre oito e dez por cento. Problemas tais como *broadcast storm* podem ocorrer quando esse percentual aumentar atingindo de vinte a vinte e cinco por cento do tráfego total. É preciso levar em consideração que, às vezes, o número de pacotes *broadcast* pode aumentar, não porque houve um problema, mas porque um usuário está entrando ou saindo de uma estação da rede.

O perigo real aparece quando redes que estão ligadas através de *bridges* experimentam um alto nível de *broadcast*. O perigo existe porque pacotes de *broadcast* podem atravessar a *bridge* e afetar outras redes.

2.4 Alto número de retransmissões

Em uma rede que tem um serviço baseado em conexão confiável precisa haver uma forma para se ter certeza que todos os pacotes enviados foram recebidos corretamente no destino. A maneira usual de assegurar uma distribuição confiável na rede é fornecer ao transmissor algum *feedback* sobre o que está acontecendo na outra extremidade da linha, [TAN 89]. Geralmente, o protocolo pede ao receptor para enviar de volta quadros de controle especiais, portanto confirmações positivas ou negativas são enviadas. Se o transmissor receber uma confirmação positiva sobre um quadro, ele sabe que o quadro chegou com segurança. Por outro lado, uma confirmação negativa significa que algo saiu errado e o quadro deve ser retransmitido.

O problema de retransmissões excessivas pode causar um alto nível de tráfego do nível físico ao de rede, eventualmente causando uma perturbação geral em toda a comunicação e operação da rede. Se esta situação está ocorrendo no nível de aplicação, a aplicação particular pode, às vezes, cair e possivelmente causar uma falha de rede no nível de servidor de arquivo. Nesse caso seria bom fazer uma monitoração pró-ativa para que o problema de retransmissão fosse detectado antes de causar tantos danos.

Algumas aplicações e certos tipos de módulos de softwares permitem modificar a configuração para prevenir retransmissões na rede. O objetivo de analisar retransmissões é para eliminá-las completamente da rede.

Se retransmissões ocorrem muito freqüentemente, é possível que existam problemas no nível físico ou algum dos protocolos de nível superior esteja tendo problemas em receber ou transmitir dados.

2.5 Mensagens ICMP de Erro e Controle

O protocolo IP fornece um serviço de entrega de datagramas não orientado à conexão e inseguro. Os datagramas trafegam pela rede passando de *gateway* para *gateway* até que um *gateway* possa entregá-lo ao endereço IP destino. Conforme [COM 91], se um *gateway* não puder rotear ou entregar um datagrama ou se ele detectar uma condição não usual, como por exemplo congestionamento na rede, que afeta sua capacidade para transmitir o datagrama, ele precisará instruir a máquina origem que causou aquele datagrama para evitar ou corrigir o problema.

O serviço não orientado à conexão trabalha bem se todas as máquinas operarem corretamente, mas o problema é que nenhum sistema trabalha corretamente o tempo todo. Além de falhas das linhas de comunicação e processadores, há também falhas na entrega de datagramas. Esse tipo de falha pode ocorrer quando a máquina destino estiver temporariamente ou permanentemente desconectada da rede, quando o contador *time to live* (TTL) expirar ou quando *gateways* intermediários se tornarem tão congestionados que eles não conseguirão processar o tráfego que chega. Quando essas falhas ocorrem, a máquina que enviou não pode dizer se a falha resultou de um problema local ou remoto. O protocolo IP não tem como ajudar a máquina que enviou a testar a conectividade ou aprender sobre tais falhas. Portanto, para permitir que os *gateways* na Internet pudessem informar sobre esses erros ou sobre circunstâncias não esperadas, os projetistas adicionaram um mecanismo de mensagens especiais para o protocolo TCP/IP. O mecanismo, conhecido como *Internet Control Message Protocol* (ICMP), é considerado uma parte do IP e deve ser incluído em todas suas implementações.

O ICMP é encapsulado pelo datagrama IP, portanto essas mensagens trafegam pela rede na porção de dados dos datagramas. Além disso, ele não é restrito para *gateways*: uma máquina arbitrária pode enviar uma mensagem ICMP para qualquer outra máquina, embora haja restrições para algumas mensagens. Tecnicamente, ICMP somente relata condições de erro para a máquina origem; a máquina origem deve então passar estes erros para programas de aplicação que tomam a ação para corrigir o problema. O ICMP não especifica a ação que deve ser tomada para corrigir cada erro encontrado.

Se uma rede possuir um alto número de mensagens ICMP seu desempenho pode ser prejudicado. Mensagens ICMP são enviadas em várias situações, [POS 81]: quando um datagrama não pode alcançar seu destino, quando o *gateway* não tem capacidade de armazenamento para "seguir adiante" um datagrama, quando o *gateway* pode instruir o *host* a enviar seu tráfego por um caminho mais curto, entre outros. O propósito dessas mensagens de controle é fornecer um *feedback* sobre problemas no ambiente da comunicação.

Cerca de uma dezena de mensagens ICMP estão definidas. Essas mensagens, utilizadas para localizar problemas e para controle, estão descritas abaixo:

- ***destination unreachable***: esta mensagem é usada quando um *gateway* não consegue enviar um datagrama IP. Geralmente, redes não alcançáveis são devido a falhas de roteamento. Segundo [COM 91], destinos podem estar não alcançáveis por problemas no hardware, porque a origem especificou um endereço destino que não existe ou porque o *gateway* não tem uma rota para a rede destino. Além disso, esse tipo de mensagem também será enviado quando um datagrama precisar ser fragmentado mas seu *flag "Don't Fragment"* estiver setado.

- ***time exceded***: esta mensagem indica que o TTL de determinado datagrama chegou a zero e que o datagrama não foi entregue ao destino. Como *gateways* computam o próximo *hop* usando tabelas locais, erros em tabelas de roteamento podem produzir *loops*. Para evitar que datagramas fiquem em *loop* indefinidamente, o *gateway* decrementa o contador TTL quando ele processa o datagrama e descarta-o quando este contador alcançou o valor zero. Datagramas também podem ser descartados sempre que ocorrer um *timeout* quando o *gateway* estiver esperando por fragmentos de um datagrama. Nesses dois casos são enviadas mensagens ICMP de *time exceded.*

- ***parameter problem***: esta mensagem é enviada sempre que um *gateway* ou *host* encontrar parâmetros incorretos no cabeçalho de um datagrama. Uma possível causa de tais problemas ocorre quando argumentos de uma opção estão incorretos. Um campo ponteiro é usado nesta mensagem para identificar o octeto do datagrama que causou o problema.

- ***source quench***: esta mensagem é um pedido para a origem reduzir sua taxa de transmissão de datagramas. Sempre que um *gateway* torna-se sobrecarregado, não conseguindo processar todos os datagramas que estão chegando, ele deve descartá-los. É importante entender que este tráfego pode aumentar por duas razões diferentes. Primeiro, um computador de alta velocidade pode ser capaz de gerar tráfego mais rápido do que uma rede pode transferi-lo. Segundo, se muitos computadores precisam simultaneamente enviar datagramas por um único *gateway*, o *gateway* pode tornar-se congestionado. Um *gateway* ou um *host* enfileiram os datagramas em uma memória temporária quando os datagramas chegam muito rapidamente. Se o tráfego é contínuo, a memória se esgota e novos datagramas que chegarem serão descartados. Neste caso, mensagens ICMP de *source quenchs* serão enviadas para cada datagrama descartado. Não há uma mensagem para reverter o efeito de um *source quench*, então a máquina origem somente volta a sua taxa de transferência normal quando ela parar de receber esse tipo de mensagem.

- ***redirect***: esta mensagem é um pedido para um *host* mudar sua rota para determinada rede. Tabelas de roteamento de *hosts* se mantém estáticas por longos períodos de tempo. Se a topologia da rede muda, tabelas de roteamento de *gateways* e *hosts* podem tornar-se incorretas, mas como *gateways* trocam informações de roteamento periodicamente, as rotas se tornarão atualizadas novamente. Em um caso especial, quando um *gateway* detecta um *host* usando uma rota não-ótima, ele envia uma mensagem ICMP ao *host*, chamado um *redirect*, pedindo ao *host* para trocar sua rota. A vantagem deste esquema é a simplicidade.
- ***echo request*** e ***echo reply***: esse tipo de mensagem é utilizado para descobrir se uma estação está viva e testar o caminho de comunicação entre duas estações. A mensagem de e*cho* é comumente conhecida como *ping*. Uma máquina envia uma mensagem ICMP de *echo request* para um destino específico. Quando a estação estiver viva e não houver problemas na comunicação, ela retornará uma mensagem de *echo reply* para cada mensagem de *echo* recebida. O *echo request* associado com o *echo reply* podem ser usados para testar se um destino está alcançável e respondendo.
- ***timestamp request*** e ***timestamp reply***: esta mensagem é usada para obter a hora de uma outra máquina. Conforme [COM 91], as máquinas na Internet se comunicam, mas operam independentemente, isto é, cada máquina mantém sua própria notação de hora corrente, o que pode causar confusão quando relógios diferem muito. Para cada mensagem de *timestamp request* recebida uma mensagem de *timestamp reply* é enviada. As máquinas utilizam três campos dessa mensagem (*originate timestamp*, *receive timestamp* e *transmit timestamp*) para calcular estimativas da hora do dia entre elas e para sincronizar seus relógios. Como a resposta inclui o campo *timestamp* do originador, uma máquina pode calcular o tempo total requerido para um pedido viajar até o destino, ser transformado em uma resposta e ser retornado. Além disso, como a resposta carrega a hora em que o pedido entrou na máquina remota, bem como a hora em que a resposta saiu, a máquina pode calcular o tempo de transmissão na rede e estimar as diferenças de relógio entre a máquina local e a remota.
- ***address mask request*** e ***address mask reply***: esta mensagem é usada por um *host* para aprender a máscara da sub-rede da rede local. Um *host* envia uma mensagem de *address mask request* para um *gateway* e recebe um *address mask reply*. O *host* pode enviar o pedido diretamente para o *gateway*, se ele souber seu endereço ou através de uma mensagem de *broadcast* caso ele não saiba.

3 Facilidades de gerência inerentes aos equipamentos a serem gerenciados

Um passo essencial para alcançar os objetivos do gerenciamento de uma rede é adquirir informações sobre a rede. Foi com este propósito que um conjunto padronizado de protocolos de gerenciamento foram desenvolvidos. Esses protocolos ajudam a extrair as informações necessárias de todos os componentes de uma rede.

O gerenciamento de uma rede depende do que se pode monitorar e controlar sobre a informação coletada, porque sem isso a pessoa que administra seria forçada a tomar decisões de gerenciamento sem poder adequar medidas qualitativas e quantitativas. Portanto, é essencial entender os métodos disponíveis para computação e controle dos dados na rede. Até alguns anos atrás, para coletar informações de dispositivos de redes diferentes era preciso aprender uma variedade de métodos pelos quais se podia obter os dados. A medida que novos produtos de rede eram desenvolvidos, seus fabricantes instalavam métodos proprietários para coletar dados de seus produtos. O resultado era que dois dispositivos com a mesma funcionalidade, mas vindos de fabricantes diferentes, poderiam fornecer métodos distintos para coletar os dados.

Com o passar do tempo, houve a necessidade de se criar um sistema padrão para esse propósito. Conseqüentemente, a comunidade de redes desenvolveu duas tecnologias diferentes projetadas especificamente para o gerenciamento de redes. O primeiro é o SNMP (*Simple Network Management Protocol*), desenvolvido pela comunidade Internet, e o segundo é o CMIS/CMIP (*Common Management Information Services / Common Management Information Protocol*), padrão ISO. Ambos protocolos fornecem uma forma de obter informações ou dar instruções para componentes de uma rede. Com a utilização desses protocolos, além de monitorar estações de trabalho, também podem ser monitorados *hubs*, *bridges*, roteadores, e demais componentes de uma rede.

Com protocolos de gerenciamento de rede, pode-se acessar qualquer dispositivo da rede de uma maneira uniforme. Consultas a dispositivos de rede podem incluir:

- seu nome;
- a versão do software utilizado;
- o número de interfaces presentes;
- o número de pacotes por segundo que passam (entram/saem) em sua interface.

Além disso, podem ser configurados, para cada dispositivo, os seguintes parâmetros:

- nome do dispositivo;
- endereço da interface de rede;
- estado operacional da interface de rede;
- estado operacional do dispositivo.

Finalmente, fornecedores da indústria de comunicações de dados têm desenvolvido sistemas de gerenciamento de rede que incorporam protocolos padrão (CMIP, CMOT, SNMP, etc) dentro de arquiteturas proprietárias.

Protocolos de gerenciamento de rede padronizados possuem um benefício adicional no qual os dados enviados e retornados pelo dispositivo possuem um formato uniforme.

3.1 Modelo do Gerenciamento Internet

Atualmente o protocolo não-proprietário correntemente em uso na Internet para transportar informações e operações de gerenciamento é o SNMP, enquanto as regras para definição de informações de gerenciamento são chamadas *Structure of Management Information* (SMI) e a coleção de informações de gerenciamento é chamada de *Management Information Base* (MIB).

Em um processo de coleta de dados é necessário adotar padrões para que diversos usuários destes dados possam falar a mesma linguagem. A forma de criação de um item de dado segue o padrão SMI que consiste em um conjunto de regras a ser obedecido para definir e identificar variáveis da MIB (formada pela coleção de informações de gerenciamento). Assim, o SMI diz "como fazer" e a MIB, juntamente com o protocolo de gerenciamento, diz "o que fazer" [SOU 95].

A representação das informações de gerenciamento utilizadas pelo SNMP é representada de acordo com um subconjunto da linguagem ASN.1 (*Abstract Syntax Notation One*), linguagem formal padronizada pela ISO que é especificada para a definição de tipos não-agregados do SMI. O SNMP utiliza esta linguagem para a descrição dos objetos gerenciados e para a descrição das unidades de dados do protocolo usadas para gerenciar aqueles objetos.

A linguagem ASN.1 também é utilizada para descrever as PDUs (*Protocol Data Unit*) [BRO 93]. Essa linguagem foi criada para facilitar uma eventual migração para protocolos de gerência padrão-OSI.

A utilização da ASN.1 é particularmente importante quando se tem sistemas computacionais heterogêneos, que, normalmente, utilizam diferentes representações

para as variáveis consideradas. Com a ASN.1 tem-se uma documentação sem ambigüidades, garantindo uma implementação facilitada dos protocolos de gerenciamento de redes ou de qualquer protocolo de aplicação.

Nodos gerenciados podem ser estações de trabalho, roteadores e servidores de terminal. Tais dispositivos tem agentes de gerenciamento responsáveis por executar as funções de gerenciamento de rede requisitadas pela estação de gerenciamento.

Um nodo gerenciado pode ser utilizado como uma estação de gerenciamento e, então, exercer as duas funções: de agente e gerente ao mesmo tempo. O modelo agente-gerente pode ser visto como um modelo ponto-a-ponto, onde cada nodo pode consultar outros nodos. Com isso em mente, pode-se construir relacionamentos hierárquicos entre estações de gerenciamento. Por exemplo, alguém poderia imaginar a construção de um sistema de gerenciamento no qual cada segmento da rede local tivesse uma aplicação de gerenciamento que conhecesse o estado dos dispositivos daquele segmento. Essas aplicações de gerenciamento poderiam passar o seu conhecimento para aplicações que estivessem rodando em estações de gerenciamento "regionais", e assim estas poderiam reportar para aplicações rodando em estações de gerenciamento que estivessem mais distantes.

Nesse exemplo, o software em cada estação de gerenciamento realiza um papel de gerente quando é feita a monitoração e controle dos dispositivos que estão subordinados na hierarquia, e um papel de agente quando transfere informações e age em cima de comandos dados por uma estação superior na hierarquia, [ROS 95].

A comunicação entre estação de gerenciamento e nodos gerenciados é feita principalmente através de *pollings*, embora a facilidade baseada em alertas também esteja disponível, [SNG 90]. Esta comunicação através de *pollings* funciona da seguinte forma: a estação de gerência envia periodicamente para a entidade gerenciada uma solicitação de informações relativas àquela entidade. Após a recepção, a entidade gerenciada responde a solicitação com os dados desejados.

Outro elemento presente na arquitetura Internet é o agente procurador (*proxy agent*), que traduz as requisições do gerente para a linguagem entendida pelos nodos gerenciados, [SIL 95]. Quando as redes são heterogêneas e alguns equipamentos não suportam a implementação do gerente nos moldes do padrão Internet, o agente procurador é utilizado, tal como ilustra a figura 3.1. Adicionalmente, pode ocorrer que o dispositivo gerenciado utilize um protocolo proprietário para comunicação entre agentes e gerentes, e então usa-se um agente procurador, tal como ilustrado na figura 3.2, para tornar possível o gerenciamento desta classe de dispositivos por um gerente padrão.

Agentes procuradores podem ser integrados a dispositivos que necessitam de funcionalidade de gerenciamento SNMP. Isto é alcançado pelo interfaceamento da porta de gerenciamento proprietária do dispositivo e da conversão de alertas e informações de estado para um formato de acordo com o SNMP. Dispositivos tais como multiplexadores, PABX's, entre outros podem ser fornecidos com funcionalidade SNMP. O gerente de rede SNMP também pode receber notificação imediata de eventos, como por exemplo multiplexadores perdendo sinal de WAN, através do uso de um agente *proxy*.

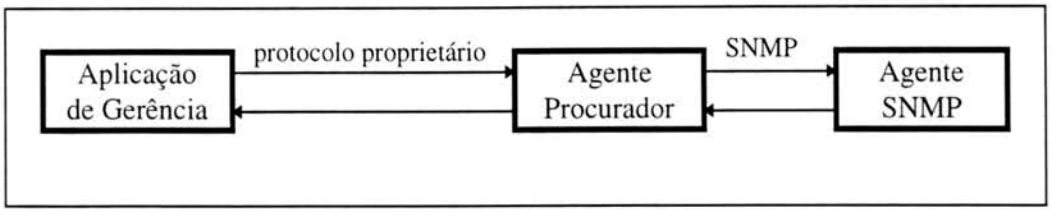

FIGURA 3.1 - Aplicação de gerência utilizando protocolo proprietário.

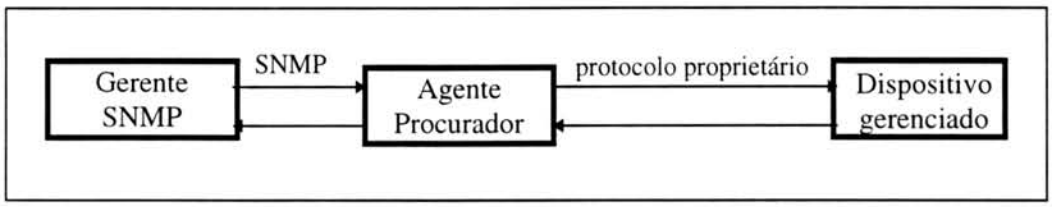

FIGURA 3.2 - Dispositivo gerenciado utilizando protocolo proprietário.

3.1.1 *Management Information Base* (MIB)

A MIB (*Management Information Base*) é considerada o repositório de informações para gerenciar redes padrão Internet. Ela contém uma coleção de objetos que podem ser acessados via um protocolo de gerenciamento de rede. Inicialmente foi desenvolvida a MIB I, definida no RFC 1158, que apresentava uma lista reduzida de objetos que poderiam ser usados para gerenciar uma rede, sendo mais tarde substituída pela MIB II.

A MIB II, especificada no RFC 1213 [McC 91], define as variáveis necessárias para a monitoração e controle dos vários elementos da Internet. Essas variáveis são agrupadas em pequenos grupos, dependendo da camada ou protocolo que será monitorado. Nem todos os grupos de variáveis são obrigatórios para todos os elementos da Internet. Por exemplo, o grupo TCP é obrigatório para as máquinas que rodam TCP, mas não para roteadores que não o utilizam. Contudo, se qualquer membro de um grupo de variáveis for suportado, então é obrigatório que todos os membros daquele grupo sejam suportados.

Apesar de haver uma MIB padrão para redes Internet, é possível adicionar extensões a ela para modelar características de organizações ou de projetos. É possível, então, definir objetos privados na sub-árvore *enterprises* da MIB. Esses objetos adicionais podem ser variáveis específicas de algum fornecedor ou para propósitos de experimentação.

As informações na MIB são armazenadas em uma base de dados estruturada como árvore. A raiz da árvore define três organizações às quais são associadas as informações gerenciais. Para alcançar um determinado objeto é preciso percorrer toda a

árvore. Cada nodo da árvore consiste de um número e associado a ele está um nome simbólico.

Os nomes para todos os tipos de objetos são definidos explicitamente na MIB padrão Internet ou em outros documentos que fazem as convenções de nomes para o SMI.

O conceito de MIB não impõe nenhuma condição ao nível de normalização de sua estrutura interna ou a nível de organização local de uma base de informações. O essencial é que todo o sistema seja capaz de identificar corretamente os objetos constituintes da MIB. Em outras palavras, a MIB é o repositório conceitual de todos os objetos gerenciados, não importando qual seja o meio para armazenamento físico das informações de gerenciamento, [SOU 95].

A MIB foi definida como uma coleção de objetos gerenciados. Aspectos importantes são a definição de uma estrutura lógica, a compreensão das ações a serem executadas sobre estes objetos gerenciados, bem como os eventos que eles podem gerar.

A seguir estão listados os grupos que compõem a MIB II:

- System
- Interfaces
- Address translation
- IP
- ICMP
- TCP
- UDP
- EGP
- Transmission
- SNMP

Atualmente existe uma infinidade de MIBs para os mais diversos dispositivos de rede. Existem MIBs para roteadores, *bridges*, *switches*, *hubs*, Windows NT, Windows 95, Rede Novell, entre outras. MIBs públicas, tal como a usada por repetidores [McM 93], estão descritas em RFC's. Também existem MIBs para diferentes tipos de tecnologias de redes, tais como Token Ring, FDDI e ATM. Todas essas MIBs possuem objetos gerenciados com os quais pode-se monitorar interfaces e portas dos dispositivos.

Mais adiante encontra-se descrita a RMON MIB, a Host MIB, a MIB da Novell e a MIB da Cisco. Como o objetivo deste trabalho é realizar a gerência de desempenho e falhas na rede da UFRGS, que utiliza o protocolo TCP/IP, optou-se por utilizar a MIB II.

3.1.2 O Protocolo SNMP

O SNMP (*Simple Network Management Protocol*), especificado no RFC 1157 [CAS 90], define um protocolo simples, no qual usuários remotos logicamente podem gerenciar elementos de rede que sejam baseados em redes TCP/IP, e em particular na Internet.

O SNMP permite a monitoração e controle do nodo gerenciado e provê uma estrutura administrativa que deve implementar as políticas de autenticação de mensagens e de autorização. Através do protocolo de gerenciamento de rede SNMP é possível inspecionar ou alterar variáveis de uma MIB que esteja em algum agente, utilizando para tanto um conjunto de operações disponíveis.

Atualmente existe o protocolo SNMP versão 2 que seria uma melhoria do SNMP versão 1. O SNMPv2, como é chamado, busca corrigir as deficiências funcionais do SNMPv1. Suas principais inovações localizam-se nos seguintes pontos:

- estrutura da informação de gerenciamento (a SMI do SNMPv2 foi expandida para incluir novos tipos de dados e melhorar a documentação);
- operações do protocolo (duas novas operações foram incluídas);
- comunicação gerente-a-gerente (suportada por uma nova base de informações, a M2M MIB); e,
- segurança (SNMPv2 incorpora as especificações contidas no conjunto de documentos conhecido como S-SNMP (*Secure* SNMP)).

3.1.2.1 Operações Suportadas pelo SNMP

O SNMP trata todas as operações de gerenciamento como alterações ou consultas das variáveis (objetos). Esta característica limita o número de funções de gerenciamento e evita a definição de novos comandos, [BRO 93]. Este é um protocolo de *request/response*, figura 3.3, sendo que cada requisição consiste de:

- operação: ***get***, ***get-next***, ***get-bulk*** (somente disponível no SNMPv2) ou ***set***;
- identificação da requisição: um valor inteiro usado pela aplicação de gerenciamento para distinguir entre aplicações pendentes; e,
- lista de variáveis, cada uma contendo um nome e um valor.

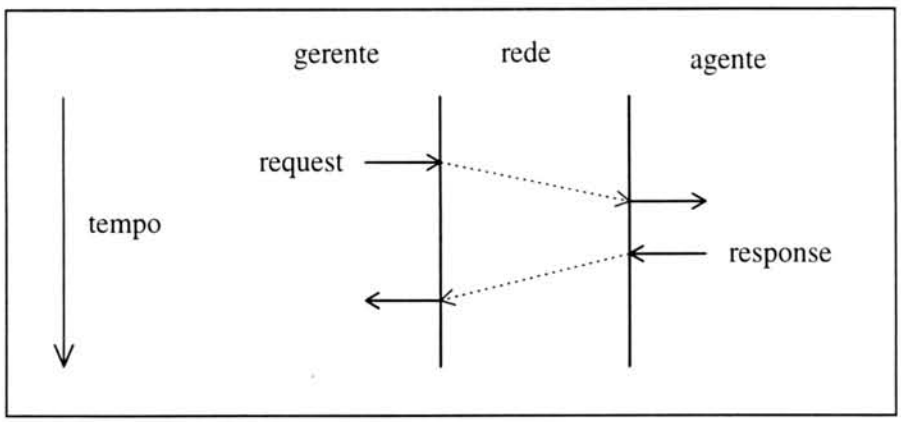

FIGURA 3.3 - Funcionamento do SNMP.

Com operações de ***get***, ***get-next*** e ***get-bulk***, pode-se recuperar informações. Quando se desejar modificar o valor de uma variável, utiliza-se a operação ***set***, e para operações de notificação quando o agente detectar que ocorreu uma condição extraordinária na rede, a operação ***trap*** é utilizada. Esta última é uma operação não solicitada pelo usuário e informa sobre a ocorrência de um evento específico.

Segundo [ROS 90], o SNMP é um protocolo assíncrono, ou seja, uma entidade SNMP não precisa aguardar por uma resposta antes de enviar outra mensagem. É permitido a ela enviar mensagens ou fazer outras atividades ao mesmo tempo.

Com efeito, SNMP utiliza o princípio de leitura de variáveis a distância (*polling*): cada máquina, onde se situa um agente, coordena um certo número de objetos gerenciados, os quais são acessados periodicamente pelo protocolo para leitura (monitoração) ou escrita (alteração).

Quando um gerente desejar consultar ou alterar alguma variável, ele envia uma requisição para o agente (nodo gerenciado), com a operação que deve ser feita. O agente então processa a requisição e envia uma resposta para o gerente. Além da identificação da requisição, uma resposta é formada por:

- *error-status*: se for um valor não zero indica que um erro ocorreu quando o agente estava processando a requisição, e que o campo com a lista de variáveis deveria ser ignorado.
- *error-index*: se for um valor não zero, indica que uma variável na requisição estava errada.
- lista com os pares nome-valor das variáveis, atualizadas pelo sucesso das operações ***get***, ***get-next*** e ***get-bulk***.

Enquanto o agente estiver processando uma requisição, um erro poderá ser encontrado, indicando que uma operação não pode ser processada. Se um erro ocorrer, a

resposta terá um valor não zero no campo *error-status*, assim como no campo *error-index*. Uma requisição também pode ser enviada para um nodo gerenciado, mas uma resposta nunca ser recebida. Isto pode ser devido a:

- a rede perdeu a requisição;
- o agente não está rodando;
- o agente não gera uma resposta para a requisição;
- a rede perdeu a resposta; ou,
- o *timeout* da aplicação de gerenciamento estava muito pequeno.

Não é possível mudar a estrutura de uma MIB utilizando as operações do SNMP. Além disso, o acesso é dado somente para objetos folha da árvore identificadora de objetos. Entretanto, por convenção, é possível executar operações em tabelas bi-dimensionais simples. Estas restrições simplificam grandemente a implementação do SNMP, mas, por outro lado, impõem limites na capacidade do sistema de gerenciamento de rede.

A comunicação de informação de gerenciamento entre o sistema de gerenciamento de rede e os elementos de rede é baseada na troca de mensagens, que requerem um serviço de datagrama não confiável. Para o SNMP o serviço de transporte sem conexão é especificado no protocolo. A primeira questão que leva a esta escolha é a necessidade que o SNMP tem em continuar operando mesmo quando houver problemas de desempenho na rede. Para outras aplicações, tais como TELNET e FTP, o usuário sempre pode "tentar mais tarde", mas em aplicações de gerência acontece justamente o contrário, pois são nos piores momentos da rede que o protocolo de gerência deve reportar os seus problemas e gargalos.

Como o SNMP é utilizado sobre as camadas de rede e de transporte é possível que a gerência de uma rede continue a operar mesmo que hajam falhas no roteamento. O nível de rede provê funções de roteamento, facilitando a escolha de uma nova rota no caso de uma falha.

3.2 Agentes disponíveis

Como o padrão do gerenciamento Internet é usar SNMP, hoje em dia a maioria dos equipamentos podem ser monitorados por esse protocolo, bastando para tanto que haja um agente SNMP instalado no equipamento. Além disso, podem existir MIBs diferentes para equipamentos diferentes, sendo elas públicas ou privadas. Para que o gerente consiga gerenciar esses equipamentos adequadamente, é necessário que ele conheça os objetos a serem gerenciados.

SNMP é um padrão de fato para gerenciamento de redes e existe no mercado um grande conjunto de sistemas de gerenciamento altamente sofisticados e amigáveis ao

usuário que utilizam esse padrão. Portanto, pode-se pretender diagnosticar a maior parte dos problemas interrogando agentes instalados nos equipamentos, softwares, etc.

Abaixo estão relacionados alguns exemplos de agentes específicos para determinados sistemas. Os motivos que levaram à escolha destes agentes foram:

- nas sessões 3.2.1, 3.2.2 e 3.2.3, o fato de serem os mais populares e utilizados tipos de agentes na rede;
- na seção 3.2.4, o agente foi incluído para exemplificar a abertura do modelo que permite o trabalho mesmo com sensores bastante simples como é o caso;
- as sessões 3.2.5 e 3.2.6 foram incluídos porque são exemplos de situações hoje existentes de uso de agente a nível de aplicação.

3.2.1 Agente de Servidor Novell

O gerenciamento de redes através do protocolo SNMP não é somente utilizado por redes Internet, um exemplo disso pode ser verificado em agentes disponíveis para gerenciar redes Novell. Esses agentes se localizam em servidores NetWare e podem fornecer estatísticas e notificar o administrador através de um alarme quando algum erro acontece. Além disso, é possível monitorar, manter e gerenciar todos os servidores NetWare do *site* local. Dependendo da implementação utilizada, uma aplicação gráfica pode estar disponível para o gerenciamento, utilizando, geralmente, o ambiente MS Windows.

Instalando uma cópia do software agente em cada servidor NetWare, pode-se monitorar, manter e otimizar o desempenho em um ambiente distribuído através de uma localização centralizada. Tendo disponível uma interface gráfica, pode-se recuperar informações estatísticas em formato gráfico sobre a configuração do servidor, utilização de CPU e alocação de memória. É possível coletar informações sobre configuração das interfaces de rede e tipos de protocolos e *frames* usados. A utilização de agentes em redes Novell permite ao administrador examinar remotamente discos de suas estações clientes, utilizando portanto um agente RMON. Desta forma, pode-se obter as partições físicas dos discos, o tamanho do volume lógico, espaços utilizados e espaços disponíveis de cada um. Uma vez o agente de gerenciamento NetWare tenha sido instalado, o servidor correspondente pode ser gerenciado por múltiplos suportes em múltiplas localizações.

A MIB da Novell, projetada para o protocolo IPX, está descrita na seção 3.3.2.

3.2.2 Agente UNIX

O Sistema Operacional UNIX possui um agente de gerenciamento SNMP, que fornece informações vitais sobre configuração do sistema, estado, desempenho, usuários, aplicações, sistemas de arquivos, processos, entre outros. Com isso, um

gerente remoto pode consolidar o gerenciamento e permitir ao administrador gerenciar centenas de sistemas UNIX de um único local.

Utilizando agentes em ambientes UNIX, pode-se monitorar variáveis da MIB, verificando condições e exceções e notificando o gerente via operações de *trap* quando estas exceções ocorrem. Através de monitorações pode-se criar arquivos de *log* e compará-lo com expressões regulares para detectar problemas.

3.2.3 Agente para Repetidor IEEE 802.3

Assim como roteadores que podem ser gerenciados através de agentes SNMP e podem possuir MIBs específicas dependendo do fornecedor, também existem repetidores que suportam agentes SNMP para serem gerenciados. Durante sua definição, para que o gerenciamento fosse dependente das características dos repetidores, foi definida, em um RFC [McM 93], uma MIB exclusiva para esses equipamentos.

Um repetidor IEEE 802.3 conecta segmentos Ethernet para estender o comprimento da rede. Todas as estações em um segmento conectadas a uma porta específica do repetidor participam de um único domínio de colisão. Um pacote transmitido por qualquer uma destas estações é visto por todo segmento.

A função do repetidor é retransmitir os dados à medida que eles são recebidos. Os dados que chegam em uma porta do repetidor são transmitidos por qualquer outra porta. Um repetidor é conectado na rede com MAUs (*Medium Attachment Units*), também conhecidos como *transceivers*, e às vezes através de AUIs (*Attachment Unit Interfaces*).

O esboço do gerenciamento desses repetidores define sete funções e sete sinais usados para descrever precisamente quando contadores de porta são incrementados. Além disso, foram definidos três grupos para fazer parte dessa MIB. O **grupo básico** contém objetos que são aplicáveis para todos os repetidores. Ele contém objetos de estado, parâmetro e controle para repetidores como um todo. O **grupo monitor** é um grupo opcional que contém estatísticas de monitoração para o repetidor como um todo ou para portas individuais. O último grupo é o de **rastreamento de endereço**. Este grupo também é opcional e contém objetos para descobrir os endereços MAC dos DTEs conectados nas portas do repetidor.

3.2.4 Agente para gerenciamento de temperatura e umidade

O agente SNMP da NMT (*Network Management Technologies*) para Temperatura e Umidade permite que as condições do ambiente sejam gravadas em locais remotos e monitoradas na rede, [NMT 96]. O equipamento que suporta esse agente é chamado de THOO1 e foi projetado para integrar facilmente plataformas de gerenciamento de rede, tais como HP Openview e Cabletron Spectrum. Sensores de temperatura e umidade em ambientes industriais ou salas de computadores podem ser monitorados via SNMP.

Através de operações de *get* e *get-next* do SNMP, o gerente de rede pode consultar o agente para verificar a temperatura e umidade corrente. A maioria das plataformas de gerenciamento podem ser configuradas para alertar um operador quando essas leituras alcançarem valores críticos.

Este agente inclui as seguintes características:

- suporte a MIB II padrão;
- *Trap ColdStart* (*trap* que indica quando o agente é reinicializado);
- *Traps* específicos de empresas, configuráveis pelo usuário;
- resposta de ***ping***;
- MIBs privadas extensivas.

Para permitir flexibilidade e fornecer um custo efetivo, o TH001 é provido com uma interface de terminal. A interface se conecta a um *laptop* rodando um programa de terminal, como por exemplo o programa de Terminal do Windows. Durante a fase de instalação o usuário conecta seu terminal a uma porta de TH001 via um cabo *crossover* RS232. O TH001 permite que o usuário configure os seguintes parâmetros, que estão armazenados em uma memória não volátil:

⇒ Detalhes do IP

- endereçamento IP
- máscara de sub-rede

⇒ Configuração de detalhes do SNMP

- nome da comunidade
- endereço IP de *trap*

⇒ Fornecimento das entradas do sistema na MIB II

- *sysDescr*
- *sysContact*
- *sysName*
- *sysLocation*

O TH001 é facilmente integrado com qualquer gerente SNMPv1.

3.2.5 Gerência de Correio Eletrônico

O trabalho realizado por [SIL 95] propõe gerenciar uma das entidades funcionais do *Message Handling System* (MHS) X.400. O MHS X.400 é a aplicação de correio eletrônico para o ambiente OSI. As fases de definição e levantamento dos requisitos para gerência de um Agente de Transferência de Mensagens (ATM) basearam-se nos modelos organizacional, funcional e informacional definidos pela OSI.

O modelo organizacional define a hierarquia entre as entidades de gerência, ou seja, a hierarquia entre agentes e gerentes. O funcional descreve os requisitos para gerência ATM segundo as áreas funcionais do modelo: gerência de falhas, de configuração, de contabilização, de desempenho e de segurança. O modelo informacional define a MIB, ou seja, os objetos que serão monitorados.

O MHS X.400 é uma aplicação do nível sete do Modelo de Referência OSI e foi projetada para satisfazer as exigências de um serviço de correio eletrônico completo, contendo: serviço de transporte confiável, mecanismo para armazenar as mensagens, segurança na transmissão das mensagens, interoperabilidade entre sistemas, utilização do serviço de diretório e serviço de notificação sobre sucesso ou falha de entrega de mensagens.

Embora a aplicação de gerência seja constituída usando a arquitetura OSI, o protocolo de gerenciamento adotado foi o SNMP, utilizando o agente SNMP do ISODE (*ISO Development Environment*).

Para que a gerência pudesse ser efetivada, foi definida uma MIB contendo vários objetos. Esses objetos foram definidos com base nos requisitos de gerência do modelo funcional do OSI e portanto possuem informações de configuração, controle de fluxo, controle de falhas, controle de mensagens, testes e relatórios, associações e históricos. Além dos objetos da MIB, há a geração de arquivos de *log* e a implementação de *traps* específicos.

Para avaliar o sistema, foi feita uma implementação no ambiente TCP/IP da UFRGS, utilizando o ambiente ISODEv8.0, o agente SNMPv1 do ISODE e o PPv6.0, o qual implementa um ATM que suporta vários protocolos de transferência de mensagens, entre eles o X.400 P1. Como gerente da aplicação, foi utilizado o software SunNet Manager. Devido ao agente SNMP e gerente possuírem domínios de gerência diferentes, a comunicação entre os dois foi feita através de um agente procurador que interage com o SNMP.

Foi usado o gerenciador SunNet Manager como uma interface gráfica para a interação com o usuário, bem como para a visualização do *layout* do sistema, uma ferramenta gráfica para visualização dos dados resultantes da gerência, *browser* para percorrer a MIB, métodos de geração de avisos, entre outros, [SIL 95].

O sistema funciona da seguinte forma: gerentes são ativados pelo usuário através da interface gráfica do SunNet Manager, figura 3.4. Então escolhe-se a máquina onde o ATM está instalado, bem como qual gerente será executado (o SunNet Manager oferece três tipos de gerentes). Feito isso, deve-se escolher os agentes que serão

monitorados naquela máquina. A escolha do agente implica em escolher quais objetos daquele agente serão monitorados.

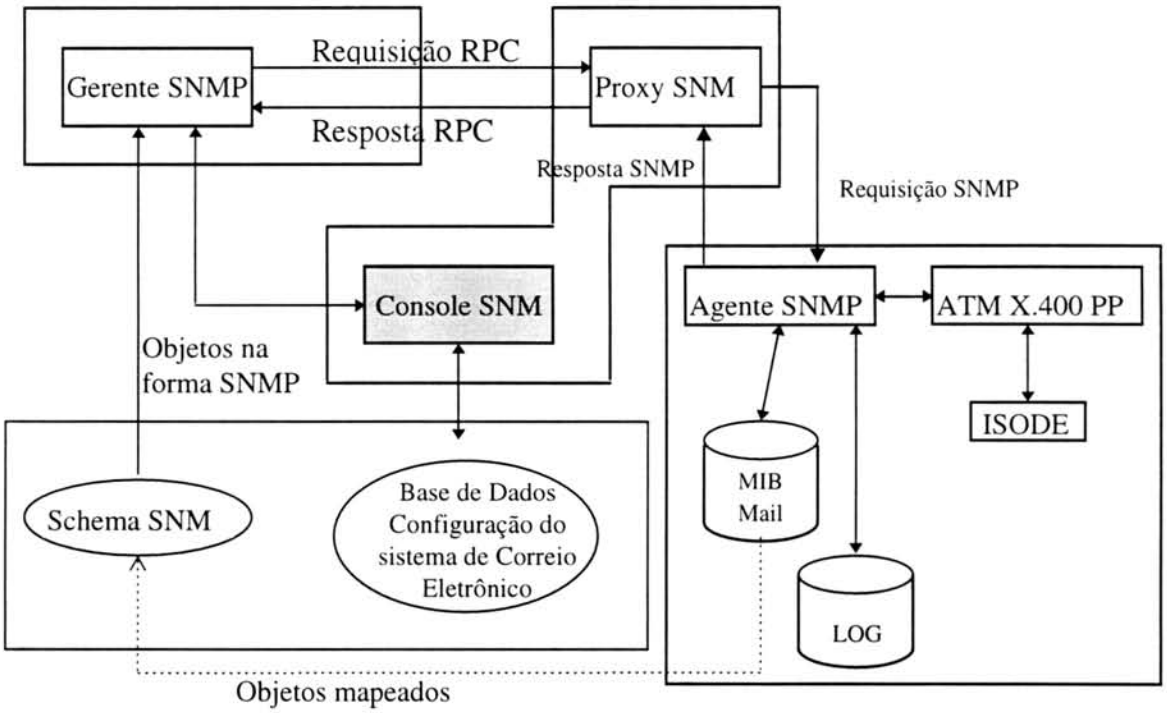

FIGURA 3.4 - Arquitetura do sistema.

Os demais softwares envolvidos, o ISODE e o PP, são respectivamente a implementação da pilha OSI sobre TCP/IP e a aplicação de correio eletrônico. Esses dois softwares são transparentes para o usuário.

A monitoração dos objetos que foram selecionados são obtidos pelo agente SNMP. Estes dados são passados ao gerente que os torna, então, visíveis ao usuário.

Existem mais dois agentes que podem ser utilizados: o agente *ping* e o agente processo. O agente *ping* verifica se a máquina, onde o ATM está instalado, está ativa. Para tanto, este agente encontra-se em outra máquina da rede do ATM. O agente processo monitora a tabela de processos da máquina, verificando possíveis problemas com os agentes que implementam o ATM. As falhas detectadas pelos agentes são gravadas no arquivo de *log*.

3.2.6 Gerência do servidor WWW usando SNMP

É possível também gerenciar servidores WWW através de agentes SNMP, criando portanto MIBs para esse propósito. Com essa facilidade, um grande número de servidores WWW podem ser controlados e monitorados remotamente.

A solução tradicional do gerenciamento de um servidor WWW consiste em examinar os arquivos de *log* criados pelo servidor e tentar identificar problemas ou outras situações que requerem algum tipo de intervenção humana (reinicialização, configuração, entre outras). Diferentemente, com facilidades SNMP integradas em um servidor WWW, é possível gerenciar o servidor de estações remotas. Portanto, utiliza-se plataformas de gerenciamento de rede avançadas que se justificam com o SNMP para monitorar e controlar o estado operacional de vários servidores WWW de um mesmo domínio, por exemplo, vários servidores WWW de uma organização.

Para gerenciar esse tipo de servidor, [PIC 95] propõe um sistema onde um agente SNMP interage com um servidor WWW para coletar e/ou mudar informações concernentes a operação do servidor. Para este fim, uma MIB foi desenvolvida com objetos que armazenam informações relevantes do servidor. O agente SNMP interage com o servidor WWW através de *sockets*. O agente obtém a informação de erro e de acesso no servidor e então incrementa contadores e atualiza as tabelas disponíveis. A partir daí, qualquer gerente de rede pode consultar as variáveis da MIB. Quando erros acontecem, um *trap* específico é enviado para a console de gerenciamento alertando sobre o problema.

As variáveis principais são aquelas que detectam a atividade do servidor. Há também algumas variáveis que fornecem informações de erros, embora os erros sejam tratados principalmente pelo mecanismo de *trap*.

As atividades das variáveis principais são subdivididas em três categorias: orientadas a cliente, orientadas a documento e variáveis gerais. As orientadas a cliente fornecem contadores para monitorar a origem das requisições que foram feitas ao cliente. As orientadas a documento são usadas para monitorar os acessos a documentos e são organizadas em forma de tabela, onde cada coluna informa o nome do documento, o total de requisições e a quantidade de octetos enviados. O último tipo, variáveis gerais, fornecem informações sobre a última requisição e contabilidade de totais, tal como, total de requisições que foram efetuadas.

De acordo com [PIC 95], a utilização desse sistema está sendo de grande valia no gerenciamento de seu servidor WWW, uma vez que agora problemas são reportados diretamente para o NOC (*Network Operation Center*), que podem tomar alguma ação imediata para resolvê-los.

3.3 MIBs Específicas

O objetivo desta seção é apenas exemplificar a utilização de diferentes tipos de MIBs no gerenciamento de certos equipamentos. Através destas MIBs muitos valores úteis podem ser coletados da rede. Dependendo do objetivo do trabalho e dos equipamentos a serem monitorados, uma MIB pode fornecer objetos mais úteis do que outra. Por exemplo, se o objetivo é monitorar apenas roteadores Cisco, a MIB da Cisco deve ser utilizada.

3.3.1 RMON MIB

O advento do gerenciamento SNMP introduziu um novo tipo de dispositivo dentro de redes locais, [ROS 95]. Este dispositivo liga um ou mais segmentos de rede e utiliza recursos internos significantes com o único propósito de monitorar o tráfego no segmento. A informação é coletada de objetos da MIB que estão disponíveis não só em estações de gerenciamento de redes locais, mas também em estações situadas em qualquer lugar na Internet. Este tipo de dispositivo é chamado RMON (*Remote Network MONitoring*).

O padrão RMON define um sistema de duas entidades: a primeira é um monitor ou *probe* (também referenciado como um agente), que reside em cada segmento de rede, e a segunda é a aplicação de gerenciamento. A aplicação de gerenciamento RMON configura agentes remotos para coletar informação, a qual é enviada para uma estação de gerenciamento sempre que ocorrem eventos específicos ou requisições.

Segundo [WAL 95], os objetivos do gerenciamento de rede remoto são:

- operação *offline*;
- monitoração preventiva (pró-ativa);
- detecção e registro de problemas;
- valoração dos dados coletados;
- múltiplos gerentes.

O padrão RMON foi inicialmente definido no RFC 1271, no qual a maioria dos objetos são definidos para gerenciar redes Ethernet, sendo mais tarde substituído pelo RFC 1757, [WAL 95]. Alguns anos depois surgiu o RFC 1513, que é uma extensão da RMON para redes Token Ring.

A RMON é definida como uma MIB, uma estrutura de dados SNMP compreendendo tabelas estatísticas que são acessadas usando comandos SNMP. Dentro da RMON MIB as várias tabelas estatísticas são conhecidas como grupos. O agente RMON usa essas tabelas para coletar dados . Em adição, um agente pode manipular as tabelas independentemente para descarregar o processamento e limitar a quantidade de dados transmitidos sobre a rede.

Abaixo estão listados os grupos que fazem parte da RMON MIB:

- **grupo *Ethernet Statistics***: este grupo contém estatísticas medidas pelo monitor para cada interface no dispositivo que é monitorada. Essas estatísticas são contadores, que iniciam do zero quando uma entrada válida é criada. Atualmente este grupo apenas monitora interfaces Ethernet, ficando as monitorações armazenadas na tabela *etherStatsTable*. Os grupos Token Ring, Mac-Layer Statistics e Token Ring Promiscuos Statistics são definidos no RFC 1513.

- **grupo *History Control***: este grupo guarda amostras de estatísticas feitas periodicamente em vários tipos de rede para uma recuperação posterior. As amostras fazem parte da configuração de cada interface, período de *polling*, estatísticas de uma interface Ethernet em particular, entre outros parâmetros. Uma vez que as amostras são coletadas, seus dados são armazenados na entrada de uma tabela específica. Cada entrada define uma amostra.
- **grupo *Ethernet History***: este grupo controla amostras de estatísticas feitas periodicamente na rede, armazenando-as para uma possível recuperação. Uma vez que amostras são obtidas, seus dados são armazenados em uma entrada de uma tabela específica. Cada entrada define uma amostra, e está associada com a *historyControlEntry* que causou a amostra a ser pega. Este grupo define a *etherHistoryTable* para redes Ethernet.
- **grupo *Alarm***: este grupo gera eventos quando as variáveis monitoradas ultrapassam certos limites. Para que limites não sejam gerados a cada monitoração, um mecanismo, chamado mecanismo de histerese é implementado. Quando este grupo for implementado, o grupo evento também deve ser. O mecanismo de histerese está descrito na seção 4.3.1.2.1.
- **grupo *Host***: este grupo contém estatísticas de cada máquina na rede. O grupo *host* monitora a rede no modo promíscuo pegando todos os pacotes bons que passam por ela. Com isso, ele mantém uma lista com endereços MAC de origem e destino das máquinas. O grupo contém três tabelas: *hostControlTable* (listas de parâmetros que estabelecem o descobrimento de máquinas na interface e na coleção de estatísticas em relação a essa máquina), *hostTable* (contém uma coleção de estatísticas para uma máquina particular descoberta em uma interface desse dispositivo) e *hostTimeTable* (mantém estatísticas de máquinas na rede, incluindo a ordem relativa do tempo na qual cada máquina foi descoberta pelo agente), [ART 96].
- **grupo *hostTopN***: este grupo é usado para preparar relatórios que descrevem as máquinas que estão no topo de uma lista ordenada por uma de suas estatísticas. As estatísticas disponíveis são amostras de uma de suas estatísticas base sobre um intervalo especificado pela estação de gerenciamento, sendo estas estatísticas baseadas em taxas. A estação de gerenciamento também seleciona quantas máquinas aparecerão no relatório. Este grupo requer a implementação do grupo *host*. A *hostTopNControlTable* é usada para iniciar a geração do relatório. A seleção dos parâmetros que aparecerão no relatório é feita pela estação de gerenciamento. Quando um relatório está pronto, entradas são criadas no *hostTopNTable* e associadas com *hostTopNControlEntry*. Essas entradas são estáticas para cada relatório.
- **grupo *Matrix***: este grupo armazena estatísticas para conversão entre conjuntos de dois endereços, isto é, entre duas máquinas. Quando o

dispositivo detecta uma nova conversão ele cria uma nova entrada em suas tabelas. Ele deve somente criar novas entradas baseadas em informações recebidas de pacotes bons.

- **grupo *Filter***: através deste grupo, somente pacotes que passam por um determinado filtro são capturados. Os filtros podem conter expressões lógicas arbitrárias para filtrar certos pacotes. A função booleana OR é utilizada para associar múltiplos filtros. Dentro de um único filtro, as condições podem estar associadas com as expressões booleanas AND ou NOT. Eventos podem ser gerados quando vários pacotes combinam com um filtro. Os eventos podem ser gravados em um *log* ou podem ser enviados ao gerente usando um *trap* SNMP. Através deste grupo pode-se capturar pacotes válidos, inválidos ou algum dos cinco tipos de pacotes de erros (pacotes curtos, pacotes longos, *jabbers*, pacotes fragmentados e pacotes com erros de CRC e alinhamento).
- **grupo *Packet Capture***: este grupo captura pacotes que combinam com um filtro e requerem a implementação do grupo *filter*. Este grupo é formado por duas tabelas: *bufferControlTable* e *captureBufferTable*. A tabela *bufferControlTable* mantém um conjunto de parâmetros que controlam a coleção de um fluxo de pacotes que têm filtros correspondentes. Os pacotes capturados são colocados em entradas na tabela *captureBufferTable*.
- **grupo *Event***: este grupo controla a geração e notificação de eventos, possuindo duas tabelas para essa função. A tabela de eventos descreve os parâmetros de eventos que podem ser gerados. A tabela de *logs* mantém a lista dos eventos que foram armazenados. Esse *log* inclui a hora de cada evento e uma descrição do evento escrito pelo distribuidor do monitor.

A maioria dos grupos descritos anteriormente definem objetos para o gerenciamento de uma rede Ethernet, mas o projeto da MIB permite objetos semelhantes serem especificados para outros tipos de rede. Se um dispositivo de monitoração remota implementa um grupo, ele deve também implementar todos os objetos daquele grupo. Todos os grupos nesta MIB são opcionais. Implementações nesta MIB também implementam os grupos *system* e *interfaces* da MIB II.

A abordagem centralizada da RMON é especialmente importante para as redes de hoje em dia que incluem Ethernet e Token Ring. Embora as primeiras implementações foram para Ethernet, muitos produtos estão agora definidos para Token Ring (RFC 1513). Token Ring foi implementada como uma extensão oficial para a MIB, fornecendo gerenciamento para uma única console centralizada.

A RMON aborda a questão das grandes transmissões de dados usando SNMP. A capacidade de captura de pacotes RMON requer que as estações de gerenciamento sejam capazes de recuperar dos agentes grandes quantidades de dados de uma forma rápida e eficiente.

3.3.2 MIB da Novell

Uma rede Novell utiliza um protocolo próprio para a comunicação entre uma máquina e outra, o protocolo IPX. Para que esse protocolo fosse gerenciado foi criada uma MIB especificamente para ele - a MIB da Novell. Essa MIB foi projetada para suportar múltiplas instâncias do protocolo IPX em um sistema via um identificador de instância de sistema que é o índice primário para cada tabela desta MIB.

Além disso, ela foi projetada para fornecer um *framework* para o gerenciamento de sistemas que implementam o protocolo IPX. MIBs adicionais podem ser criadas, especialmente na área dos protocolos de roteamento do IPX, para conter informações mais específicas. Quando possível, essas MIBs adicionais deveriam seguir o formato da MIB do IPX e deveriam ser "linkadas" a esta MIB via o uso do identificador de instância de sistema mencionado acima.

Os grupos que compõem essa MIB são os seguintes:

- **Grupo *System***: este grupo contém informações gerais sobre todas as instância do IPX rodando em um sistema. A informação desse grupo é armazenada em duas tabelas: *Basic System Table* e *Advanced System Table*. Através de monitorações nos objetos desse grupo pode-se coletar valores como: número de pacotes recebidos e enviados, número de pacotes descartados, número de pacotes com *checksum* incorreto, número de pacotes NETBIOS recebidos, entre outros.

- **Grupo *Circuit***: este grupo contém informações de gerenciamento para cada circuito usado pelo IPX no sistema. É composto por uma tabela, *Circuit Table*, que armazena as informações. Monitorando-se seus objetos, pode-se obter o estado operacional do circuito, o tamanho máximo do pacote em bytes que o circuito suporta, o número de pacotes comprimidos enviados, recebidos e rejeitados, entre outras informações.

- **Grupo *Forwarding***: este grupo possui informações de roteamento que devem ser fornecidas por qualquer protocolo de roteamento do IPX. É formado por duas tabelas: a *Destination Table* e a *Static Routes Table*. A *Destination Table* contém informações sobre todos os destinos conhecidos. A informação de roteamento mostrada nessa tabela representa o caminho correntemente usado para alcançar o destino. A *Static Routes Table* contém informações sobre todas as rotas estáticas definidas, mas somente a rota sendo usada será mostrada nessa tabela.

- **Grupo *Services***: este grupo contém informações sobre todos os serviços conhecidos. Essas informações ficam armazenadas em três tabelas: *Services Table*, *Destination Services Table* e *Static Services Table*. A primeira contém informações indexadas pelo nome e tipo do serviço, na segunda as informações de serviço são indexadas por endereço, nome e tipo e a terceira

contém informações sobre todos os serviços alcançados via uma rota estática.

3.3.3 *Host Resources* MIB

A *Host Resources* MIB define um conjunto de objetos úteis para o gerenciamento do *host*. Conforme [GRI 93], o termo "*host*" é usado para denominar qualquer computador que se comunique com um computador semelhante através da Internet e que é usado diretamente por um ou mais seres humanos. Esses computadores são independentes de sistema operacional, serviços de rede ou qualquer software de aplicação.

A *Host Resources* MIB define objetos que são comuns para muitas arquiteturas de sistemas. Além disso, há objetos na MIB II que também fornecem funcionalidade para gerenciar *hosts*. Portanto, a implementação dos grupos *System* e *Interfaces* é obrigatória nessa MIB.

Abaixo estão descritos os grupos que fazem parte dessa MIB:

- **Grupo *Host Resources System***: este grupo contém informações gerais sobre o *host*, como por exemplo, a quantidade de tempo desde que ele foi inicializado, a data e hora local, o número de usuários logados no *host*, entre outras. A implementação desse grupo é obrigatória.
- **Grupo *Host Resources Storage***: este grupo contém informações sobre o sistema de arquivos do *host*. Ele é composto por um objeto, que informa a quantidade de memória física do *host*, e por uma tabela (*hrStorageTable*). Esta tabela possui informações sobre as áreas de armazenagem lógicas do *host*. Ela é útil para diagnosticar falhas como por exemplo, falta de memória e perda de *buffers*. Além disso, ela pode fazer o papel de uma ferramenta de monitoração de desempenho para fornecer informações sobre a memória, o disco e *buffers* usados. A implementação desse grupo é obrigatória.
- **Grupo *Host Resources Device***: este grupo é útil para identificar e diagnosticar os dispositivos em um sistema. As informações desse grupo são armazenadas na tabela *hrDeviceTable*, a qual contém informações comuns para qualquer tipo de dispositivo. Além disso, alguns dispositivos possuem tabelas específicas com informações mais detalhadas, um exemplo desse tipo de tabela é a *hrPrinterTable* (somente existirá uma entrada nesta tabela se o valor correspondente do objeto *hrDeviceType* for "*hrDevicePrinter*"). A implementação desse grupo é obrigatória.
- **Grupo *Host Resources Running Software***: este grupo contém informações sobre os softwares rodando no *host*, tais como: sistema operacional do *host*, *device drivers* e aplicações. As informações sobre os softwares são

armazenadas na tabela *hrSWRunTable*. A implementação desse grupo é opcional.

- **Grupo *Host Resources Running Software Performance***: este grupo armazena informações sobre o desempenho dos softwares que estão rodando no *host*. Para cada entrada da tabela *hrSWRunTable* (pertencente ao grupo anterior) há uma entrada correspondente na tabela *hrSWRunPerfTable*, tabela utilizada por esse grupo. A implementação desse grupo é opcional.
- **Grupo *Host Resources Installed Software***: este grupo é composto por dois objetos e pela tabela *hrSWInstalledTable* que contém uma entrada para cada software instalado localmente no *host*. Softwares que são carregados remotamente de um servidor da rede não são incluídos nesta tabela. A tabela *hrSWInstalledTable* é útil para identificar e relacionar os softwares em um *host*, bem como diagnosticar problemas de incompatibilidades e versões incorretas entre várias partes de hardware e software. A implementação desse grupo é opcional.

3.3.4 MIB da Cisco

A MIB da Cisco é fornecida com todos os softwares da Cisco e com o software de gerenciamento de roteador *CiscoWorks*. Esta MIB contém variáveis que podem ser configuradas ou lidas e fornecem informações sobre dispositivos de rede e interfaces. A MIB da Cisco é composta por um conjunto de variáveis que são extensões privadas da MIB II. Essas variáveis são acessadas via protocolo SNMP. Além disso, todos os produtos da Cisco suportam os *traps* SNMP especificados na MIB II.

A MIB privada da Cisco é representada pelo identificador de objeto 1.3.6.1.4.1.9 ou iso.org.dod.internet.private.enterprise.cisco. Ela inclui as seguintes sub-árvores: local (2), temporary (3), otherEnterprises (6) e ciscoMgmt (9). A figura 3.5 apresenta sua hierarquia.

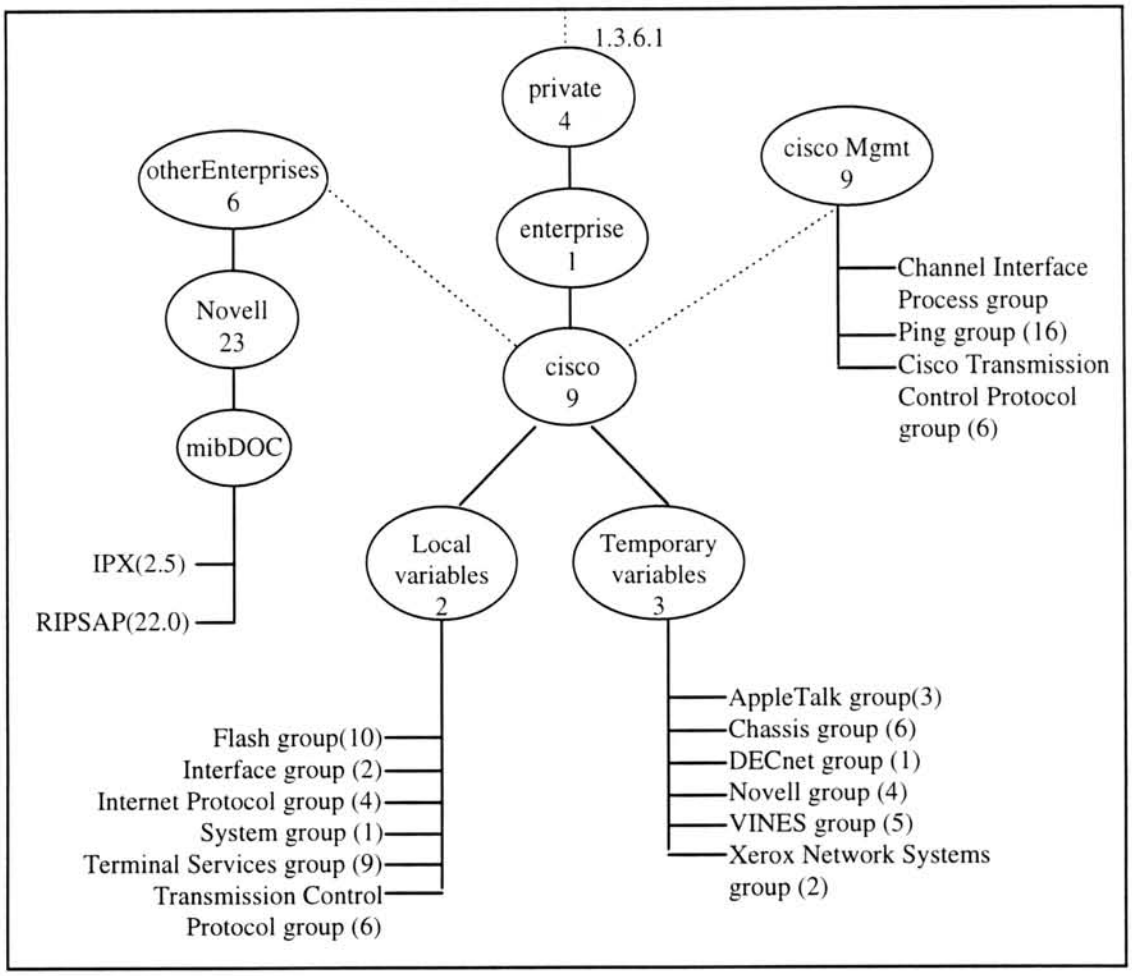

FIGURA 3.5 - Hierarquia da MIB privada da Cisco.

A seguir, cada grupo apresentado na figura acima, está descrito separadamente. Na seção de variáveis locais, os seguintes grupos foram definidos:

- **Grupo *Flash***: este grupo é relativo à memória Flash usada para armazenar e dar boot. As informações disponíveis nesse grupo dizem respeito ao tamanho da memória Flash e ao seu conteúdo. A memória Flash pode ser atualizada através de comandos.
- **Grupo *Interface***: este grupo fornece informações sobre as interfaces de um equipamento Cisco. Essas informações incluem estatísticas de tráfego, estado da linha, velocidade média da entrada e saída dos pacotes e verificação de erros. Os dados obtidos desse grupo podem ser usados no gerenciamento de desempenho. A tabela *lifTable* contém todos os objetos desse grupo.

- **Grupo *Internet Protocol* (IP)**: este grupo fornece informações sobre equipamentos que rodam IP. Através dessas informações pode-se identificar como uma interface obteve seu endereço, quem forneceu esse endereço, o número de mensagens ICMP e número de pacotes perdidos. A maioria das informações desse grupo são armazenadas em tabelas.
- **Grupo *System***: este grupo fornece os parâmetros do sistema para equipamentos Cisco. Os parâmetros do sistema que podem ser coletados são os seguintes: versão de software, nome da máquina, nome do domínio, tamanho do *buffer*, arquivos de configuração e estatísticas do ambiente.
- **Grupo *Terminal Services***: este grupo fornece informações sobre serviços de terminal, tal como número de linhas físicas, estado da linha, velocidade da linha, tipo de controle de fluxo e tipo de modem.
- **Grupo *Transmission Control Protocol* (TCP)**: este grupo fornece estatísticas sobre o número de bytes e pacotes que estão entrando e saindo por conexões TCP. Algumas informações desse grupo são armazenadas na tabela *ltcpConnTable*. O índice dessa tabela inclui o endereço e número de porta da máquina local e o endereço e número de porta da máquina remota para cada conexão TCP ativa naquele equipamento. Portanto, para "n" conexões TCP, haverá "n" entradas nessa tabela.

Existe um espaço experimental definido pela SMI e que contém objetos úteis de serem monitorados, mas que estão além das capacidades da Cisco para controlar e manter. Portanto, o suporte desses objetos pode mudar de uma versão de software dos Sistemas Cisco para outra.

As variáveis temporárias consistem dos seguintes grupos:

- **Grupo *AppleTalk***: este grupo é relativo a dispositivos que estão rodando o protocolo AppleTalk. As informações que podem ser obtidas a partir desse grupo são as seguintes: número total de pacotes de entrada e saída, número de pacotes com erros e número de pacotes com pedidos e respostas do protocolo ARP (*Address Resolution Protocol*).
- **Grupo *Chassis***: este grupo é relativo a informações de hardware dos equipamentos da Cisco, as quais encontram-se armazenadas em duas tabelas a *Chassis Card Table* e a *Chassis Interface Table*. Tais informações incluem: tipos de placas usadas pelo dispositivo, a versão de hardware das placas e o número de slots no chassis.
- **Grupo *DECnet***: este grupo é relativo a dispositivos que rodam o protocolo DECnet. Tais informações incluem: contagem de *hop*, nome da máquina, total de bytes recebidos e enviados e o número de pacotes com erros no cabeçalho.

- **Grupo *Novell***: este grupo é relativo a dispositivos que rodam o protocolo da Novell. Tais informações incluem: o número total de pacotes de entrada e saída, o número de pacotes com erros e o número de pacotes com pedidos e respostas do SAP (*Service Access Point*).
- **Grupo *Virtual Integrated Network System* (VINES)**: este grupo é relativo a dispositivos que rodam o protocolo VINES. O protocolo VINES é derivado do protocolo *Xerox Network Systems* (XNS). Os objetos desse grupo incluem as seguintes informações: número total de pacotes de entrada e saída, número de pacotes com erros e número de pacotes com pedidos e respostas do protocolo ICMP.
- **Grupo *Xerox Network Systems* (XNS)**: este grupo é relativo a dispositivos que rodam o protocolo XNS. Tais informações incluem: número de pacotes "seguidos adiante", número total de pacotes de entrada e número total de pacotes com erros.

A sub-árvore *ciscoMgmt* consiste dos seguintes grupos:

- **Grupo *Channel Interface Processor***: este grupo especifica os objetos usados para gerenciar o *channel interface processor card* da cisco.
- **Grupo *Ping***: este grupo é relativo a requisições de ping (ICMP *echo request*) feitas por um usuário a um dispositivo Cisco específico, e consiste de uma tabela única, a *ciscoPingTable*.
- **Grupo *Cisco Transmission Control Protocol***: este grupo fornece estatísticas sobre o número de bytes e pacotes de conexões TCP que estão entrando e saindo.

A sub-árvore *otherEnterprises* consiste dos seguintes grupos:

- **Grupo *Novell IPX System***: este grupo contém informações gerais sobre todas as instâncias de IPX em um sistema.
- **Grupo *Novell RIPSAP***: este grupo define informações sobre os protocolos RIP e SAP em um ambiente IPX. Essa é uma informação adicional à contida na própria MIB do IPX. Todas as tabelas nesta MIB são "linkadas" a uma instância de IPX via o identificador de instância de objeto definido na MIB do IPX.

4 Automação da Gerência

Um sistema de gerenciamento de rede é um pacote de software projetado para fornecer significativamente eficiência e produtividade na rede. Embora o administrador da rede possa fazer manualmente o mesmo trabalho que faz um sistema de gerenciamento de rede, é preferível que o software execute esta tarefa.

Sistemas de gerenciamento de rede podem ajudar os administradores que trabalham em muitos ambientes diferentes. Além disso, um sistema poderia fazer uma análise mais completa da rede, bem como examinar tendências nos padrões de tráfego. Ele poderia encontrar usuários que violam regras, afetando a segurança da rede. Um sistema desse tipo também poderia encontrar sistemas mal configurados e ajudar a localizar áreas com problemas. Com um sistema de gerenciamento de rede fazendo essas tarefas, o administrador teria mais tempo de fazer a rede corresponder mais a demanda e necessidades de seus usuários, além de poder concluir projetos que foram adiados por ele não ter tempo suficiente para terminá-los.

Um sistema de gerenciamento de rede pode fazer mais do que só tarefas rotineiras. Ele pode ajudar a localizar problemas através de monitorações contínuas sobre a rede e pode produzir *logs* com informações da rede e então usar estes *logs* para estudá-la e analisá-la. Um outro item verificado refere-se a tarefas que não podem ser feitas com tanta rapidez e eficiência quando forem executadas por seres humanos.

Um ambiente mais complexo usualmente requererá que o sistema faça mais tarefas e ajude o administrador da rede em um nível mais sofisticado. Contudo, para redes de dados de qualquer tamanho, um sistema de gerenciamento de rede pode habilitar administradores a trabalhar mais eficientemente juntos, ajustando a rede para servir as necessidades de seus usuários.

Várias universidades têm construído sistemas desse tipo para gerenciar suas redes, suas características variam bastante, embora a grande maioria utiliza o protocolo SNMP, com agentes localizados em vários componentes da rede. Abaixo estão relacionados alguns exemplos de sistemas utilizados no gerenciamento de redes:

- *NetGuardian* (pacote SNMP para Windows que inclui um gerente, um agente e um analisador gráfico para apresentar o desempenho da rede. Desenvolvido na Universidade de Lisboa. Disponível em http://www.di.fc.ul.pt/software/netguard.html);
- *SnmpMan* (aplicação para Windows que permite gerenciar múltiplas máquinas da mesma rede. Desenvolvido na Universidade de Lisboa. Disponível em http://www.di.fc.ul.pt/software/ snmpman.html);
- *Beholder* (software que implementa a RMON MIB e roda nas plataformas OS/2, SunOS, Solaris, Ultrix e Linux. Coleta dados de redes Ethernet e foi

desenvolvido na Universidade Tecnológica de Delft, Holanda. Disponível por FTP anônimo em dnpap.et.tudelft.nl);

- *HNMS* (*Hierarchical Network Management System* - sistema projetado para auxiliar o administrador da rede no gerenciamento de uma rede IP. Utiliza SNMP e X Windows para coleta de dados e apresentação gráfica. Desenvolvido no NASA Ames Research Center. Disponível em http://www.cosmic.uga.edu/pub/hnms.info.shtml).

4.1 Áreas funcionais da gerência de redes

O gerenciamento de redes surgiu da necessidade de se manter uma rede de dados para se maximizar sua eficiência e produtividade. Dependendo das capacidades do sistema que gerencia a rede, o processo usualmente incluirá uma coleta de dados, o processamento destes dados e então sua apresentação para o administrador da rede. Pode também analisar os dados e oferecer soluções e possivelmente corrigir uma situação anormal sem intervenção do administrador. Além disso, ele usualmente incluirá geração de relatórios. Para realizar tudo isso, o fórum de gerenciamento de rede da ISO definiu cinco áreas funcionais para o gerenciamento de uma rede [LEI 93]:

- gerenciamento de falhas;
- gerenciamento de configuração;
- gerenciamento de segurança;
- gerenciamento de desempenho; e,
- gerenciamento de contabilização.

A seguir estão descritas mais detalhadamente cada uma destas áreas.

O **gerenciamento de falhas** é o processo de localizar falhas ou problemas, na rede. Este processo envolve descobrir o problema, isolá-lo e corrigi-lo (se possível). Usando técnicas de gerenciamento de falhas o administrador da rede pode localizar e resolver problemas mais rapidamente do que poderia ser feito sem utilizar essas técnicas. Visando a detecção de falhas de uma rede, deverão existir facilidades para manter e examinar *logs*, aceitar notificações de detecção de erros e atuar em função das mesmas, rastrear problemas, executar seqüências de testes de diagnóstico e corrigir os problemas. O gerenciamento de falhas lida com eventos e *traps* que ocorrem na rede. É importante ter em mente que usar mecanismos para relatar alarmes ou alertas é a melhor maneira de realizar a verificação de desempenho de um objeto gerenciado específico sem ter o dobro da quantidade de *polling* sendo realizada. A maioria dos usuários dos sistemas executam *polling* em objetos gerenciados para pesquisar condições de erro e ilustrar o problema de uma forma gráfica ou uma mensagem textual.

O **gerenciamento de configuração** é o processo de encontrar e configurar dispositivos de rede que controlam o comportamento da rede de dados. Esta é, provavelmente, a parte mais importante do gerenciamento de redes, pois é difícil gerenciar corretamente uma rede a menos que se possa gerenciar sua configuração. Mudanças, adições e retiradas de componentes de uma rede precisam ser coordenadas com um sistema de gerenciamento de rede. Atualização dinâmica da configuração precisa ser realizada periodicamente para assegurar que a configuração é conhecida. Em resumo, o gerenciamento de configuração tem como função a manutenção da estrutura física e lógica da rede.

O **gerenciamento de segurança** é o processo de controlar o acesso a informação em uma rede de dados. Existem informações armazenadas por computadores ligados na rede que são inapropriadas para todos os usuários, como por exemplo, detalhes sobre novos produtos de uma companhia ou sua base de clientes. Alguns sistemas de gerenciamento de rede possuem alarmes sonoros quando a segurança é violada. Este tipo de gerenciamento está associado à geração, distribuição e armazenamento de chaves de criptografia, assim como definições de permissões de controle de acesso a recursos e informações críticas.

O **gerenciamento de desempenho** consiste em medir o desempenho de uma rede e dos meios de comunicação. A atividade que é medida pode ser, por exemplo, a vazão global de octetos, o percentual de utilização, as taxas de erros ou o tempo de resposta. Usando informações de gerenciamento de desempenho, o administrador pode assegurar se a rede tem capacidade para acomodar as necessidades dos usuários. Através deste tipo de gerenciamento, pode-se também examinar tendências da rede. Pode-se usar a tendência dos dados para prever o pico de utilização da rede e conseqüentemente, evitar que o desempenho piore resultando em uma rede saturada.

O **gerenciamento de contabilização** preocupa-se com a manutenção e monitoração de quais e quantos recursos estão sendo utilizados, bem como com a determinação de quem utiliza estes recursos. Isto envolve estabelecer métricas, verificar quotas e determinar custos e contas de usuários.

No decorrer deste trabalho, foi projetada uma ferramenta para apoiar a execução das atividades inerentes à gerência de falhas e de desempenho. Através deste sistema pode-se acompanhar a evolução no tempo de determinados indicadores de desempenho. Então, quando alguma situação crítica for atingida, o sistema sinaliza ao administrador, indicando o causador da situação indesejada. Em conjunto um "Registro de Problemas" é criado para cada problema encontrado.

4.2 Qualidade de Serviço de uma rede

A proliferação da arquitetura cliente/servidor distribuída coloca mais tráfego na rede adicionando sobrecarga. Com novos sistemas sendo adicionados na rede tem-se cada vez mais usuários exigindo tempos de resposta mais rápidos e uma maior qualidade de serviço. Para comportar essas exigências, é preciso assegurar que a rede

opere no seu pico de desempenho, que aplicações importantes estejam rodando sem problemas e que quando um problema começar a se desenvolver, se receberá um aviso para rapidamente isolá-lo e resolvê-lo antes que os usuários percebam.

Em uma rede de computadores, qualidade de serviço (QoS) está ligada a um determinado serviço que a rede presta. Por exemplo, serviço de voz, dados ou imagem. A qualidade de serviço não é um parâmetro geral da rede e sim de um determinado serviço. Em relação aos dados, é preciso haver um serviço confiável para que os dados cheguem corretamente ao destino, independente de por onde eles passem.

Qualidade de serviço também diz respeito a manter a rede funcionando vinte e quatro horas por dia, bem como solucionar os problemas que ocorrem o mais rápido possível, tentando não prejudicar o trabalho dos usuários. É a área de gerenciamento de redes que tenta manter a qualidade de serviço em níveis adequados. Problemas relacionados com falhas, configuração, desempenho, entre outros, são resolvidos a partir do gerenciamento de uma rede.

Uma forma de solucionar os problemas de forma rápida, sem que eles progridam rapidamente, é ter ferramentas que monitoram a rede durante todo o dia e avisem o administrador da mesma assim que problemas forem detectados. Além desse tipo de ferramenta, pode-se ter sistemas capazes de armazenar em uma base de dados todos os problemas que estão ocorrendo ou já ocorreram na rede. Estes sistemas são chamados de *Trouble Tickets* e permitem deixar registrados problemas que estão afetando a rede, e conseqüentemente o usuário.

A qualidade de serviço está diretamente ligada com o *throughput*, retardo, taxas de erros, sigilo, entre outros. Sempre que uma rede não está funcionando corretamente, alguma medida deve ser tomada para que ela volte ao seu funcionamento normal. Vários fatores podem afetar a qualidade de serviço de uma rede. Por exemplo, se uma máquina da rede possui pouca memória, uma aplicação pode demorar algum tempo para ser carregada completamente, mesmo que ela esteja funcionando corretamente. A velocidade da linha é outro fator que influencia bastante na qualidade dos serviços prestados por uma rede. Se a velocidade for muito baixa e a rede possuir muitos usuários, estes ficarão um bom tempo esperando até que seus dados sejam transferidos por completo. É através do gerenciamento da rede que pode-se determinar quais os fatores que estão prejudicando sua qualidade de serviço. Utilizando-se algumas técnicas consegue-se identificar quais componentes da rede estão interferindo no seu funcionamento.

O gerenciamento de desempenho é, às vezes, confundido com o gerenciamento de falhas. O gerenciamento de desempenho é importante não só para garantir a qualidade de serviço prestada aos usuários, como também para assegurar que esta é atingida com os menores custos possíveis. Pode-se, por meio do gerenciamento de desempenho, adequar os meios de comunicação utilizados pelos usuários às suas necessidades, auxiliando o setor responsável pela administração da rede na manutenção dos níveis de desempenho dos serviços oferecidos, como por exemplo, o tempo de resposta. O gerenciamento de desempenho está diretamente relacionado ao planejamento da capacidade do sistema.

A qualidade de serviço muda de rede para rede, é preciso estudar as necessidades dos usuários para tentar fazer com que a rede atenda ao máximo essas necessidades.

4.3 Usando um Sistema de *Trouble Ticket* para apoiar o processo de diagnóstico automatizado

As redes de computadores, hoje em dia, possuem uma diversidade de equipamentos, tecnologias, protocolos e com isso o grau de complexidade da rede se torna muito grande gerando uma diversidade de problemas cada vez maior. O centro de operações tem a responsabilidade de manter os serviços da rede em níveis adequados, e portanto ele deve ter meios eficientes de lidar com toda a diversidade de problemas que pode ocorrer. Um Sistema de *Trouble Ticket* é um meio utilizado por centros de operações, no qual todas as ocorrências que acontecem numa rede ficam registradas.

Um Sistema de *Trouble Ticket* pode ser utilizado para coordenar o trabalho de várias pessoas envolvidas em resolver os problemas da rede. Uma vez que esse sistema esteja sendo utilizado, extensões podem ser adicionadas para ajudar na eficiência das operações da rede. Um exemplo pode ser a geração de relatórios estatísticos utilizando para isso as informações contidas em um *ticket*, isto é, um registro de problemas. Uma outra forma de melhorar a eficiência, e conseqüentemente a consistência das informações, seria com um preenchimento automático dos campos. Além disso, *Trouble Tickets* também podem ser utilizados para a troca de informações entre centros de operações que ajudam a manter uma rede de computadores funcionando.

Segundo [JOH 92], um "bom" Sistema de *Trouble Ticket* deveria servir para muitos propósitos:

- em um centro de operações várias pessoas podem trabalhar em um problema específico. O *Trouble Ticket* contém a história completa do problema, com isso qualquer operador pode começar a trabalhar rapidamente com este problema, sabendo qual a próximo passo a ser tomado, sem ter que consultar outros operadores que estão trabalhando com alguma outra tarefa ou não estão no centro de operações naquele momento;
- seria interessante se o sistema fosse capaz de ordenar todos os registros de problemas abertos por ordem de prioridade, isso permite que os operadores visualizem as tarefas e escolham a próxima de seu turno;
- se o sistema trabalha com correio eletrônico, atribuições de tarefas ou simplesmente pedidos de ajuda podem ser feitos pelo envio de um "*mail*" a determinadas pessoas;
- o sistema deve permitir configurar o *timeout* para cada registro de problemas. Decorrido um tempo específico, alertas podem ser usados para

avisar que determinado problema ainda não foi resolvido ou para relembrar sobre atitudes a serem tomadas com relação a algum registro específico;

- geralmente, quando se tem uma rede com várias sub-redes, certas pessoas do centro de operações ficam responsáveis por uma sub-rede. Para que cada encarregado de uma sub-rede receba informações sobre o andamento de seus registros de problemas, o sistema pode resumir informações relativas a um domínio e enviá-las para seu encarregado no final do dia. Tal relatório pode conter uma lista das ocorrências que ainda se encontram abertas, um resumo das ocorrências abertas nos dias anteriores ou uma lista de ocorrências relacionadas entre si para aqueles locais. Estes relatórios ajudam os responsáveis a terem uma idéia de quais são os problemas e tendências correntes na sua sub-rede;
- os campos de forma fixa dos *Trouble Tickets* permitem classificação dos registros de problemas, isto é, registros podem servir para análise da qualidade de equipamentos e da produtividade do centro de operações. Relatórios sobre tempo médio entre falhas e tempo médio de reparo podem ser gerados para um equipamento específico. Além disso, pode-se ter um controle da qualidade dos equipamentos em uso, permitindo que máquinas sejam trocadas antes que falhem completamente;
- nem sempre tudo ocorre como o esperado em um centro de operações e crises podem surgir, irritando usuários e administradores. Com a base de registros de problemas, todas as atividades do centro de operações podem ser mostradas a seus usuários, indicando o esforço do centro em resolver os problemas.

Uma vez que um problema tenha sido detectado, um *Trouble Ticket* precisa ser criado para ele. Ter um sistema deste tipo é vital para uma instituição monitorar o tipo de rede que está sendo usado e por quem. Ele tem também uma função chave a medida que reúne informações necessárias para calcular o custo da manutenção. Além disso, outra característica importante de um sistema deste tipo é a forma como o diagnóstico pode ser automatizado. Um Sistema de *Trouble Ticket* guarda consigo a história completa de um problema. Logo, quando outro problema de mesmo tipo ou semelhante for detectado na rede, a mesma solução poderá ser aplicada para resolver o problema, não precisando passar por várias etapas desnecessárias até sua solução.

4.3.1 O Projeto CINEMA

O Projeto CINEMA (*Cooperative Integrated Network Management*) é um exemplo no qual um Sistema de *Trouble Ticket* pode ser encontrado. Este projeto foi definido por [MAD 93] para gerenciar redes TCP/IP de uma forma cooperativa e integrada, sendo composto por dois sistemas, o Sistema de *Trouble Ticket* e o Sistema

de Alertas. No Sistema de *Trouble Ticket* operadores podem cooperar entre si para resolver um problema comum. Em adição, o Sistema de Alertas ajuda o pessoal de operações a detectar situações críticas pela monitoração de indicadores específicos ao redor da rede. Além disso, esse sistema pode criar *tickets* automaticamente através da interface de programação de aplicações fornecida pelo Sistema de *Trouble Ticket*.

A justificativa para as características deste sistema podem ser encontradas no trabalho que deu origem ao mesmo [MAD 94] e também no RFC 1297, que justifica as propriedades ideias de um Sistema de *Trouble Ticket*.

4.3.1.1 O Sistema de *Trouble Ticket*

O Sistema de *Trouble Ticket* do ambiente CINEMA trabalha de um modo onde há apenas uma base de *tickets*. Por esse motivo, o pessoal que trabalha com os registros de problemas (*tickets*) executam todas operações que necessitam em um único local, não tendo informações replicadas ou perdidas quando estas forem cadastradas por pessoas diferentes. Sua função é ajudar a manter o tempo médio de reparo e o tempo médio entre falhas em níveis aceitáveis e ao mesmo tempo suportar a gerência de rede de forma cooperativa e integrada. Esses operadores ajudam o centro de operações a identificar o que inspecionar e o que fazer na presença de problemas repetitivos.

Em um registro de problemas três operações podem ser feitas: abri-lo, emitir notas e fechá-lo. Um registro de problemas se mantém aberto até que o problema que o originou seja corrigido. Para cada uma dessas operações, informações distintas estão associadas. Quando a operação de abertura for executada será preciso preencher certos campos que compõem a estrutura do registro, tais como: "identificação", "reclamante", "responsável", "instante de abertura", entre outros, [MAD 94]. Os campos a serem preenchidos são de extrema importância para a pessoa que irá trabalhar com o sistema. Para a emissão de notas sobre um registro de problemas, um campo importante é o "próxima atitude", pois nele está contida qual a próxima atitude que deve ser tomada. Este campo pode ser utilizado para solucionar mais rapidamente o problema, adiantando seu fechamento.

Quando um registro de problemas for fechado, as seguintes informações são registradas: "identificação do *ticket*", "instante de fechamento", "autor", "código de fechamento", "componente afetado" e "observações". Este último campo é um espaço livre para comentários a respeito do processo de solução do problema.

Um registro de problemas só será aberto quando o problema não puder ser resolvido na hora em que ele for avisado. Sempre que um problema ocorre, o centro de operações é avisado, e a pessoa que dá suporte aos usuários tenta resolver o problema naquele momento. Caso este problema não seja solucionado de imediato, será aberto um registro de problemas para ele contendo várias informações sobre o problema em questão. Geralmente muitos problemas não podem ser resolvidos rapidamente e talvez o processo de reparo dependa de técnicos ou especialistas não disponíveis no momento, então são feitas notas no registro sempre que algum evento importante ocorre.

Uma descrição sucinta do problema é mantida no registro de problemas quando este é aberto. O preenchimento deste campo é semi-automático, pois é orientado a menus, facilitando consultas posteriores e facilitando também sua identificação. Assim, se mais de uma pessoa registrar um mesmo problema que tenha ocorrido em dias diferentes e máquinas diferentes, este ficará padronizado, possuindo a mesma descrição do anterior. Além disso, como várias pessoas podem alterar um mesmo registro, ter uma descrição padronizada facilitará quando este precisar ser encontrado.

Enquanto um registro está aberto, pode-se deixar registada todas as atividades que precisam ser feitas para tentar resolver o problema. Com isso, se outra pessoa pegar aquele registro, ela saberá qual a próxima atitude a ser tomada. Isso é muito importante quando várias pessoas trabalham no mesmo registro, além de lembrar de alguma forma as atividades a serem feitas. À medida em que novas informações e tentativas de solução forem surgindo a respeito do problema, elas vão sendo armazenadas, tendo-se, portanto, a história do problema. Um registro de problemas deve ter um tempo para permanecer aberto. Caso este tempo seja ultrapassado, uma notificação é enviada ao usuário.

Daqui para frente no texto, sempre que for mencionada a palavra usuário significa o administrador da rede, pessoa responsável por gerenciar a rede e mantê-la funcionando. É o administrador da rede que vai trabalhar como o sistema em questão.

Um modo de controlar a qualidade dos serviços efetuadas numa rede é calcular a diferença entre a data e hora de abertura e de fechamento do registro de problemas. Através deste cálculo pode-se determinar o tempo médio de reparo. Além disso, tendo a diferença entre a data e a hora do fechamento de um registro anterior com a data e a hora de abertura do registro seguinte para um mesmo equipamento, pode-se calcular o tempo médio entre falhas. Tendo esses dois tempos disponíveis, é possível gerar relatórios relacionando esses dados com dados de fabricantes/vendedores buscados do banco de dados de configuração do centro de operações.

O sistema de *Trouble Ticket* desenvolvido por [MAD 94] possui cinco módulos, conforme figura 4.1:

- **console**: este módulo serve para deixar o sistema seguro, isto é, para que pessoas não autorizadas não consigam abrir e encerrar um registro de problemas. É através deste módulo que o administrador do sistema cadastra todas as pessoas aptas a abrir, gerar notas a respeito e encerrar ocorrências, garantindo assim que outras pessoas não consigam fazer qualquer operação pois não foram cadastradas.
- **cão-de-guarda**: através deste módulo um aviso de um registro de problemas aberto é enviado ao encarregado daquele registro, não deixando que ele caia no esquecimento. Este é um módulo muito importante no sistema, uma vez que a recuperação das falhas deve ser efetuada o mais rápido possível para não prejudicar a imagem do centro de operações.

- ***kernel***: este é o módulo principal do Sistema de *Trouble Ticket*, pois é ele que permite que pessoas de centros de operações diferentes acessem a interface com a base de registros de problemas.
- **interfaces de banco de dados e de notificação**: para que o *kernel* não saiba dos detalhes da tecnologia sendo utilizada para armazenamento e entrega da informação, são utilizados módulos de interfaces de banco de dados e de notificação. Estas interfaces deixam transparente para o *kernel* o tipo de banco de dados utilizado.

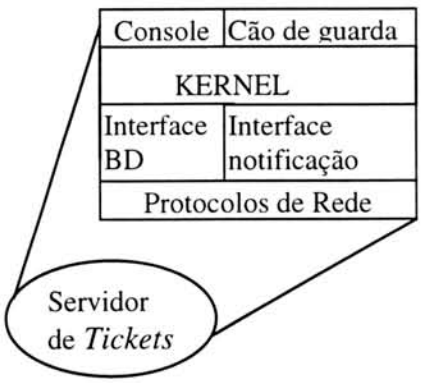

FIGURA 4.1 - Organização dos módulos do sistema.

O Sistema de *Trouble Ticket* tem como finalidade a solução de problemas de uma forma rápida. Visto que problemas ocorrem várias vezes, será mais fácil corrigi-lo quando o problema já ocorreu uma primeira vez, pois cada registro carrega consigo a história completa do problema. Tendo uma base de registros de problemas, pode ser feita uma pesquisa para verificar qual solução foi tomada dada a ocorrência de algum problema na rede.

Além disso, as informações contidas em um registro de problemas podem ser utilizadas em relatórios estatísticos. Registros de problemas podem ser utilizados para trocar informações entre os centros de operações de instituições que ajudam a manter uma rede de computadores funcionando.

O ambiente CINEMA pode ser integrado com aplicações, que detectam situações críticas na rede, através de uma interface de programação de aplicações (API) que foi definida para ele. Assim, quando outras aplicações detectam algum problema na rede elas podem utilizar automaticamente esta interface para gerar um registro para aquele problema.

Quando o ambiente CINEMA foi desenvolvido utilizou-se a API do pacote SunNet Manager. No momento atual está sendo feita uma adaptação de tal sistema para uso no Netscape (como interface com o operador). O sistema de banco de dados está sendo mantido o mesmo (Postgres), mudando apenas a interface que torna-o mais fácil

de ser exportado para outros sistemas, uma vez que a maioria das pessoas possuem o Netscape instalado em suas máquinas.

4.3.1.2 O Sistema de Alertas

O Projeto CINEMA, além de especificar um Sistema de *Trouble Ticket*, também definiu um sistema para alertar o usuário quando determinadas condições forem atingidas. Este, ao contrário do Sistema de *Trouble Ticket*, não chegou a ser implementado completamente e foi tomado como base para a definição do paradigma proposto neste trabalho.

É muito importante para as pessoas encarregadas de administrar uma rede de computadores a utilização de ferramentas que monitoram a mesma. Essas ferramentas permitem avaliar o desempenho da rede de forma ágil, assim como tomar as atitudes cabíveis tão logo perspectivas de falhas surjam. Através das ferramentas de monitoração, o administrador não precisa se preocupar em coletar informações manualmente, pois é muito fácil de ocorrerem erros, além de levar um tempo consideravelmente grande. Dependendo do tipo de ferramenta utilizada, após a monitoração ser efetuada, o administrador é avisado dos problemas que estão ocorrendo na rede e então cabe a ele tentar resolvê-los. Essas ferramentas geralmente emitem relatórios sobre o estado atual da rede, fornecendo estatísticas proveitosas para o administrador.

O Sistema de Alertas funciona da seguinte forma: objetos e máquinas definidas pelo usuário, através de um arquivo de configuração, são monitorados. Detalhes da forma de monitoração, que usam o SNMP, podem ser encontrados na seção 5.2. Cada valor amostrado é decrementado de seu antecessor, dando a certeza ao usuário de que os valores obtidos refletem a evolução do indicador desde o início da amostragem. Após a coleta destas informações, o sistema faz uma análise estatística e gera eventos se algum dos objetos ultrapassar um determinado parâmetro, também chamado "limite". Cada evento gerado só é indicado ao usuário se o mesmo se transforma em um alerta. Para que isso ocorra, é preciso descobrir se os eventos são críticos e irão prejudicá-la de alguma forma. Quando alertas são gerados, o administrador fica sabendo que um problema ocorreu com a rede e que deve ser corrigido o mais rápido possível. Desta forma, o administrador fica livre para realizar outras tarefas enquanto o sistema efetua as monitorações definidas por ele.

As informações de desempenho devem ser analisadas sobre um determinado período de tempo, também definido pelo usuário no arquivo de configuração. Esse período de tempo pode variar de poucos segundos para um mês ou mais. Portanto, o sistema coleta os dados em equipamentos definidos pelo usuário, filtra-os após uma análise estatística e gera eventos. Nesta análise é feita uma comparação do dado coletado com alguns parâmetros, denominados limites, que podem ser calculados de forma automática pelo sistema ou fornecidos estaticamente pelo usuário. Eventos subseqüentes e de mesma natureza não são gerados repetidamente, pois o sistema utiliza o Mecanismo de Histerese, como proposto por [WAL 95].

Eventos são indicadores de situações anormais detectadas a partir da filtragem dos dados coletados junto aos equipamentos da rede. Enquanto alertas são indicadores de que estas situações anormais são críticas, e representam algum tipo de ameaça à manutenção do nível de qualidade dos serviços da rede.

O Sistema de Alertas foi especificado para a rede da UFRGS, e por esta ser uma rede TCP/IP, o sistema suporta a base de informações de gerência MIB II.

4.3.1.2.1 O Mecanismo de Histerese

Este é um mecanismo proposto na RFC 1271 para o grupo de Alarmes da RMON MIB [WAL 95] e utilizado no Sistema de Alertas para a geração de eventos. Seu funcionamento é definido da seguinte forma: dois limites são definidos para cada objeto monitorado, o limite de subida (ou superior) e o de descida (ou inferior). Os limites indicam o intervalo dentro do qual o valor de um objeto amostrado deve encontrar-se quando a rede está no seu estado "normal", ou seja, tudo está funcionando normalmente sem muito atraso e sem a rede estar muito lenta. Caso o valor recebido pelo sistema encontrar-se fora desse intervalo, um evento é gerado A figura 4.2 ilustra a aplicação desse mecanismo.

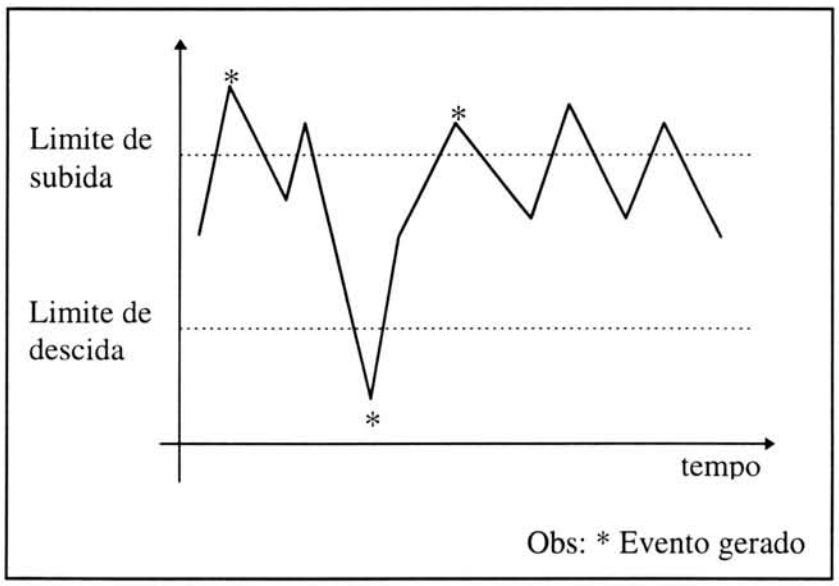

FIGURA 4.2 - O mecanismo de Histerese.

Segundo [MAD 94], um evento de subida (ou superior) é gerado se:

- o valor recebido é o primeiro desde o início da amostragem e o *flag* para ação inicial definido pelo usuário é EVENTO_DE_SUBIDA ou EVENTO_DE_SUBIDA_OU_DESCIDA;
- o valor recebido não é o primeiro a ser amostrado, e este valor é maior ou igual ao limite de subida, sendo que o último valor recebido até então era menor que o limite de subida e o último evento gerado foi um evento de descida.

De um modo complementar, um evento de descida (ou inferior) é gerado se:

- o valor recebido é o primeiro desde o início da amostragem e o *flag* para ação inicial definido pelo usuário é EVENTO_DE_DESCIDA ou EVENTO_DE_SUBIDA_OU_DESCIDA;
- o valor recebido não é o primeiro a ser amostrado, e este é menor ou igual ao limite de descida, sendo que o último valor recebido até então era maior que o limite de descida e o último evento gerado foi um evento de subida.

Este mecanismo garante que eventos idênticos nunca serão gerados em seqüência. Conforme a figura 4.3, uma determinada instância de um objeto pode ultrapassar o mesmo limite várias vezes mas de forma esparsa no tempo sem que o limite oposto seja ultrapassado neste tempo. Entretanto, existem casos onde é preciso que eventos idênticos sejam gerados repetidamente. Isto pode ocorrer quando o mesmo evento é gerado em intervalos esparsos no tempo, sendo necessário gerar um novo aviso para o administrador. Foi pensando nisso que [MAD 94] propôs algumas alterações no mecanismo original. Então para que mais de um evento possa ser gerado, o sistema considera o tempo decorrido desde o último evento gerado. Se o tempo decorrido for maior que o tempo de espera definido pelo usuário, um evento idêntico será gerado, caso as demais condições sejam satisfeitas.

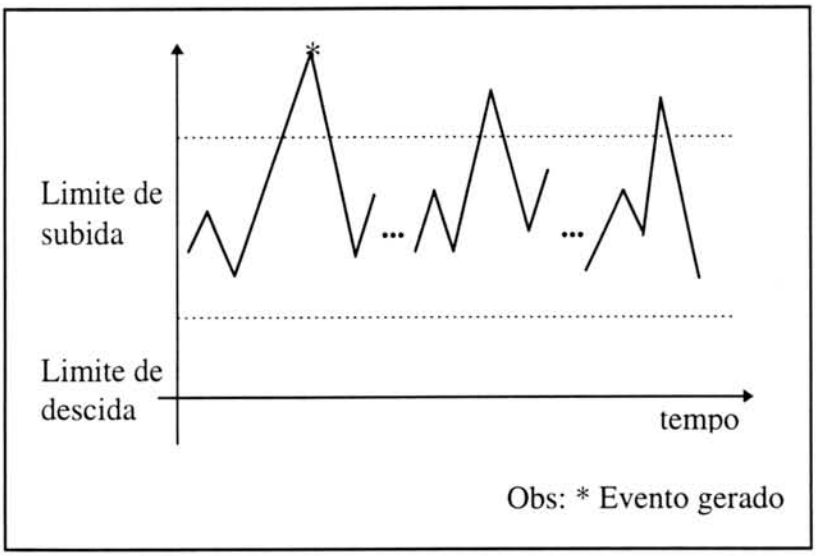

FIGURA 4.3 - Mecanismo de Histerese original desconsidera tempo entre eventos idênticos.

Outra modificação em relação ao mecanismo original é a não implementação do *flag* de ação inicial. Neste caso, se o primeiro valor amostrado encontrar-se fora dos limites definidos pelo sistema, um *flag* de ação inicial constante e igual a EVENTO_DE_SUBIDA_OU_DESCIDA será gerado.

É importante também levar em consideração quando limites tendem ao infinito, sendo eles de subida ($+\infty$) ou de descida ($-\infty$), e ainda um igual ao outro. Em cada caso deve-se tomar bastante cuidado para que eventos sejam gerados no tempo correto e segundo as condições definidas inicialmente.

Outro aspecto importante observado pelo sistema é em relação à utilização da rede que se diferencia durante todo o dia. Geralmente no início da manhã o número de pessoas que estão trabalhando em suas estações é bem menor que no final da manhã, bem como no início e final da tarde onde o tráfego também varia bastante. Foi pensando nisso que se criou uma outra extensão para o Mecanismo de Histerese, o cálculo automático dos limites. Este cálculo é feito de acordo com o tamanho da janela de amostragem definido pelo usuário no arquivo de configuração, que será explicado na seção 4.3.1.2.2. Conforme [MAD 94], é muito importante compreender a real implicação de calcular os limites dinamicamente pelo sistema: os limites superior e inferior tornam-se flutuantes ao longo do tempo (principalmente para indicadores de tráfego e outros que sofrem variações no tempo). Além disso, também é importante notar que a distância entre os limites superior e inferior de um mesmo objeto monitorado sempre variam no tempo, exceto para objetos monitorados cujos valores para as instâncias são constantes ao longo do tempo.

Os limites de subida e descida são calculados através da média e desvio padrão dos valores que foram amostrados na monitoração anterior. Desta forma, as expressões utilizadas são as seguintes:

$$LS = \mu + (S \times \sigma)$$

$$LI = \mu + (I \times \sigma)$$

onde:

LS $\Rightarrow$ limite superior (ou de subida);

LI $\Rightarrow$ limite inferior (ou de descida);

μ $\Rightarrow$ média aritmética para os valores amostrados contidos na janela de amostragem;

S $\Rightarrow$ fator para cálculo do limite superior;

I $\Rightarrow$ fator para cálculo do limite inferior;

σ $\Rightarrow$ desvio padrão para valores amostrados contidos na janela de amostragem.

Além de uma média aritmética comum, pode-se calculá-la conforme o tipo de monitoração definido: "delta" (são consideradas as variações entre os valores a cada novo intervalo de amostragem), "delta por segundo" (são consideradas as variações entre os valores a cada segundo) ou "absoluto" (as instâncias dos objetos são analisadas na forma em que foram amostradas). Dependendo do tipo de amostragem, a média será calculada de forma diferente e irá influenciar no valor do desvio padrão e posteriormente no limite inferior e superior.

Os fatores de cálculo das expressões acima são providos de tal forma que transformam-se em pontos de customização para a administração da rede na obtenção dos limites. Estes fatores podem ser usados para alargar ou estreitar a faixa na qual os valores amostrados devem situar-se.

4.3.1.2.2 Janela de Amostragem

Quando um limite for calculado dinamicamente pelo sistema ele precisa saber quais são as amostragens que devem ser levadas em consideração para este cálculo. É o valor da janela de amostragem que define este parâmetro.

O cálculo das médias e desvios padrão descrito anteriormente deve ser feito de acordo com um tempo estipulado previamente no arquivo de configuração. É através do mecanismo de janela de amostragem que o espaço de tempo entre as monitorações é determinado.

Existem dois tipos de janelas de amostragem: as de tamanho cumulativo, incluindo todas amostras efetuadas desde o início da coleta de dados até o momento

atual, bem como as de tamanho fixo. As janelas de tamanho fixo podem ainda ser estáticas, quando limites são calculados uma única vez e sem muita freqüência, ou deslizantes, quando limites são calculados cada vez que chegam novos dados à estação de gerência.

Os três tipos de janelas têm suas vantagens e desvantagens, sendo as janelas de amostragem deslizantes aquelas utilizadas no sistema em questão. A principal vantagem desse tipo de janela é o fato dos parâmetros de filtragem sempre serem atuais e auto-adaptáveis a situações de mudança. Assim, o cálculo da média e desvio padrão sempre são recalculados a cada nova amostra recebida. Quando uma nova amostra chega e a janela encontra-se cheia, a amostra mais antiga é descartada e a nova é colocada em seu lugar.

4.3.1.2.3 A Reinicialização de Contadores

A maioria dos objetos da MIB II são contadores. Esses objetos podem apenas aumentar seu valor, sendo portanto cumulativos. Mas existe um determinado momento em que eles são reinicializados, voltam para zero e recomeçam a partir dali, este processo é conhecido como "*wrap around*". Isto acontece quando o valor 2^{32} - 1 ou 4.294.967.295 é atingido, conforme [McC 91]. Um exemplo desse tipo de objeto é o número de octetos que saem de uma determinada interface (*ifOutOctets*). Contudo, existem outros motivos que podem levar um contador a sua reinicialização. Um outro caso em que pode correr interrupção durante a contagem de uma monitoração é quando, por exemplo, um roteador é reinicializado. Neste caso, como no anterior, também há uma reinicialização dos contadores, voltando todos a contagem a partir do valor zero. Ferramentas que se utilizam de monitorações contínuas em objetos contadores devem se preocupar em ter alguma forma de determinar quando interrupções destes tipos ocorrem.

Ainda outra situação onde pode ocorrer interrupção dos contadores é quando operações de escritas são efetuadas por diferentes gerentes em um mesmo contador não "*read-only*", [MAD 94]. Mas como é impossível determinar a ocorrência de operações de escrita em qualquer momento passado, esta situação não foi considerada. Logo, é importante diferenciar quando ocorrem *resets* e quando ocorrem *wrap arounds*. Em ambos os casos, quando se estiver trabalhando com diferenciais de contadores em tempos diferentes (t_{i-1} e t_i) esses diferenciais não devem ser baseados em apenas uma subtração simples como $A(t_i)$ - $A(t_{i-1})$, mas deve-se levar em consideração o valor obtido antes da reinicialização com a seguinte expressão:

$$\Delta A_i = (A(Z) - A(t_{i-1})) + A(t_i)$$

onde:

$A(Z)$ $\Rightarrow$ valor obtido imediatamente antes da reinicialização;

$A(t_{i-1}) \Rightarrow$ valor obtido no instante da última amostragem realizada;

$A(t_i)$ $\Rightarrow$ valor obtido no instante da amostragem sendo realizada (após a reinicialização).

A figura 4.4 ilustra o comportamento de um contador qualquer quando ocorre uma reinicialização.

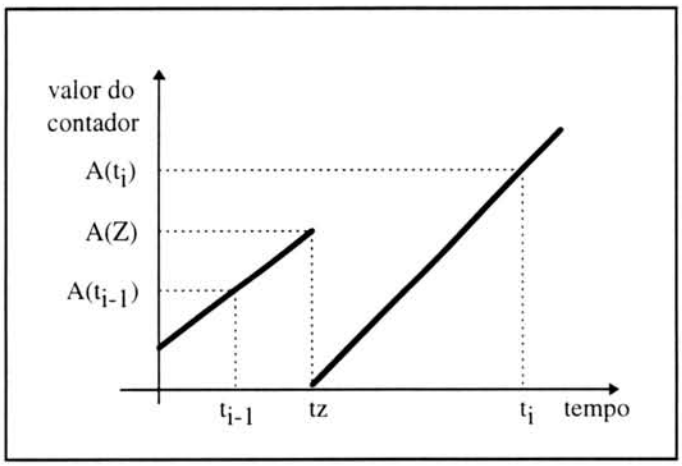

FIGURA 4.4 - Reinicialização de um contador.

Na figura:

t_z ⇒ instante da reinicialização;

t_{i-1} ⇒ instante da última amostragem realizada;

t_i ⇒ instante da amostragem sendo realizada (após a reinicialização).

A forma de detectar quando ocorreu um *reset* e um *wrap around* é bastante simples. Monitorando-se paralelamente o objeto *SysUpTime*, que indica há quanto tempo o agente está rodando, é possível determinar se uma máquina foi reinicializada ou não. Caso a instância desse objeto seja menor do que a última amostragem, então assume-se que houve a reinicialização do equipamento gerenciado, ou seja, ocorreu um *reset*. Caso contrário, assume-se que ocorreu um *wrap around* no contador.

Sabendo-se que ocorreu um *wrap around* e que isto ocorre quando um contador chega ao seu limite máximo 2^{32} - 1 (4.294.967.295) é possível utilizar-se da seguinte expressão para a determinação dos diferenciais.

$$\Delta A_i = ((\text{VALOR MÁXIMO}) - A(t_{i-1})) + A(t_i)$$

Na verdade, é impossível determinar de forma precisa se mais de um *wrap around* ocorreu entre um instante e outro de monitoração, se estes forem muito distantes em tempo. Para tanto, é muito importante dimensionar o mais adequadamente possível o

intervalo de amostragem de forma que ocorra no máximo uma interrupção a cada instante de amostragem.

Da mesma forma é muito difícil determinar o instante exato antes de um *reset* ocorrer. Portanto, sempre que *resets* são detectados pelo sistema, o valor calculado como indicado acima é colocado em seu lugar.

4.3.1.3 Forma de integração com o CINEMA

Inicialmente o Sistema de *Trouble Ticket*, especificado no Projeto CINEMA, foi implementado sobre o SunNet Manager, utilizando para tanto uma interface de programação de aplicações (API) desse pacote.

O Pacote SunNet Manager, por ser proprietário da Sun Microsystems, não torna o sistema portável para as demais plataformas existentes. Por esse motivo, ele foi reestruturado de forma que utilize o software Netscape para interface com o usuário e o pacote CMU-SNMP (pacote de software da Carnegie Mellon University que implementa o protocolo SNMP) como ferramenta de suporte ao uso do SNMP, [TAR 96].

Quando este sistema foi proposto, foi projetada uma interface de programação de aplicações para ele com o objetivo de permitir a integração com outras ferramentas que compõem o ambiente de gerência. Portanto, o paradigma proposto utilizará essa API para comunicar-se com o Sistema de *Trouble Ticket*. Abaixo, na tabela 4.1 estão apresentadas as funções que compõem a API:

TABELA 4.1 - API do Sistema de *Trouble Ticket*.

Classe	**Primitiva**	**Descrição**
Gerência de Sessão	CineTT_Establish()	estabelece uma sessão com o servidor de *tickets*
	CineTT_Terminate()	termina uma sessão com o servidor de *tickets*
Escrita	CineTT_OpenTkt()	abre um *ticket*
	CineTT_NoteTkt()	emite nota sobre determinado *ticket*
	CineTT_CloseTkt()	fecha determinado *ticket*
	CineTT_UpdateTkt()	atualiza informações
	CineTT_DeleteTkt()	remove um dado *ticket* da base
Leitura	CineTT_GetFirstTkt()	busca o primeiro *ticket* de uma dada consulta
	CineTT_GetNextTkt()	busca o próximo *ticket* da consulta atual
Exceção	CineTT_Error()	retorna um *string* dado um código de erro

A interface de programação é orientada à conexão, sendo as primitivas para chamadas do sistema do tipo **socket()**. Portanto, sempre que o cliente desejar fazer qualquer operação nos *tickets*, é necessário primeiramente estabelecer uma conexão com o servidor de *tickets*, através da primitiva **CineTT_Establish()**. Quando a operação for encerrada, a aplicação deverá utilizar a primitiva **CineTT_Terminate()** para liberar a conexão.

5 Características do MAD

Hoje em dia é comum para muitos administradores de rede coletar e arquivar medidas relacionadas ao gerenciamento da rede que indicam como está sua utilização, seu crescimento e as falhas que estão ocorrendo. O principal objetivo é facilitar o isolamento de problemas e o planejamento da rede dentro de uma organização. Existe uma variedade de ferramentas de gerenciamento de rede para coletar e apresentar as medidas disponíveis. Contudo, tendo-se tipos diferentes de medidas e técnicas de apresentação distintas torna-se difícil comparar dados entre redes.

O paradigma descrito neste capítulo define uma forma para automatizar o diagnóstico de problemas assim que os mesmos forem detectados pelo sistema. Diferentes objetos disponíveis na MIB II podem ser monitorados e à medida que problemas são descobertos um *Trouble Ticket* é criado, contendo uma sugestão de como solucioná-lo ou alguns passos para se chegar a solução. A forma de diagnosticar porque determinado problema está ocorrendo é feita através de um conjunto de regras. Para cada problema uma regra específica foi definida e portanto o sistema atua como um sistema especialista.

Um dos maiores avanços na área de gerenciamento de redes adveio do uso de sistemas especialistas. Por exemplo, na área de gerenciamento de falhas, sempre que um problema acontecer, um sistema tenta tomar diversas decisões, analisando o histórico das ocorrências do sistema, o que reduz em muito os trabalhos rotineiros executados por operadores de rede, [BRI 93]. Outro exemplo pode ser encontrado na área de gerenciamento de desempenho onde uma vez detectada uma redução do nível de serviço a um usuário, automaticamente os sistemas tomam determinadas decisões a fim de retornar o nível de serviço adequado. Através da análise de dados de tráfego das redes, pode-se sugerir alterações que visem otimizar os custos da rede, mantendo os mesmos níveis de serviço.

5.1 Sistema Especialista

Um sistema especialista é um software de solução de problemas que incorpora conhecimento especializado em um domínio limitado para fazer um trabalho que geralmente é feito por uma pessoa muito bem treinada [CRO 88]. Sistemas especialistas capturam as estratégias para solucionar os problemas dos especialistas e aplicam-nas seletivamente sobre circunstâncias específicas. Um sistema especialista pode trabalhar com dados incompletos e inexatos, lidar com complexidade, fornecer explicações de suas conclusões e talvez aprender por experiência. Nem todos os sistemas especialistas possuem todas essas características, mas pode-se dizer que esses sistemas têm a

capacidade, dado algum estado de um processo, de dizer o que fazer, baseado em um conhecimento que pode ser aplicado para aquela situação.

Em geral, sistemas especialistas aplicam técnicas de Inteligência Artificial, tendo como núcleo de tais sistemas uma base de conhecimento, que consiste numa coleção de fatos, definições e regras heurísticas, adquiridas diretamente do especialista humano [TAR 90]. Essa base de conhecimento é geralmente separada da parte do programa que faz o raciocínio, a máquina de inferência (responsável pela análise do conhecimento e a dedução das conclusões). Com isso, é possível modificar uma parte sem ter que alterar a outra, como por exemplo, adicionar mais dados na base de conhecimento, ou regular a máquina de inferência para melhorar o desempenho, sem ter que modificar o código. Esses sistemas baseados em conhecimento são sistemas que resolvem problemas pela aplicação de uma representação simbólica, ao invés de empregarem algoritmos e métodos estatísticos.

Especialistas tentam representar o conhecimento da maneira mais uniforme possível. Isto torna o conhecimento mais fácil de codificar e entender, e ajuda a manter a máquina de inferência simples, mas também pode causar problemas se diferentes tipos de conhecimento são forçados dentro de um mesmo formalismo [JAC 86].

Ao contrário dos programas de solução de problemas mais convencionais, que empregam métodos algoritmos e numéricos, programas baseados em conhecimento tentam oferecer ao usuário algum tipo de explicação de como as conclusões são alcançadas, mostrando-lhe quais módulos de conhecimento se tornam ativos e em qual ordem.

O uso de heurísticas, que servem para pesquisar as soluções de problemas, é uma característica chave de um sistema especialista. Essas heurísticas são basicamente "regras" que especialistas no problema utilizam para resolvê-los [PAS 86]. A heurística não diz especificamente o que está certo ou errado, mas ela ajuda a decidir qual dos dois ou mais caminhos alternativos podem levar a solução da nova situação, sendo por isso usada pela máquina de inferência.

Toda a informação conhecida por um especialista deve ser armazenada e representada de alguma forma dentro do computador. Não é possível introduzir todo o conhecimento do especialista humano através de uma simples tradução da linguagem natural para a linguagem formal e portanto, existem diferentes formas de representar o conhecimento, tais como conjuntos de regras, grafos generalizados e lógica de predicados. A escolha da representação mais adequada depende do tipo de problema a ser resolvido e da forma na qual o conhecimento pode ser mais facilmente especificado e utilizado. É muito importante que a linguagem de representação tenha uma lógica adequada para expressar o conhecimento que se deseja representar. A representação do conhecimento é uma forma sistemática de codificar o que um especialista sabe sobre certo domínio. A maioria dos sistemas especialistas aplicados na área de gerenciamento de redes utilizam tecnologias baseadas em regras ou outras tecnologias simples para representar o conhecimento.

No desenvolvimento de um sistema especialista é necessário, em primeiro lugar, adquirir o conhecimento de um especialista humano (pessoa que lida diariamente

com o problema) criando-se assim a base de conhecimento, para posterior manutenção e aperfeiçoamento da mesma. O MAD utiliza regras e redes semânticas para representar o conhecimento e portanto, somente estas duas formas estão descritas a seguir.

5.1.1 Sistemas de Produção

Uma das formas de representar o conhecimento é através de regras de produção, as quais possuem a forma de regras *IF-THEN*. Essa é uma linguagem declarativa, isto é, ela especifica o que fazer, ao contrário de linguagens procedurais que especificam como fazer alguma coisa.

Um sistema de produção consiste de um **conjunto de regras**, um **interpretador**, que decide que tipo de regras aplicar, e uma **memória de trabalho**, que pode conter dados, metas ou resultados intermediários. Uma regra é, simplesmente, um par "condição-ação": dada a existência da condição expressa, faça a ação. As regras são expressas da seguinte forma:

IF <condição> *THEN* <ação>

A principal função da memória de trabalho é armazenar dados freqüentemente na forma, "objeto-atributo-valor". Esses dados são usados pelo interpretador para que ele possa decidir quais regras serão ativadas em determinado momento.

É através de inferências, análises baseadas em fatos e premissas, que o sistema consegue tirar suas conclusões. Tendo um ambiente baseado em regras, a inferência determina quais regras são aplicáveis e quais destas regras deveriam ser usadas em uma determinada situação. Se uma condição não contém variáveis, então ela é satisfeita somente se uma expressão idêntica estiver presente na memória de trabalho. Se uma condição contém uma ou mais variáveis, então ela só será satisfeita após encontrar uma expressão na memória de trabalho que combine com a condição, fazendo-se as devidas instanciações. O sistema opera aceitando entradas e aplica regras apropriadas para tirar conclusões sobre essas entradas. As regras são aplicadas nas conclusões para tirar conclusões futuras; este processo continua até que o problema seja resolvido.

Portanto, a inferência de um sistema especialista baseado em regras de produção implementa o seguinte conceito: antes de tomar uma ação, deve-se considerar todas as possibilidades de escolha, para então tomar uma decisão de qual escolha é a melhor.

O interpretador para um conjunto de regras de produção consiste da seguinte seqüência de passos, [JAC 86]:

1. combinar os chamados "padrões de regras" contra elementos da memória de trabalho.
2. se há mais de uma regra que poderia ser aplicada, então decidir qual delas aplicar; isto é chamado de "resolução de conflito".

3. aplicar a regra, talvez adicionando um novo item na memória de trabalho ou deletando um item existente, e então voltar para o passo 1.

A maioria dos interpretadores de regras de produção não suportam *backtracking*, pois modificações nas estruturas de dados são destrutivas, dificultando retornar ao estado anterior.

É possível projetar um conjunto de regras tais que, para todas as configurações de dados, somente uma regra seja disparada. Tais conjuntos de regras são chamados determinísticos - se as condições são verdadeiras então as conclusões também são verdadeiras. Apesar disso, a maioria dos conjuntos de regras no qual sistemas especialistas estão interessados são os não-determinísticos, onde não se pode ter absoluta certeza que as condições ou as conclusões são verdadeiras.

O controle de um sistema que utiliza regras de produção pode ser feito tanto para frente (**encadeamento progressivo**) quanto para trás (**encadeamento regressivo**). Esse tipo de controle é feito como se o especialista humano estivesse resolvendo o problema. O encadeamento progressivo começa pelas evidências e tenta determinar que conclusões são deriváveis, já o regressivo inicia assumindo algumas condições como verdadeiras e então usa as regras para tentar prová-las.

5.1.2 Redes Semânticas

A inteligência artificial, com a finalidade de descrever certos tipos de estruturas de dados abstratas, incorporou a terminologia da teoria dos grafos. De acordo com essa teoria, todas as estruturas assumem que existam entidades primitivas chamadas nodos e arcos. Os nodos são as origens e destinos dos arcos e geralmente possuem rótulos para serem distinguidos uns dos outros. Os arcos podem ou não possuir rótulos, dependendo se há ou não mais de um tipo de arco.

Rede semântica é o tipo de rede mais usada para estruturar tipos gerais de informação. Neste modelo tem-se um grafo orientado e rotulado, o qual é composto por um conjunto de nodos e arcos. Os nodos representam conceitos enquanto os arcos definem os relacionamentos entre eles. As ligações mais comuns em redes semânticas são as ligações do tipo "é-um" e "é-parte-de". A figura 5.1 ilustra um exemplo de representação através de uma rede semântica:

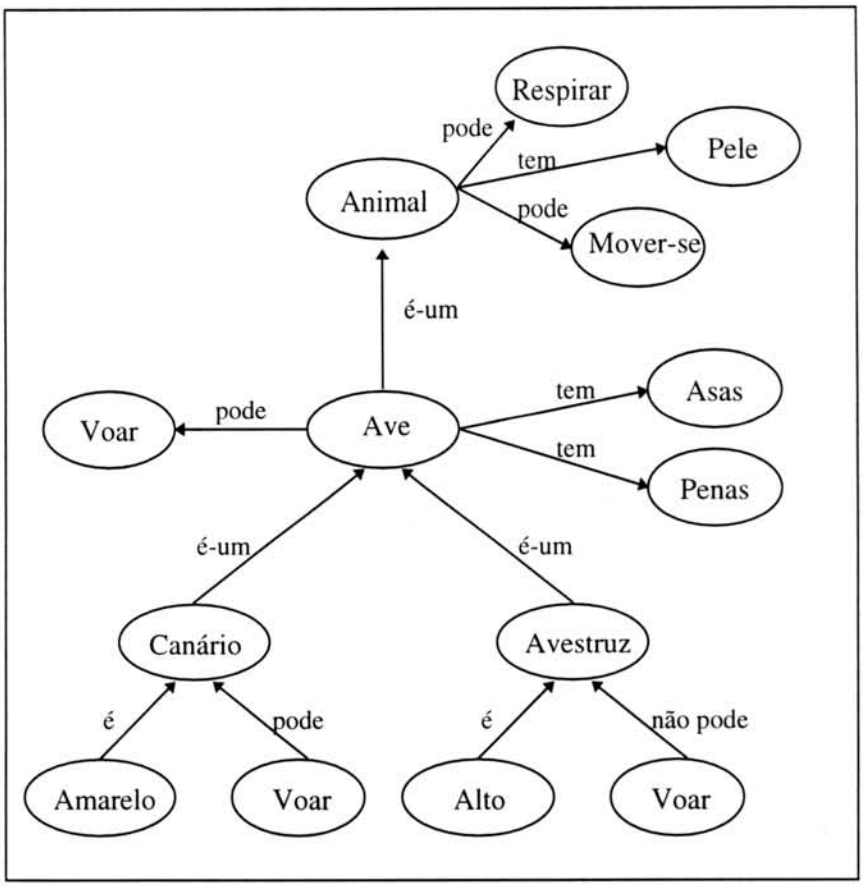

FIGURA 5.1- Exemplo de Rede Semântica.

Não há restrições quanto à forma de designar os nodos e arcos. A flexibilidade é a maior vantagem deste esquema de representação. Novos nodos e arcos podem ser definidos à medida que se precisa.

Segundo [HAR 88], a hereditariedade é outra característica das redes semânticas. Refere-se à capacidade de um nodo "herdar" características de outros nodos com ele relacionados. A hereditariedade de propriedades é uma implicação de uma relação **é-um** e significa que os exemplos de uma classe se supõem herdeiros de todas as propriedades de classes mais gerais de que são membros.

5.2 O MAD

MAD é um módulo de automatização de diagnóstico projetado para indicar ao administrador quando algum problema ocorre, dando conseqüentemente sugestões de possíveis soluções do problema ou recomendações de como determinar sua solução. O MAD é um sub-projeto de um projeto maior que está em desenvolvimento no grupo de redes da UFRGS, o projeto CINEMA, sendo integrado ao Sistema de *Trouble Ticket* e

ao Sistema de Alertas, descritos no seção 4.3.1, conforme figura 5.2. Ele utiliza parte desse projeto para fazer as monitorações e gerar eventos. Os eventos gerados devem então ser analisados para verificar se há necessidade de ser gerado um alerta ao administrador da rede. Caso esta necessidade tenha sido comprovada, um registro de problemas (*trouble ticket*) é gerado contendo o problema encontrado, bem como a possível solução do mesmo. Sua principal característica é a busca de situações críticas com base nas informações que possui à sua disposição.

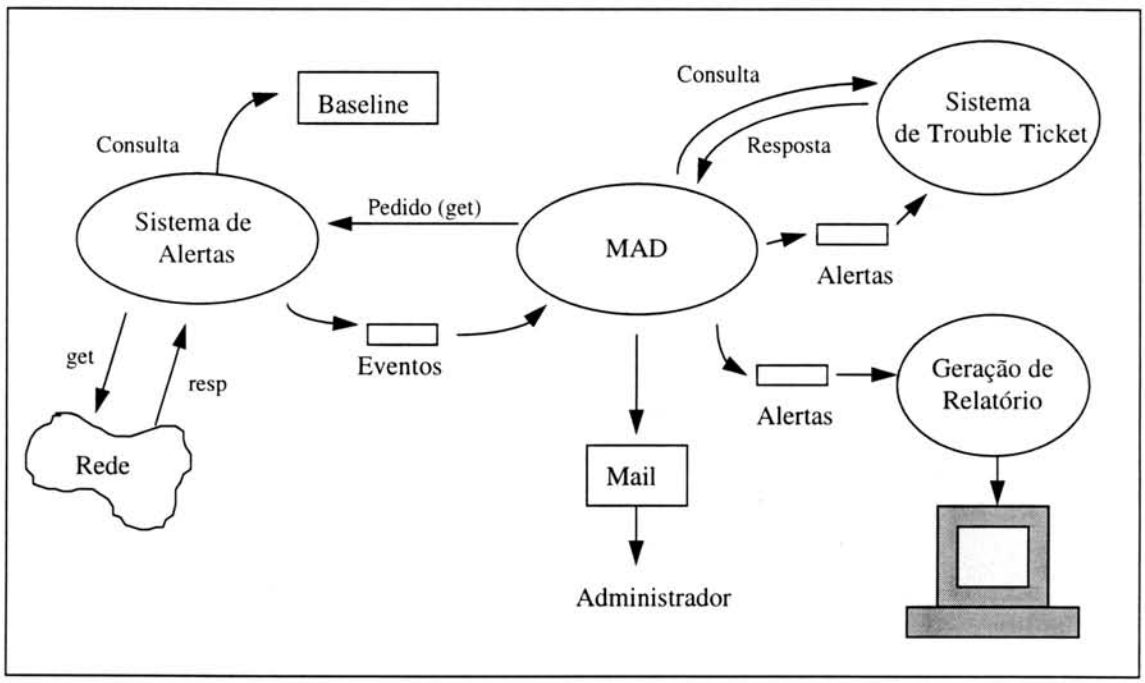

FIGURA 5.2 - Integração do MAD com o CINEMA.

No MAD está armazenado o conhecimento necessário para avaliar as informações obtidas via Sistema de Alertas e discriminar os eventos menores que não são dignos de registro, daqueles que precisam resultar na criação de um registro de problemas e precisam ser sinalizados para o administrador da rede. Através de monitorações contínuas é possível encontrar a utilização corrente dos enlaces e segmentos de rede, identificar áreas de possível congestionamento, isolar altas taxas de erros e examinar os padrões de tráfego da rede. Cada um desses valores pode auxiliar o gerente da rede a assegurar se a rede está ou não atingindo as expectativas de seus usuários.

O paradigma proposto funciona da seguinte forma: objetos da MIB II e máquinas definidas pelo usuário, através de um arquivo de configuração, são monitorados. Neste arquivo de configuração, além das informações citadas acima, o usuário deve informar outros dados relativos à monitoração em questão, como por exemplo, o período de tempo da monitoração. Os valores são amostrados pelo Sistema de Alertas que os decrementa de seus antecessores, dando a certeza ao usuário de que os

valores obtidos refletem a evolução do indicador desde o início da amostragem. A maioria dos objetos da MIB II interessantes de serem monitorados são objetos contadores, portanto é necessário fazer sua diminuição com o valor coletado anteriormente.

Após a coleta destas informações, o Sistema de Alertas faz uma análise estatística e gera eventos se algum dos objetos ultrapassar um determinado parâmetro, denominado "limite". Cada evento gerado só é indicado ao usuário se o mesmo se transforma em um alerta. Essa transformação de eventos em alertas é feita pelo MAD. Para que isso ocorra, é preciso descobrir se os eventos são críticos e irão afetar a rede de forma grave. Quando alertas são gerados, o administrador fica sabendo que um problema ocorreu com a rede e que deve ser corrigido o mais rápido possível.

Para o sistema saber quais parâmetros são normais na rede, é preciso criar uma referência chamada de *baseline*. A *baseline* indica a atividade da rede por um determinado período de tempo e identifica o desempenho normal dessa rede nas diversas horas do dia através de médias e desvios padrão. Com esses dados pretende-se obter um perfil da operação de uma determinada rede. A *baseline* que foi obtida da rede da UFRGS é composta por determinados objetos da MIB II que podem ajudar no diagnóstico de falhas e na degradação do desempenho da rede da universidade. Portanto, foram selecionadas quatorze máquinas (bb2.pop-rs.rnp.br, penta, routcv, routcc, caracol, minuano, chuí, rígel, cristal, castor, if1.ufrgs.br, vortex, ipe.srh.ufrgs.br e tchepoa) para serem monitoradas durante um mês. As monitorações foram efetuadas de uma em uma hora, obtendo-se o valor de objetos específicos. Os objetos que compõem a *baseline* encontram-se no Anexo A-1. A topologia da rede da UFRGS está ilustrada nas figuras do Anexo A-3.

As máquinas que foram monitoradas possuem mais de uma interface. A monitoração de cada uma delas é possível através dos objetos do grupo interfaces da MIB II. É preciso verificar qual o número de interfaces que determinada máquina possui e então monitorá-la separadamente utilizando os objetos com as extensões adequadas.

Todos os objetos monitorados que são contadores aumentam seu valor com o passar do tempo e são reinicializados quando atingem um valor máximo igual a $2^{32} - 1$ (4.294.967.295). Por se tratar de objetos contadores, os valores coletados de cada objeto devem ser decrementados uns dos outros, para cada dia de monitoração, tendo-se assim o quanto realmente esse contador aumentou ou diminuiu. Feito isso, calcula-se a média e o desvio padrão de todos os objetos, formando assim a *baseline* da rede.

Para utilizar o Sistema de Alertas foi necessário implementá-lo, uma vez que apenas a especificação do sistema tinha sido definida anteriormente. Além disso, foi necessário fazer uma alteração no sistema original na parte de comparação com os limites. Como uma *baseline* foi criada para a rede, os parâmetros com os quais os valores coletados devem ser comparados são obtidos dessa *baseline* caso o objeto que está sendo monitorado encontrar-se nela. Caso contrário, o procedimento especificado pelo Sistema de Alertas é utilizado, ou seja, calcula-se os limites automaticamente.

Quando um evento chega ao MAD o primeiro passo a ser dado é localizar o equipamento, que foi monitorado, dentro da rede. Se um equipamento está mais

próximo da saída da rede para o mundo externo e existem várias máquinas dependendo dele para sair da instituição, ele é considerado um equipamento crítico. Diz-se que este equipamento é crítico pois várias outras máquinas estão subordinadas a ele. Se algum problema grave ocorrer nesta máquina, o acesso externo ficará prejudicado e os usuários não ficarão satisfeitos com a qualidade de serviço prestado pela rede. Após o equipamento ter sido localizado, utiliza-se um conjunto de regras para determinar se um registro de problemas precisará ser criado, alertando o administrador da rede sobre o problema encontrado. Portanto, definiu-se prioridades para as máquinas de acordo com a sua localização na rede. Máquinas que estão mais próximas da saída possuem prioridades mais baixas e máquinas que estão mais distantes possuem prioridades mais altas. Essas prioridades são utilizadas pelas regras para definir se um registro de problemas deve ser criado naquele momento ou se deve-se aguardar um determinado tempo para que o problema encontrado seja confirmado.

Para tornar mais fácil a localização de determinado equipamento na rede da UFRGS criou-se uma tabela, tabela 5.1, onde estão listadas algumas máquinas importantes para a instituição e que foram monitoradas (formam a *baseline*), indicando seu grau de importância para a instituição. Regras diferentes são utilizadas para a geração de um alerta dependendo da importância do equipamento na rede.

TABELA 5.1 - Prioridades definidas para cada máquina monitorada.

Máquina	Prioridade
bb2.pop-rs.rnp.br	1
tchepoa.ufrgs.br	1
routcc.ufrgs.br	1
routcv.ufrgs.br	1
caracol.inf.ufrgs.br	1
penta.ufrgs.br	2
vortex.ufrgs.br	2
ipe.srh.ufrgs.br	2
if1.ufrgs.br	2
rigel.inf.ufrgs.br	2
cristal.inf.ufrgs.br	2
castor.inf.ufrgs.br	2
chui.inf.ufrgs.br	2
minuano.inf.ufrgs.br	2

O equipamento mais crítico para a saída ao mundo externo é o roteador bb2.pop-rs.rnp.br, pois se algum problema afetar seu funcionamento fazendo-o parar de funcionar ou entregar os pacotes mais lentamente, toda a rede da UFRGS será atingida.

Dependendo do tipo de problema diagnosticado, formas diferentes são utilizadas para determinar se um alerta deve ser criado ou não. Algumas vezes um problema pode ser originado por vários fatores, portanto é preciso descobrir quais são esses fatores para então apresentar uma possível solução ao administrador.

Quando for determinada a necessidade de se gerar um alerta, o sistema deve então consultar a base de registro de problemas para descobrir se o mesmo problema já aconteceu anteriormente para aquela máquina. Caso algum registro seja encontrado, sua solução é adicionada ao novo registro criado.

Além da geração de alertas e criação de registros de problemas, o sistema também possui a capacidade de gerar relatórios dos últimos alertas gerados. À medida que alertas vão surgindo eles são armazenados em um *log*. Pode-se escolher dois tipos de relatório, um que mostra todo o *log*, isto é, todos os alertas gerados, e outro que mostra apenas alertas de uma determinada máquina.

Para se ter uma qualidade de serviço aceitável em uma rede é importante determinar quais são os recursos críticos que devem ser constantemente analisados. Pode-se determinar os recursos críticos de duas formas:

- alguém informa quais máquinas e objetos devem ser monitorados, ou,
- o sistema descobre por aprendizagem percebendo pela análise do registro de problemas passado quais os recursos que, quando afetados, perturbam uma porção mais significativa da rede.

Abaixo estão listadas informações que podem ser detectados a partir da utilização do sistema:

- pacotes descartados
- percentual de utilização de uma interface
- taxa de erros
- taxa de pacotes *broadcast*
- estado da interface
- reinicialização constante do roteador
- percentual de datagramas IP
- percentual de mensagens ICMP
- percentual de segmentos TCP

5.2.1 A representação dos problemas

Como foi visto no início deste capítulo, existem várias formas de representar o conhecimento quando se utiliza um sistema especialista. Uma dessas formas é a representação por meio de redes semânticas. Esta é uma forma gráfica de representar o conhecimento e foi utilizada neste trabalho para melhorar a visualização de alguns tipos de problemas que podem ocorrer com uma rede tipo a da UFRGS. Todos os problemas indicados neste capítulo podem ser descobertos a partir da monitoração de objetos da MIB II.

A figura 5.3 fornece uma visão global de alguns problemas que podem ser encontrados na rede e mostra a forma como eles foram separados de acordo com sua origem. De uma forma mais geral, os problemas em redes de computadores podem ser originados por três motivos:

- quando o problema está no meio físico ou nas interfaces (a causa é física);
- quando o problema está nas entidades locais (originados pela própria máquina sendo monitorada); ou,
- quando o problema está nas entidades remotas (originados por equipamentos remotos à máquina sendo monitorada).

Cada problema indicado na figura 5.3 foi analisado separadamente para que suas causas fossem identificadas. As figuras subseqüentes, figura 5.4, 5.7 e 5.10, mostram a especificação de cada causa separadamente (causa remota, local e no meio e interfaces). A causa de alguns problemas pode ser devida a problemas internos dos protocolos. As demais figuras, figura 5.5, 5.6, 5.8 e 5.9, apresentam a lógica de diagnóstico para problemas remotos e locais.

É claro que muitos outros problemas podem ser encontrados em uma rede, mas o objetivo deste trabalho é apresentar apenas problemas que podem ser descobertos através de consultas em objetos da MIB II. A descrição de cada problema está descrita na seção subseqüente, seção 5.2.2.

Todos os nodos finais das figuras possuem os seguintes atributos comuns:

- limiar monitorado: inteiro
- severidade: [alta, média, baixa]
- recomendações/comentários: texto

Os atributos de cada nodo são diferenciados pelo valor que cada um deve possuir. A descrição dos atributos "limiar monitorado" e "recomendações/comentários" está apresentada nas tabelas 5.2, 5.3, 5.4 e 5.6 da seção subseqüente. Esses atributos não foram especificados nas figuras por falta de espaço e porque eles prejudicariam a visibilidade das redes semânticas. As redes semânticas representam o relacionamento

entre os objetos da MIB II. Cada nodo especificado nas figuras corresponde à monitoração de um determinado objeto da MIB II.

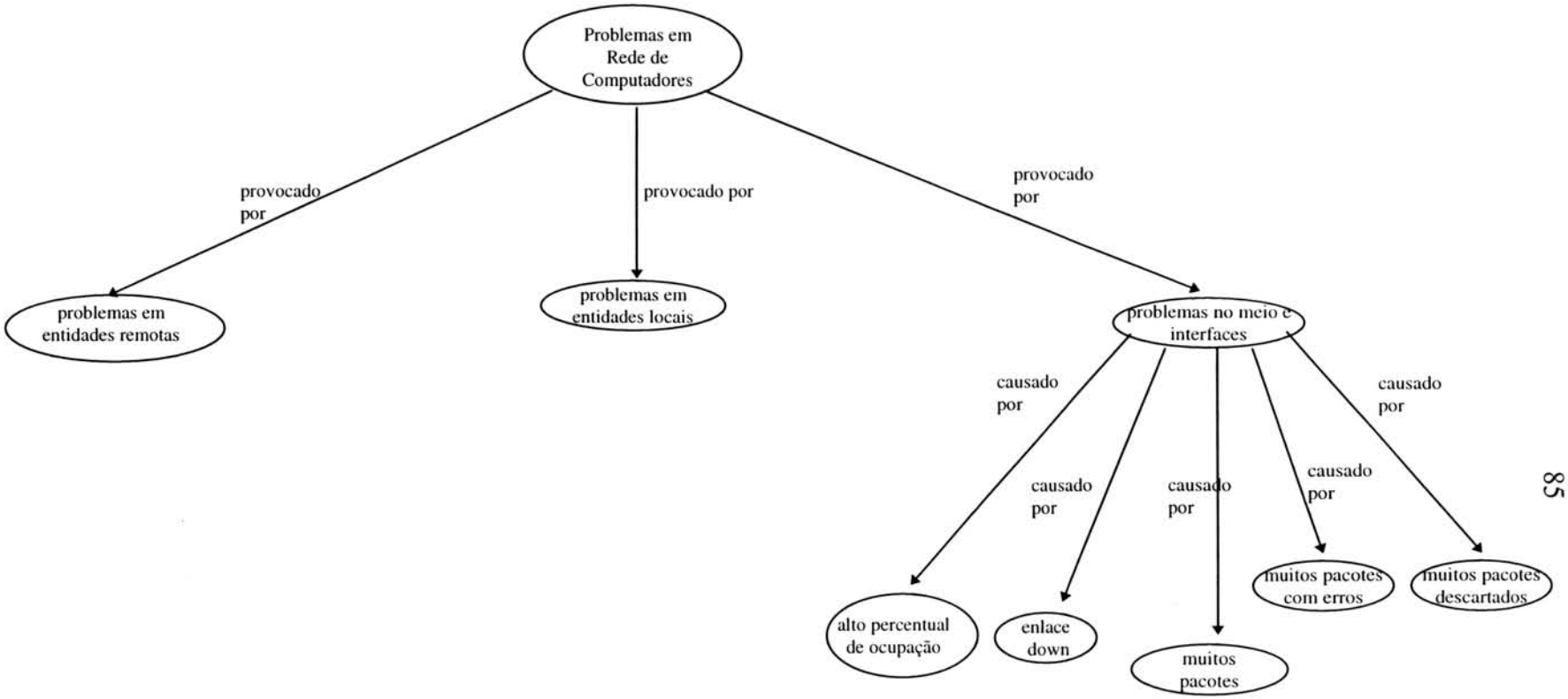

FIGURA 5.3 - Rede semântica geral dos problemas.

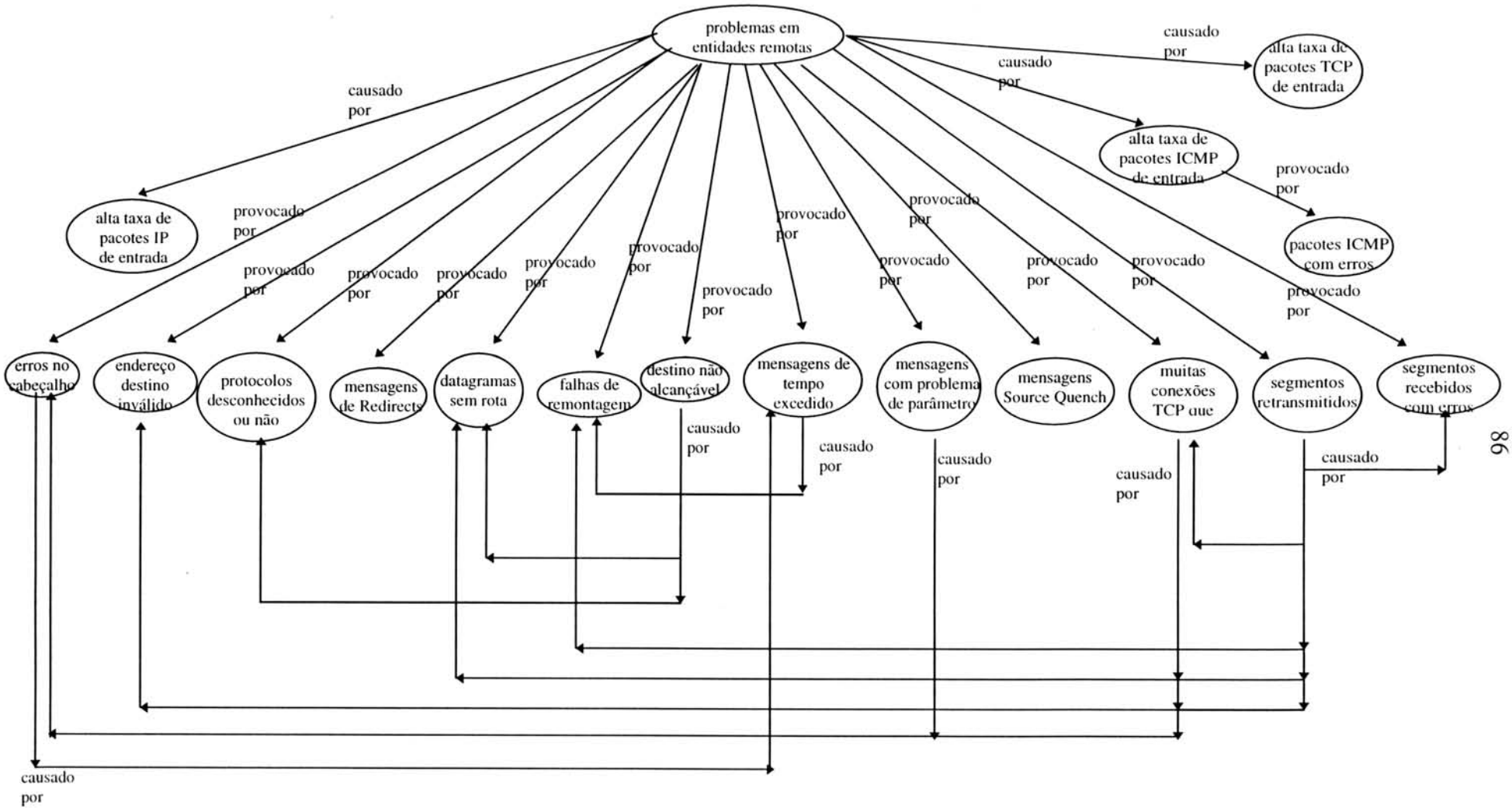

FIGURA 5.4 - Rede semântica dos problemas em entidades remotas.

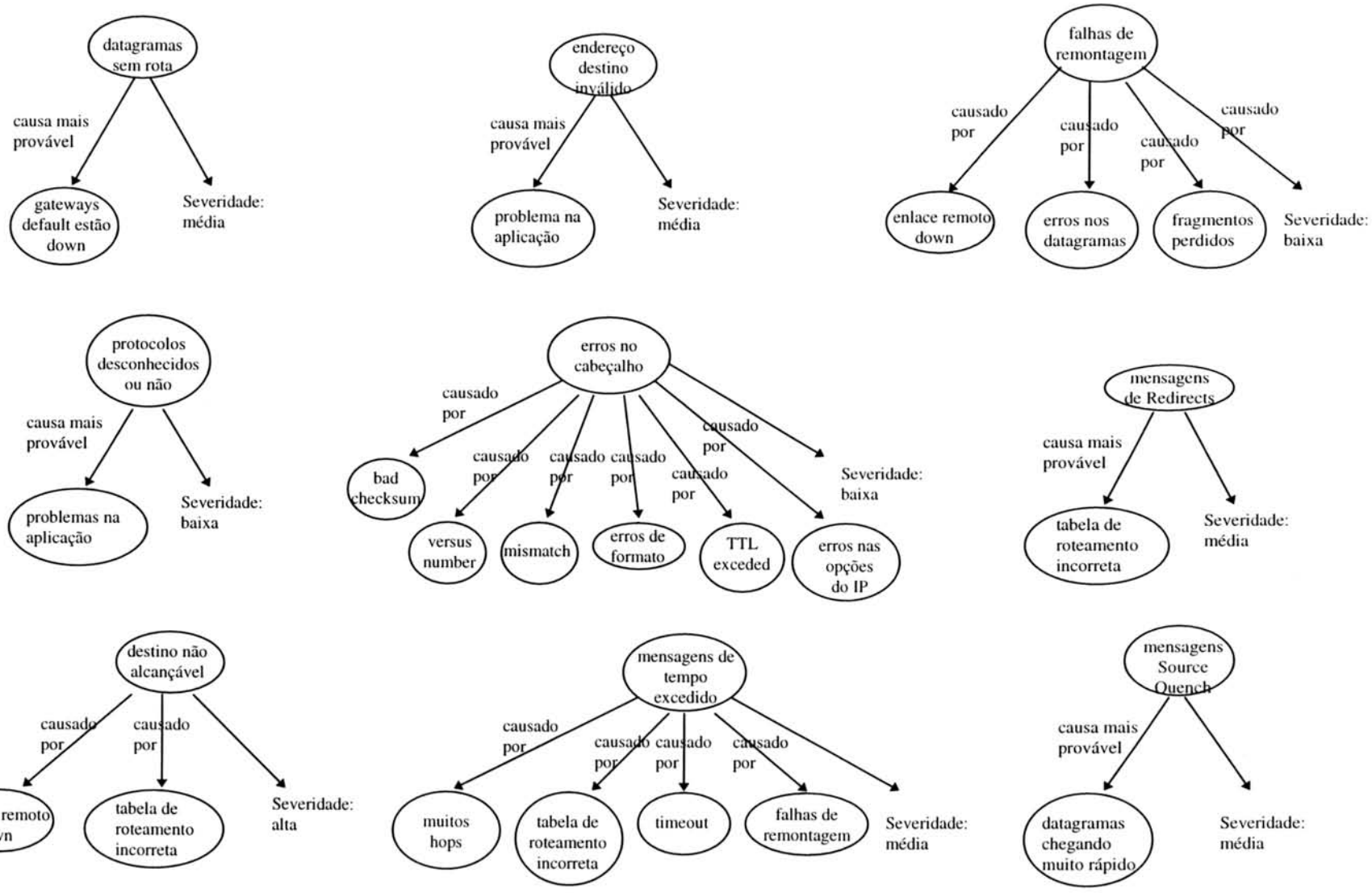

FIGURA 5.5 - Lógica de diagnóstico dos problemas em entidades remotas - Parte 1.

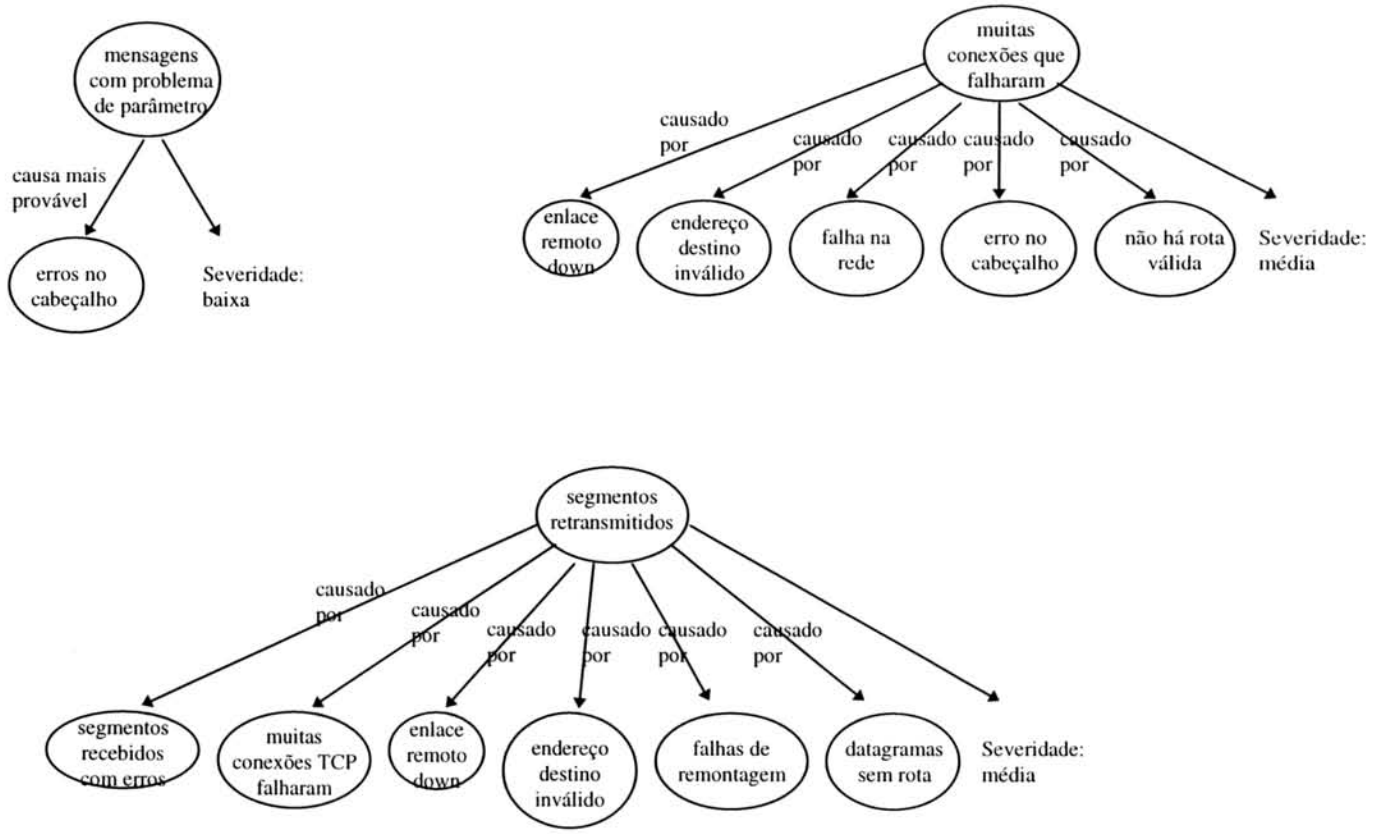

FIGURA 5.6 - Lógica de diagnóstico dos problemas em entidades remotas - Parte 2.

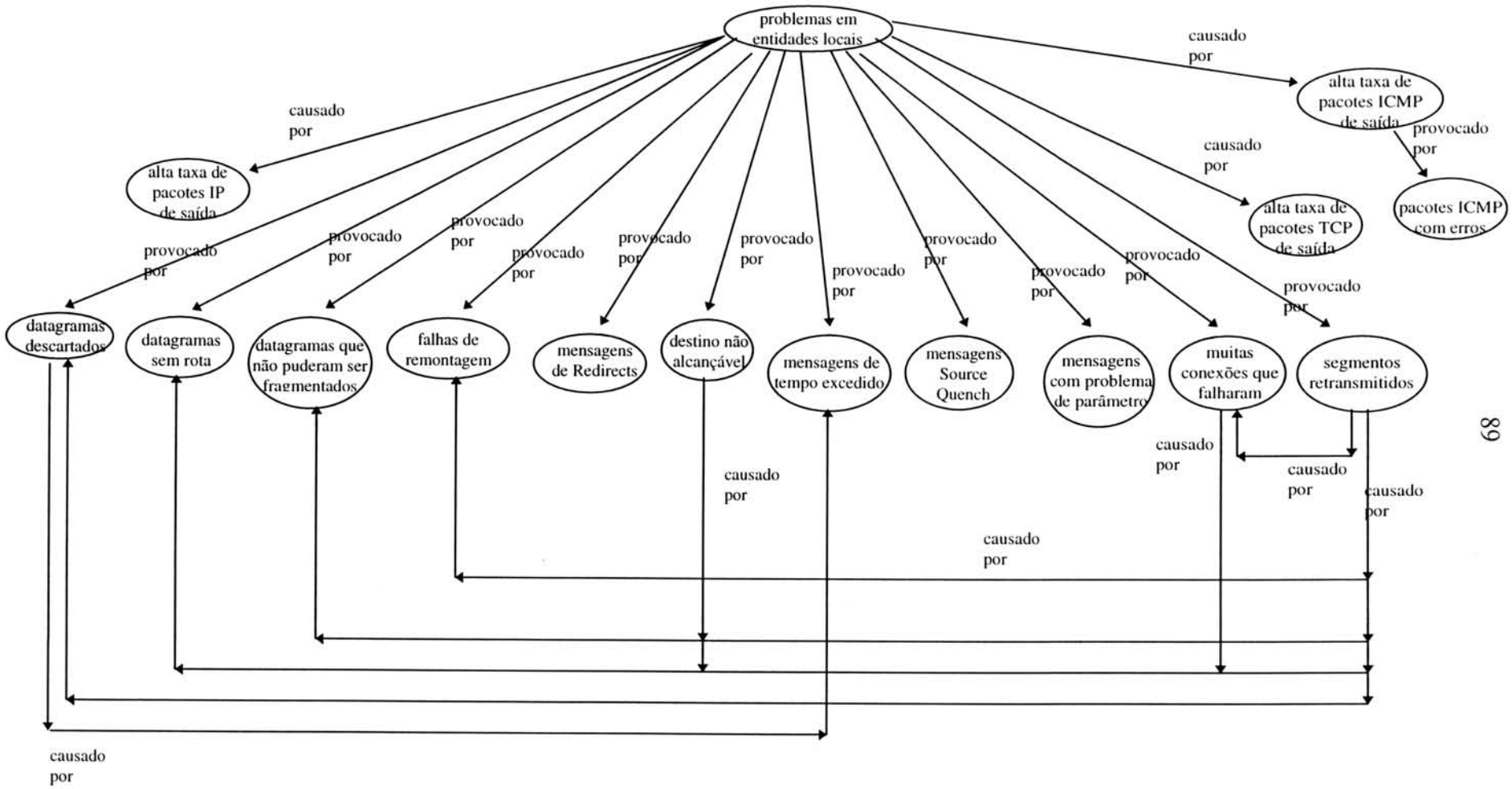

FIGURA 5.7 - Rede semântica dos problemas em entidades locais.

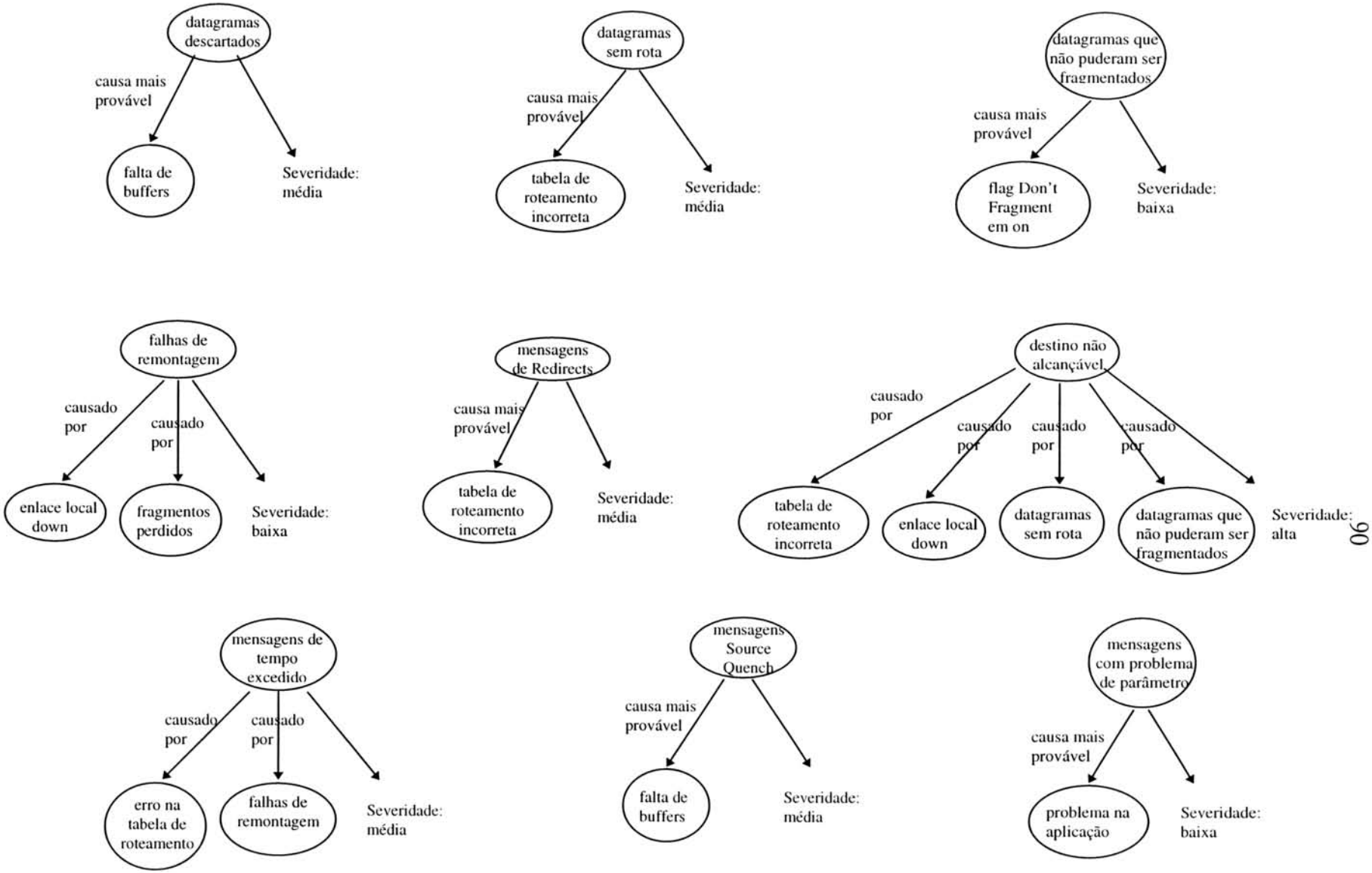

FIGURA 5.8 - Lógica de diagnóstico dos problemas em entidades locais - Parte 1.

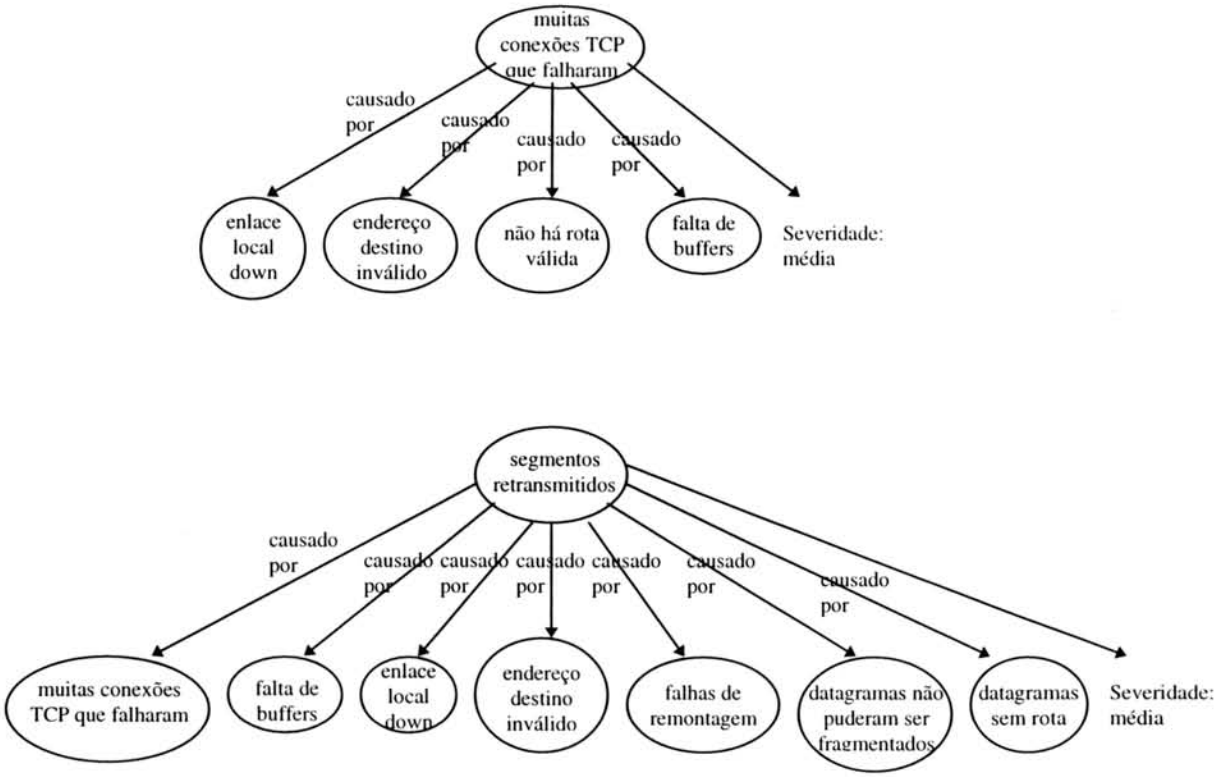

FIGURA 5.9 - Lógica de diagnóstico dos problemas em entidades locais - Parte 2.

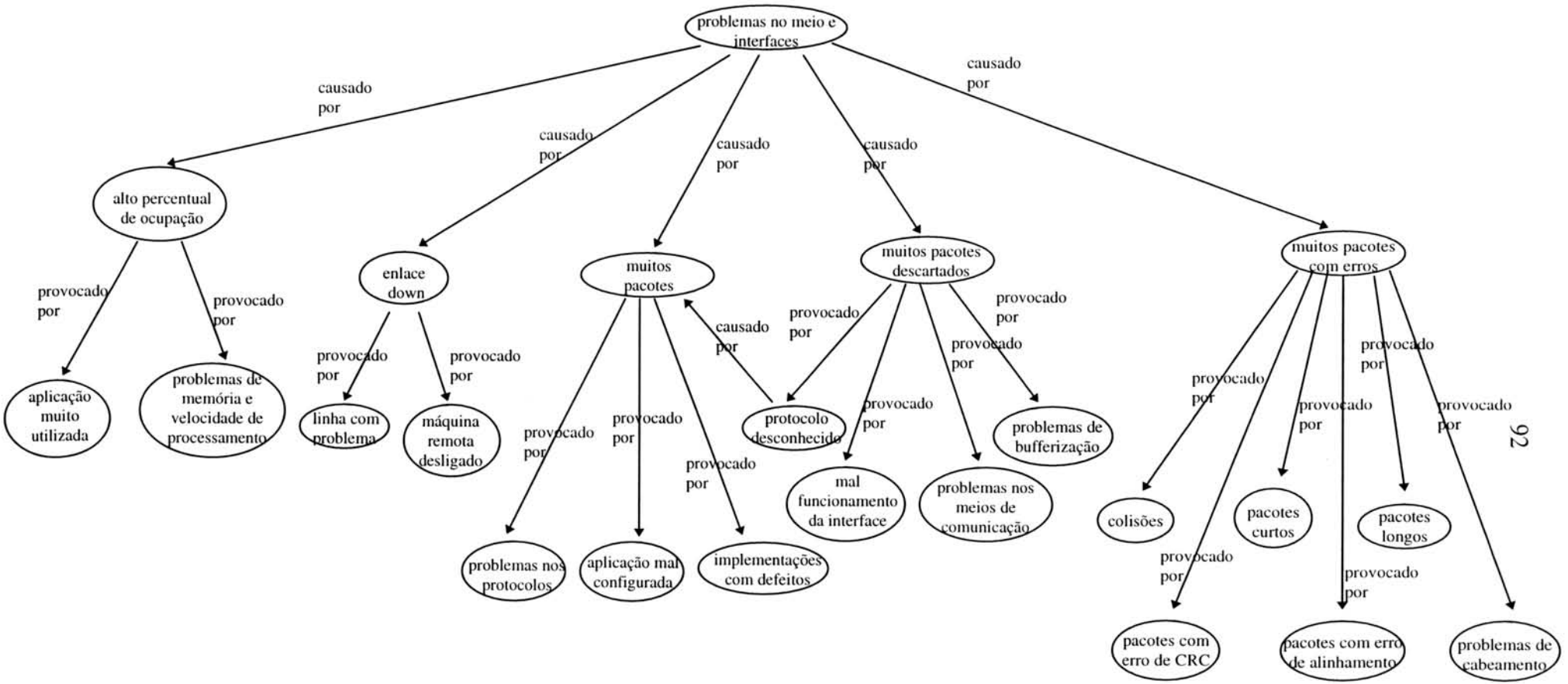

FIGURA 5.10 - Rede semântica dos problemas nos meios e interfaces.

5.2.2 Regras utilizadas pelo MAD

Alguns problemas descritos no capítulo 2 podem ser diagnosticados por meio de monitorações em objetos da MIB II. Um exemplo que pode ser observado é quando a monitoração do objeto *ipOutNoRoutes* apresentar-se com um percentual muito elevado. Neste caso, pode estar ocorrendo algum problema na tabela de roteamento da máquina.

Sempre que um evento é gerado decorrente de um limite que foi ultrapassado, é preciso verificar se esse evento é grave para então criar um alerta que será apresentado à pessoa responsável. Uma forma de descobrir se eventos devem ser transformados em alertas é passar os valores coletados por uma base de regras. Pensando nisso, regras diferentes foram criadas dependendo do evento gerado.

Uma regra comum para todos os eventos é aquela utilizada para descobrir a prioridade da máquina que está sendo monitorada. Para tanto, consulta-se a tabela 5.1 na qual as prioridades foram designadas a cada máquina dependendo de sua localização na rede. Feito isso, ações diferentes são tomadas até que se crie um alerta.

É importante tentar diferenciar rajadas que podem ocorrer eventualmente na rede e não indicam necessariamente um problema, de condições que persistem e provavelmente estão causando algum problema. Um forma de identificar rajadas é monitorando a rede por mais algum tempo depois que um evento for recebido. Caso seja identificado que o problema continua durante esse tempo, então um alerta deve ser criado. Todas as regras descritas a seguir levam isso em consideração.

A base de regras criada com base nas redes semânticas das figuras 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9 e 5.10 está apresentada abaixo na forma de tabelas: 5.2, 5.3, 5.4 e 5.5. O percentual utilizado pelas regras foi calculado de acordo com a *baseline* da rede. A base de regras criada é produto de extensa interrogação realizada no âmbito deste trabalho, visando determinar critérios para distinguir um comportamento normal de uma situação que precisa ser diagnosticada. Para chegar ao conjunto de regras foi coletado tanto o conhecimento formal, recomendações contidas em livros e artigos, [MIL 91] [MIL 89] [SNG 90], quanto conhecimento empírico dos administradores de redes da UFRGS, que foram entrevistados e consultados.

Uma regra que não se encontra nesta tabela diz respeito ao estado da interface. Uma interface pode estar com o seu estado operacional em *down*, mas seu estado administrativo, estado em que ela deveria estar, encontrar-se em *up*. Esta regra está definida mais adiante neste capítulo. Outra regra definida mais adiante é a que trata da reinicialização do roteador.

A maioria das regras definidas neste trabalho são percentuais do total de tráfego. O total de tráfego corresponde ao somatório do número de pacotes que estão entrando e saindo pelas interfaces de determinada máquina.

TABELA 5.2 - Características das regras implementadas.

Regra	Limiar monitorado	Recomendações/Comentários
pacotes descartados	maior que 1% do tráfego total para interface Ethernet e *ifInUnknownProtos* menor que 1%.	Quando este limiar for atingido recomenda-se verificar cabos, conectores e interfaces da máquina. Caso o problema persista, recomenda-se verificar a memória da máquina e se necessário deve-se incrementá-la.
percentual de utilização	maior que 50% do tráfego total para interface Serial. Ou maior que 30% para interface Ethernet das demais máquinas.	Em primeiro lugar recomenda-se verificar se determinada aplicação está sendo muito utilizada. Recomenda-se também incrementar a memória da máquina monitorada, caso necessário, e ainda confinar o tráfego criando outro segmento.
taxa de erros	maior que 2% do tráfego total para interface Ethernet. Ou maior que 5% do tráfego total para interface Serial.	Recomenda-se verificar se a taxa de pacotes com erros de CRC e alinhamento e pacotes longos e curtos está muito elevada (maior que 1% do tráfego total). Esses erros podem estar ocorrendo por falhas em *transceivers* ou *drivers* que controlam a placa, problemas no cabeamento, entre outras. Ou então a interface está enviando/recebendo pacotes maiores do que sua capacidade.
taxa de *broadcast*	maior que 8% do tráfego total para interface Ethernet.	Recomenda-se tentar identificar qual a máquina do segmento monitorado que está enviando muitos pacotes *broadcast*. Feito isso, deve-se verificar a configuração dessa máquina a nível de endereçamento e aplicação. Caso a configuração esteja correta, o problema pode estar nos protocolos ou em implementações com defeito.
percentual de pacotes ICMP	maior que 5% do tráfego total.	Este percentual foi calculado a partir da *baseline* da rede. Quando este percentual for atingido, o sistema

		investiga porque isso ocorreu e então envia as recomendações necessárias dependendo do problema que foi identificado. Ver tabela 5.3.
percentual de pacotes IP	maior que 10% do tráfego total.	Este percentual foi calculado a partir da *baseline* da rede. Neste caso o sistema investiga porque determinado limiar foi atingido e então envia as recomendações necessárias dependendo do problema que foi identificado. Ver tabela 5.4.
percentual de pacotes TCP	maior que 10% do tráfego total.	Este percentual foi calculado a partir da *baseline* da rede. Neste caso o sistema investiga porque determinado limiar foi atingido e então envia as recomendações necessárias dependendo do problema que foi identificado. Ver tabela 5.5.

Quando a taxa de pacotes ICMP ultrapassar o limiar definido como normal para uma determinada máquina as regras apresentadas na tabela 5.3 serão utilizadas. Essas regras também podem ser utilizadas separadamente uma vez que a taxa de algum tipo de mensagem ICMP pode estar muito elevada, mas não chega a aumentar significativamente a taxa total de mensagens ICMP recebidas ou enviadas.

TABELA 5.3 - Regras para percentual de pacotes ICMP.

Regra	**Limiar monitorado**	**Recomendações/Comentários**
pacotes ICMP com erros	maior que 1% do tráfego total.	Quando este percentual for ultrapassado, o administrador da rede é informado do acontecido. Um alerta é criado avisando que estão ocorrendo muitos erros em mensagens ICMP. A causa desses erros pode ser por falta de *buffers*, *checksums* errados, comprimento incorreto da mensagem ou o sistema origem pode estar encapsulando segmentos de forma errada.

destino não alcançável	maior que 1% do tráfego total.	Para descobrir a causa deste tipo de mensagem deve-se verificar o estado do enlace da entidade, a taxa de *ipOutNoRoutes* e a taxa de *ipFragFails*, (ver tabela 5.4). O sistema faz essas verificações automaticamente. Caso nenhuma dessas causas seja identificada, o sistema emite a seguinte recomendação: tentar identificar o endereço não alcançável para descobrir a causa desse número elevado de mensagens (pode-se tentar descobrir a causa com os comando "ping" e "traceroute").
tempo excedido	maior que 1% do tráfego total.	Para este tipo de mensagem o sistema verifica a taxa de *ipInHdrErrors* e *ipReasmFails* (ver tabela 5.4). Caso a taxa do primeiro objeto esteja elevada significa que o campo TTL foi zerado (devido a tabelas de roteamento mal configuradas ou muitos *hops* até o destino). Caso o problema seja encontrado no segundo objeto, significa que datagramas não puderam ser remontados pois fragmentos foram perdidos.
problemas de parâmetros	maior que 1% do tráfego total.	Quando este percentual está elevado, o sistema verifica a taxa de *ipInHdrErrors* (ver tabela 5.4). Geralmente esse problema é devido a algum problema na aplicação que está preenchendo o cabeçalho do datagrama IP incorretamente.
source quench	maior que 1% do tráfego total.	Dois motivos podem causar este tipo de mensagem: pacotes descartados por falta de *buffer* (*ipInDiscards*) ou quando datagramas estão chegando mais rápido do que a entidade pode processá-los.
redirect	maior que 1% do tráfego	Neste caso recomenda-se verificar

	total.	as tabelas de roteamento das máquinas.

Algumas regras também foram definidas para encontrar a causa de pacotes IP e TCP que ultrapassam os limiares definidos para cada máquina. As tabelas 5.4 e 5.5 apresentam os limiares para cada uma dessas regras, bem como a recomendação que é enviada com o alerta gerado. Assim como na tabela acima, essas regras podem ser utilizadas separadamente.

TABELA 5.4 - Regras para percentual de pacotes IP.

Regra	**Limiar monitorado**	**Recomendações/Comentários**
datagramas descartados	maior que 1% do tráfego total.	O descarte excessivo de datagramas pode ser ocasionado pela falta de *buffers* para armazená-los. Neste caso recomenda-se verificar a memória disponível na máquina e aumentá-la se necessário.
erros de cabeçalho	maior que 1% do tráfego total.	Datagramas com erros de cabeçalho incluem: *bad checksum, versus number, mismatch, other format errors, time-to-live exceded, errors discover improving their IP options,* etc. Neste caso, gera-se um alerta informando que pode ter ocorrido alguma falha na rede, que a aplicação pode estar com algum problema ou que a tabela de roteamento pode estar incorreta.
erros de endereçamento	maior que 1% do tráfego total.	Esses erros podem ser devido a uma aplicação fornecer um endereço inválido ou algum erro na tabela de roteamento.
protocolos desconhecidos	maior que 1% do tráfego total.	Geralmente causado por pacotes *broadcast* (ver tabela 5.2).
rota desconhecida	maior que 1% do tráfego total.	A causa desta alta taxa é devido a problemas de configuração nas tabelas de roteamento ou quando todos os *gateways default* estão *down*. Neste caso, uma verificação

		do estado do enlace deve ser feita. Além disso, recomenda-se tentar identificar o endereço cuja rota não existe pois esse endereço pode não existir. Deve-se também verificar se existe rota default em sua tabela de roteamento.
falhas de remontagem	maior que 1% do tráfego total.	Falhas de remontagem podem ser devido a: enlace *down*, erros ou fragmentos perdidos. Deve-se verificar o enlace por onde os datagramas são recebidos. Caso ele esteja *down*, este tipo de falha pode ocorrer.
problemas na fragmentação	maior que 1% do tráfego total.	Este tipo de problema ocorre quando datagramas precisam ser fragmentados, mas seu *flag Don't Fragment* está *on*.

TABELA 5.5 - Regras para percentual de pacotes TCP.

Regra	**Limiar monitorado**	**Recomendações/Comentários**
falha na tentativa de estabelecer conexão	maior que 1% do tráfego total.	Este tipo de falha pode ocorrer por uma série de razões: erros no cabeçalho do datagrama IP (*ipInHdrErrors*), endereço destino inválido (*ipInAddrErrors*), não há rota válida (*ipOutNoRoutes*), falta de *buffers* (*ipInDiscards*), enlace inativo, entre outras. O sistema verifica automaticamente qual desses problemas estão ocorrendo antes de gerar um alerta.
segmentos retransmitidos	maior que 1% do tráfego total.	Como no caso anterior, o sistema verifica as causas deste alto percentual automaticamente. Segmentos retransmitidos podem ser ocasionados por muitas razões: muitos segmentos com erros (*tcpInErrs*), falhas na tentativa de estabelecer conexão

		(*tpcAttemptFails*), destino inativo, falta de *buffers* (*ipInDiscards*), endereço destino inválido (*ipInAddrErrors*), falhas de remontagem (*ipReasmFails*), falhas na fragmentação (*ipFragFails*), datagramas sem rota (*ipOutNoRoutes*), entre outras. Ver tabela 5.4. O sistema envia uma recomendação diferente dependendo do problema diagnosticado.

A descrição de cada regra está especificada abaixo.

5.2.2.1 Pacotes descartados

A ocorrência de pacotes descartados pode degradar o desempenho de uma rede quando este nível for muito elevado, ultrapassando o que é "normal" para ela. Conforme definido em [LEI 93], pacotes descartados podem resultar de uma variedade de razões:

- mal funcionamento da interface;
- problemas nos meios de comunicação;
- problemas de armazenamento no dispositivo.

Além disso, é muito importante ter em mente que nem sempre pacotes descartados representam um problema. Por exemplo: um dispositivo pode ter uma percentagem alta de pacotes descartados porque ele está recebendo muitos pacotes que são de protocolos desconhecidos. Tal valor pode ser encontrado através de monitorações no objeto *ifInUnknownProtos*.

Em uma rede pode-se ter equipamentos que roteiam somente o Protocolo Internet (IP). Esses equipamentos podem ter uma interface Ethernet por meio da qual mensagens são trocadas, incluindo mensagens de *broadcast*. Quando as interfaces recebem pacotes *broadcast*, elas conseqüentemente recebem muitos pacotes que não sabem como processar e então descartam-os. Quando isso acontece, o número de pacotes descartados (*ifInDiscards*) aumenta proporcionalmente com o número de pacotes com protocolos desconhecidos (*ifInUnknownProtos*). Como pode-se verificar nesta situação, mesmo o número de *ifInDiscards* e *ifInUnknownProtos* ser elevado não significa que um problema esteja ocorrendo.

Através de monitorações contínuas sobre a rede pode-se observar que o número de pacotes descartados de uma hora para outra permanece geralmente constante, variando muito raramente. Portanto, normalmente este percentual não deve ultrapassar

1% do tráfego total. Quando isto ocorre, deve-se verificar se estão sendo recebidos muitos pacotes com protocolos desconhecidos. Esses pacotes geralmente são causados por pacotes *broadcast*. Caso não seja este o caso e esta taxa continue alta por mais algum tempo, um registro de problemas deve ser criado. As recomendações incluídas na abertura de um registro de problemas são as seguintes:

- verificar cabos e conectores;
- verificar as interfaces da máquina;
- verificar a quantidade de memória disponível na estação sendo monitorada e incrementá-la quando necessário. Pacotes podem estar sendo descartados por falta de memória para alocá-los.

Normalmente, pacotes descartados indicam que a rede tem mais tráfego do que ela pode manipular ou que pode estar havendo algum problemas nas tabelas de roteamento.

Quando *ifOutDiscards* for utilizado em conjunto com *ifOutOctets* tem-se uma indicação se há ou não um congestionamento na rede. Se uma entidade está descartando muitos pacotes que estão tentando sair pela interface, como indicado por *ifOutDiscards*, e o número total de bytes de saída está sendo decrementado, como mostrado por *ifOutOctets*, a interface pode estar congestionada.

Os descartes de saída são considerados mais sérios do que os de entrada. Enquanto descarte de entrada pode ser um evento normal, um descarte de saída pode ser um problema. Um pacote que está saindo da entidade será descartado sempre que a entidade não souber como enviá-lo.

5.2.2.2 Percentual de Utilização

Quando se deseja otimizar uma rede ou mesmo verificar se ela está muito sobrecarregada, é importante conferir a utilização da largura de banda. Calculando-se a utilização de um enlace é possível isolar problemas de desempenho correntes ou ajudar a evitar congestionamento através do planejamento da capacidade da rede. Através de muitas análises e técnicas específicas é possível medir o que é considerado "normal", bem como qual é o pico para uma rede particular. O pico é o maior valor que foi coletado durante o tempo de monitoração. Esta medida pode variar dependendo da rede, devido a diferença entre ambientes, tais como o tipo de aplicação e o número de componentes e usuários .

O percentual de utilização de uma rede vai depender principalmente da capacidade efetiva de transmissão na rede, dependendo com isso da velocidade que uma determinada interface é capaz de transmitir. Para uma rede Ethernet costuma-se utilizar um percentual de 30% como um nível máximo de utilização média, e um percentual de 55% para casos isolados onde em determinado momento este valor foi atingido, baixando logo em seguida. Este percentual de 55% é considerado o pico.

Quando esses percentuais são ultrapassados o desempenho da rede degrada prejudicando a qualidade de serviço prestada ao usuário. A utilização do enlace dependerá de muitos fatores incluindo o protocolo básico de enlace dos dados, o algoritmo de retransmissão sendo utilizado pelos *hosts* e as aplicações usando o enlace. A perda de desempenho é normalmente caracterizada pelo alto tempo de resposta. O percentual de ocupação também pode ser influenciado pelos seguintes fatores:

- os servidores estarem muito lentos;
- o número de usuários aumentar em determinado segmento de rede;
- uma determinada aplicação começar a ser muito utilizada.

Através de monitorações no grupo interfaces da MIB II pode-se encontrar o percentual de utilização para um único equipamento em um meio *multicast* como Ethernet. O cálculo do percentual de ocupação é feito da seguinte forma (x e y são segundos):

$$total\_de\_bytes = (ifInOctets_y - ifInOctets_x) + (ifOutOctets_y - ifOutOctets_x)$$

$$total\_de\_bytes\_por\_segundo = \frac{total\_de\_bytes}{(y-x)}$$

$$utilização = \frac{(total\_de\_bytes\_por\_segundo * 8)}{ifSpeed}$$

O valor 8 é utilizado para converter a unidade de bits (*ifInOctets* e *ifOutOctets*) para bytes (*ifSpeed*).

Com o uso dos mesmos objetos pode-se calcular a utilização de um enlace ponto-a-ponto full-duplex. Para este cálculo deve-se utilizar o maior dos bytes de entrada e saída, como mostrado abaixo:

$$total\_de\_bytes = \max((ifInOctets_y - ifInOctets_x), (ifOutOctets_y - ifOutOctets_x))$$

$$total\_de\_bytes\_por\_segundo = \frac{total\_de\_bytes}{(y-x)}$$

$$utilização = \frac{(total\_de\_bytes\_por\_segundo * 8)}{ifSpeed}$$

No caso de se tratar de um interface serial, o limite médio aumenta para 50%, conforme foi verificado quando analisou-se a *baseline* da rede.

Quando percentuais acima desses limites forem verificados no sistema em questão, um registro de problemas é criado para que uma medida possa ser tomada. As recomendações apresentadas são as seguintes:

- verificar se determinada aplicação está sendo muito utilizada e está gerando muito tráfego.
- verificar a quantidade de memória disponível na estação sendo monitorada e incrementá-la quando necessário. Aumentar a velocidade de processamento dos servidores da rede.
- confinar o tráfego criando outro segmento.

5.2.2.3 Taxa de Erros

Quando o desempenho de uma rede começa a degradar, ou seja, a rede começa a ficar muito lenta, o que pode estar acontecendo é a taxa de erros estar em níveis muito elevados. Quando isso ocorre, deve-se investigar a causa para então tentar resolvê-la.

Freqüentemente, a análise da percentagem de erros de entrada e saída pode ajudar a isolar um problema de rede. Segundo [LEI 93], erros de entrada possivelmente indicam problemas com os dados sendo recebidos (tal como pacotes que são muito grandes ou muito pequenos). Em alguns casos erros de saída podem ser o resultado de problemas encontrados nos meios físicos, tal como perda de sincronismo em um enlace serial ou com o sistema fonte.

Segundo estudos realizados, a causa para que esses problemas ocorram pode ser devido a:

- problemas em equipamentos como *transceivers* ou *drivers* que controlam a placa de rede;
- problemas no cabeamento (como por exemplo, cabos fora da especificação);
- a interface pode estar enviando ou recebendo pacotes maiores do que ela é capaz de enviar ou receber (neste caso pode-se consultar o objeto *ifMtu* que indica o tamanho do maior datagrama que pode ser enviado ou recebido pela interface [McC 91]);
- problemas de armazenamento;
- interface com problemas.

Quando o percentual de erros que está entrando ou saindo por uma interface for maior que 2% do tráfego total da rede, deve-se verificar se esse percentual elevado continua por mais algum tempo. Caso isso seja verificado, deve-se tentar descobrir sua causa tentando identificar algum dos problemas mencionados acima. É importante observar que o percentual de erros vai depender do tipo de interface utilizada (Ethernet ou Serial).

O percentual de erros da interface Serial foi definida levando-se em conta os valores obtidos da *baseline* da rede. Portanto, o percentual utilizado nesse caso será de 5% do tráfego total. Ver Anexo A-1.

5.2.2.4 Taxa de *Broadcast*

Pacotes *broadcast* são pacotes destinados a todas estações de uma rede, tendo um endereço comum. Esses pacotes servem para verificar meios de comunicação, bem como informar para a rede da existência de outros dispositivos como roteadores e servidores de arquivos.

Toda a rede possui um nível de pacotes *broadcast* que é normal, necessário para mantê-la em operação. Contudo alguns *broadcasts* podem causar problemas, especialmente se eles são passados de outras redes, mas não são requeridos na rede onde foram capturados.

Na maioria das redes o normal desse tipo de pacote é entre 8 e 10% do tráfego total. Problemas tais como *broadcast storm* podem ocorrer quando esse percentual aumentar atingindo de 20 a 25%. Este percentual elevado pode ser devido a problemas em algum equipamento da rede. Além disso, *broadcast storm* ocorre freqüentemente quando certos tipos de roteadores ou servidores de arquivos são instalados para não receberem um tipo esperado de resposta de outro dispositivo. Eles às vezes realmente causam um nível de *broadcast* que atrapalham (retardam) a comunicação através da rede. É preciso levar em consideração que freqüentemente o número de pacotes *broadcast* pode aumentar, não porque houve um problema, mas porque um usuário está entrando ou saindo da rede.

O problema da rede apresentar taxas elevadas de *broadcast* é verificado quando ela está conectada através de *bridges*, pois todo o tráfego de *broadcast* é passado pelas *bridges* e afeta outras redes.

Para verificar o percentual de pacotes *broadcast* entrando ou saindo por uma interface monitora-se os objetos *ifInNUcastPkts* e *ifOutNUcastPkts* respectivamente, e divide-os pela taxa de tráfego que está entrando ou saindo da interface. Os objetos *ifInNUcastPkts* e *ifOutNUcastPkts* indicam o número de pacotes não *unicast* sendo enviados de uma máquina para muitas outras.

Quando o percentual de pacotes *broadcast* estiver muito elevado recomenda-se tentar identificar a máquina daquele segmento que está enviando muitos pacotes deste tipo. Em seguida, deve-se verificar sua configuração a nível de endereçamento. Um número elevado desse tipo de pacote também pode ser causado por algum problema nos protocolos ou algum defeito na aplicação.

5.2.2.5 Verificação do estado de uma interface

É possível identificar quando uma interface está *down* através da observação de dois objetos da MIB II, *ifOperStatus* (estado operacional da interface) e *ifAdminStatus* (estado administrativo da interface). Para tanto a seguinte regra é utilizada:

Se (*ifOperStatus* = *down*)

Então Se (*ifAdminStatus* = *up*)

Então Se (após três tentativas em intervalos de cinco minutos a interface continuar *down*)

Então (Cria um registro de problemas avisando para o administrador da rede que a interface de determinada máquina está *down* e a hora que isto foi diagnosticado.)

Quando uma interface está *down* pode-se dizer que tem-se uma situação bastante crítica, pois dados não poderão trafegar pela interface e uma rede inteira pode ficar isolada caso não haja outras interfaces de comunicação com o mundo externo. Se nada de anormal é verificado com uma interface, o intervalo de monitoração continua sendo de uma hora. Uma interface pode estar *down* por motivos como: falta de portadora ou o "outro lado" não enviou um *keepalive* em dez segundos (isto é uma característica de roteadores CISCO onde ambos os lados enviam uma mensagem informando que estão vivos).

Em uma WAN o estado de uma interface pode ficar oscilando entre *up* e *down*, portanto é muito importante identificar essas rajadas no tempo para que alertas não sejam enviados desnecessariamente.

5.2.2.6 Verificação de reinicialização do roteador

Para gerência de falhas pode-se verificar se um roteador está sendo reinicializado muito freqüentemente. Isso pode estar ocorrendo devido à falta de energia elétrica, mas também pode ser devido a problemas internos ao roteador. A principal forma de descobrir o que está causando a reinicialização é analisando o contador *sysUpTime* daquele roteador. Se for verificado que ele está sendo reinicializado em intervalos irregulares e freqüentes, deve-se verificar se outras máquinas, que estão próximas a ele, também foram reinicializadas. Se acontecer dessas máquinas não terem sido reinicializadas pode-se dizer que o problema não é por falta de energia e sim por algum problema interno que está ocorrendo no roteador. Deve-se então avisar ao administrador da rede informando o problema encontrado.

5.2.2.7 Percentual de datagramas IP

Uma outra causa para a perda de desempenho em uma rede é quando a taxa de datagramas IP, que está sendo recebida ou enviada pela entidade estiver muito elevada. Uma forma de verificar o tráfego IP em uma rede é por meio de monitorações em objetos do grupo IP da MIB II, como por exemplo, para encontrar o percentual de datagramas IP recebido por uma interface divide-se *ipInReceives* pelo somatório de *ifInUcastPkts* e *ifInNUcastPkts* para cada interface. Um cálculo semelhante pode ser feito usando o objeto *ipOutRequests* para obter o percentual de datagramas que saem por uma interface.

O grupo IP é composto por vários objetos que indicam quando ocorrem determinados problemas com os datagramas. Abaixo encontra-se uma análise de cada um desses objetos:

- **datagramas descartados** - através dos objetos *ipInDiscards* e *ipOutDiscards* pode-se descobrir se muitos datagramas recebidos e enviados estão sendo descartados. O descarte de datagramas pode ocorrer pela falta de recursos do sistema, isto é, falta de *buffers* para armazená-los, ou por alguma outra razão que não permita o processamento adequado dos datagramas. Neste caso, sugere-se verificar a quantidade de memória disponível na estação sendo monitorada e incrementá-la quando necessário.
- **erros no cabeçalho** - uma outra condição de erro pode ocorrer quando datagramas chegam na entidade com um cabeçalho inválido. O objeto da MIB II que fornece esse valor é o *ipInHdrErrors* e ele sempre é incrementado quando um dos seguintes erros forem detectados: *bad checksum, versus number, mismatch, other format errors, time-to-live exceded, errors discover improving their IP options,* etc. Uma forma de verificar se datagramas estão sendo descartados por "*time-to-live exceded*" é monitorando o objeto *icmpOutTimeExcds* (descrito na próxima seção).
- **endereço destino inválido** - todos os datagramas recebidos com um endereço IP inválido são descartados. Através de monitorações no objeto *ipInAddrErrors* é possível obter o número de datagramas descartados devido a erros de endereçamento. Endereços IP inválidos incluem o endereço 0.0.0.0, bem como endereços de classes não suportadas, como a classe E. Além disso, este contador também é incrementado quando máquinas não são *gateways* e recebem datagramas com um endereço que não é o seu no campo de endereço destino.
- **protocolos desconhecidos ou não suportados** - quando a entidade está tendo que processar um grande número de datagramas para os quais o protocolo de nível superior não é suportado, medido pelo objeto *ipInUnknownProtos*, o desempenho da rede pode ser atingido. Tipicamente, neste ponto a entidade já recebeu o datagrama, verificou se havia algum erro

e determinou que o destino era a própria entidade, mas agora é preciso descartá-lo porque ele é destinado a um protocolo de nível superior desconhecido.

- **datagramas sem rota** - através da monitoração do objeto *ipOutNoRoute* pode-se identificar o número de vezes que a entidade não teve uma rota válida para um datagrama. Uma causa para este problema é algum erro na tabela de roteamento da máquina ou porque todos os *gateways default* estão *down*. Quando esse problema for diagnosticado, é importante verificar se existe uma rota *default* na tabela de roteamento, uma vez que toda tabela de roteamento deve possuir tal rota.
- **falhas de remontagem** - uma grande quantidade de falhas na remontagem de datagramas pode reduzir o desempenho da rede. O objeto *ipReasmFails* é incrementado sempre que uma falha desse tipo ocorre. Falhas na remontagem de datagramas podem ser devido a: erros ou fragmentos perdidos. Caso um enlace que estava ativo cair e não houver outro meio de comunicação, problemas deste tipo irão aparecer.
- **datagramas que não puderam ser fragmentados** - sempre que determinado datagrama precisar ser fragmentado mas seu *flag Don't Fragment* estiver *on* o datagrama será descartado pela entidade. Monitorações no objeto *ipFragFails* informam a quantidade de datagramas descartados por esse motivo.

Normalmente quando ocorre algum problema com um datagrama IP uma mensagem ICMP é enviada para a máquina que o originou indicando o problema ocorrido. Portanto, a maioria dos contadores do grupo ICMP são incrementados quando ocorrerem problemas como os descritos acima.

5.2.2.8 Percentual de mensagens ICMP

Conforme visto na seção 2.4, um sinal evidente de qualquer problema num ambiente TCP/IP é um alto número de mensagens ICMP. O protocolo ICMP é usado pelos protocolos TCP/IP para monitorar a comunicação em geral. Se um número alto de mensagens ICMP está presente na rede, significa que houve algum erro no processamento do datagrama IP. ICMP usa o suporte básico do IP como se ele fosse um protocolo de nível mais alto, contudo, ele é realmente uma parte integral do IP e deve ser implementado por todos os módulos IP. A mensagem ICMP é transmitida dentro do campo de dados do datagrama, portanto, ambos IP e ICMP estão envolvidos neste processo. Para encontrar o percentual de mensagens ICMP recebido por uma interface divide-se *icmpInMsgs* pelo somatório de *ifInUcastPkts* e *ifInNUcastPkts* para cada interface. Um cálculo semelhante pode ser feito usando o objeto *icmpOutMsgs* para obter o percentual de mensagens ICMP que saem por uma interface.

Um número excessivo dessas mensagens pode degradar o desempenho de uma rede. Enquanto durante períodos de tráfego normal o poder de processamento consumido pode ser mínimo, em horas mais ocupadas, o envio de grandes números de pacotes ICMP pode requerer recursos suficientes para sensivelmente atrapalhar o desempenho da entidade, [LEI 93].

É importante notar que nem sempre a recepção ou envio de pacotes ICMP podem significar que um problema de desempenho exista, mas possuir essas estatísticas pode ajudar a resolver um problema futuro.

Todas essas mensagens podem ser monitoradas através do grupo ICMP da MIB II. Para cada tipo diferente de mensagem que o protocolo ICMP envia existem objetos da MIB II que são contadores e correspondem a essas mensagens. Abaixo estão descritas as mensagens mais importantes com seus respectivos objetos:

- **mensagem de destino não alcançável (*icmpInDestUnreachs* / *icmpOutDestUnreachs*)** - esta mensagem é enviada quando for verificado que uma rede, um *host*, um protocolo ou uma porta não podem ser localizados. Este problema pode ser diagnosticado verificando o estado do enlace da entidade monitorada. Além disso, esse tipo de mensagem também indica quando uma rota, para determinado destino, falhou (indicado por *ipOutNoRoutes*) e quando um datagrama precisa ser fragmentado mas seu *flag Don't Fragment* está *on* (indicado por *ipFragFails*). Através de monitorações nesses objetos é possível identificar qual o motivo deste alto número de mensagens e então gerar um alerta com o que foi diagnosticado. (Ver percentual de datagramas IP em 5.2.2.7).

- **mensagem de tempo excedido (*icmpInTimeExcds* / *icmpOutTimeExcds*)** - esta mensagem serve para notificar à máquina origem que o datagrama enviado foi descartado porque seu campo TTL foi zerado. O objeto *ipInHdrErrors* também será incrementado quando o datagrama for descartado por esse motivo. Em adição, ele também serve para informar que datagramas não puderam ser remontados pois fragmentos foram perdidos (*ipReasmFails*). Quando o valor do campo TTL cai a zero a culpa é muitos *hops* ou muito tempo em filas esperando para serem enviados. Além disso, tabelas de roteamento podem estar mal configuradas e por isso o datagrama não consegue chegar ao destino (pode estar ocorrendo *loop* de roteamento).

- **mensagem indicando problemas de parâmetro (*icmpInParmProbs* / *icmpOutParmProbs*)** - quando uma máquina que está processando um datagrama encontrar um problema com os parâmetros do cabeçalho tal que não possa completar seu processamento, esta máquina deve descartar o datagrama. Pode-se comprovar que um problema deste tipo tenha ocorrido, monitorando-se o objeto *ipInHdrErrors*. A origem de tal problema pode ser devido a argumentos incorretos em uma opção. Uma alta taxa deste tipo de mensagem geralmente está relacionada com algum problema em uma

aplicação, isto é, alguma aplicação está preenchendo o cabeçalho do datagrama IP inadequadamente.

- **mensagem *source quench* (*icmpInSrcQuenchs / icmpOutSrcQuenchs*)** - esta mensagem é enviada quando o *gateway* não tem espaço em *buffer* necessário para armazenar um datagrama antes de enviá-lo e então descarta-o. Neste caso há um relacionamento com o objeto *ipInDiscards*. Um *host* também pode enviar essa mensagem se datagramas chegam mais rápido do que o *host* pode processá-los. A mensagem *source quench* é um pedido para o *host* reduzir a taxa na qual ele envia dados ao destino. A taxa deve ser reduzida até que essas mensagens não sejam mais recebidas. Quando esse tipo de alerta é gerado para todas as máquinas significa que pode estar ocorrendo congestionamento na rede. Caso o alerta seja de somente uma máquina significa que a máquina está transmitindo ou recebendo muito tráfego.
- **mensagem *redirect* (*icmpInRedirects / icmpOutRedirects*)** - esta mensagem é enviada por um *gateway* para avisar ao *host* que ele deve enviar seu tráfego por um outro caminho que é mais curto, isto é, serve para trocar uma rota particular. Caso o percentual deste tipo de mensagem seja muito elevado, recomenda-se verificar as tabelas de roteamento das máquinas, pois rotas mais curtas podem ser utilizadas.

5.2.2.9 Percentual de segmentos TCP

O TCP é um protocolo de transporte que fornece conexões seguras entre aplicações. Ao monitorar certos objetos do grupo TCP pode-se descobrir por que alguns problemas de desempenho estão ocorrendo na rede, por exemplo, pode-se identificar o percentual de segmentos TCP de entrada e saída dividindo-se *tcpInSegs* e *tcpOutSegs,* respectivamente, pelo somatório de *ifInUcastPkts* e *ifInNUcastPkts* para cada interface. Quando a taxa encontrada estiver muito elevada deve-se analisar os objetos do grupo TCP separadamente para identificar a causa. A seguir estão descritos alguns desses indicadores:

- **falha na tentativa de estabelecer conexão** - uma tentativa de estabelecer conexão pode falhar por vários motivos; por exemplo, o sistema destino não existe ou a rede pode estar com uma falha. Saber o número de vezes que essas falhas ocorrem pode ajudar a quantificar a confiabilidade da rede, onde um número menor de rejeições pode indicar uma rede mais segura. O objeto *tcpAttemptFails* informa esse valor. Falhas desse tipo podem ser ocasionadas por: erros no cabeçalho IP (*ipInHdrErrors*), endereço destino inválido (*ipInAddrErrors*), não há uma rota válida (*inOutNoRoutes*), datagramas descartados por falta de *buffer* (*ipInDiscards*), enlace *down* na máquina destino, entre outras.

- **segmentos retransmitidos** - o objeto *tcpRetransSegs* fornece o número de segmentos TCP que tiveram que ser reenviados. A retransmissão de um segmento TCP não reflete diretamente um problema de desempenho; contudo, o número de retransmissões pode ajudar a verificar se a entidade está tendo que enviar múltiplas cópias de dados em um esforço de assegurar confiabilidade. Muitos segmentos podem estar sendo retransmitidos pois muitos segmentos com erros estão sendo recebidos (*tcpInErrs*), tentativas de estabelecer conexão estão falhando (*tcpAttemptFails*), o destino pode não estar ativo (enlace *down*), pode estar ocorrendo falta de *buffers* (*ipInDiscards*), endereço destino inválido (*ipInAddrErrors*), falha na remontagem de datagramas (*ipReasmFails*), falha na fragmentação de datagramas (*ipFragFails*), datagramas sem rota (*ipOutNoRoutes*), entre outros. Monitora-se cada um desses objetos para se tentar descobrir qual a causa dessa alta taxa de retransmissão e em seguida cria-se um alerta com o resultado encontrado.

Através de monitorações nos objetos *tcpCurrEstab* (número de conexões estabelecidas) e *tcpMaxConn* (número máximo de conexões que podem estar abertas) pode-se verificar se seus valores são aproximadamente iguais. Caso isso seja comprovado e a carga da CPU não esteja muito elevada, pode-se tentar aumentar o número máximo de conexões que podem estar abertas. Feito isso, deve-se verificar a carga de CPU novamente e verificar se não estão ocorrendo muitos *timeouts*.

6 Especificação do Protótipo

A especificação do protótipo através de uma linguagem formal foi feita utilizando-se a Linguagem de Descrição e Especificação SDL (*Specification and Description Language*) desenvolvida pelo CCITT. Segundo [TRI 92], o propósito de sua recomendação é prover uma linguagem para especificação e descrição não ambígua no comportamento de sistemas de telecomunicações. SDL é utilizada para descrever o comportamento real de um sistema.

De acordo com a figura 6.1, o MAD possui dois grandes blocos: o bloco "Envia_Requisição" e o bloco "Trata_Evento". O bloco "Envia_Requisição" é responsável por ler o arquivo de configuração criado pelo usuário, enviar requisições ao Sistema de Alertas e receber eventos vindos do Sistema de Alertas. O bloco "Trata_Evento" é responsável por transformar eventos em alertas. Esses dois blocos se comunicam quando o evento recebido do Sistema de Alertas é enviado para o bloco "Trata_Evento" para ser tratado (a representação, na figura 6.1, está sendo feita pelo sinal "Evento" do canal C4). O sinal "Evento" possui os valores coletados da rede que ultrapassaram seu limite, bem como a quantidade de tráfego que está passando pela rede naquele momento. Ambas variáveis são do tipo "*unsigned long*". Além disso, também é passado pelo canal C4 a identificação da monitoração em questão, a qual é do tipo "inteiro".

O bloco "Envia_Requisição" recebe requisições do usuário, indicadas pelos sinais do canal C1. As requisições do usuário são enviadas ao Sistema de Alertas pelo canal C2, as quais podem ser: iniciar alguma monitoração ou cancelar a monitoração corrente. O sinal "Requisições" desse canal é uma "matriz de *strings*". A posição inicial dessa matriz, posição zero, possui um código diferente dependendo do tipo de operação a ser feita. Esses códigos estão listados abaixo:

- 01 - INICIA (Quando este código é usado indica que uma monitoração deve ser iniciada. Neste caso, todas as informações que foram lidas do arquivo de configuração, são armazenadas nas demais posições dessa matriz. Portanto, cada linha da matriz possui as seguintes informações: número da monitoração, entidade monitorada, objeto, identificador de expressão, intervalo, número de *pollings*, tipo de amostragem, janela de amostragem, limite inferior, limite superior, fator inferior e fator superior.);
- 02 - REMOVE_TODOS (Pedido de encerramento de todas as monitorações correntes);
- 03 - DESCONECTA (Indica que o MAD está sendo finalizado).

Quando o Sistema de Alertas está monitorando a rede e gera um evento, este evento é enviado ao MAD pelo sinal "Eventos" do canal C3. O sinal "Eventos" é

definido como uma variável do tipo "*string*" e pode conter o valor que ultrapassou o limite. Dependendo do tipo de código contido na posição zero desse *string*, informações diferentes serão encontradas em suas posições subseqüentes. Os códigos que podem ser encontrados estão listados abaixo:

- 10 - RELATA (Com esse código, a posição 1 do sinal conterá o número da monitoração e o restante conterá o valor que ultrapassou o limite.);
- 11 - ENCERRADO (Indica que todas as monitorações foram encerradas.);
- 12 - OK_INICIA (Confirmação do pedido de inicialização de uma monitoração. Neste caso, a posição 1 do sinal conterá o número da monitoração que foi inicializada.);
- 13 - PKTSNU (Indica que o valor que está sendo enviado faz parte do tráfego não *unicast* que está passando pela rede. Com esse código, a posição 1 do sinal conterá o número da monitoração e o restante conterá o valor do tráfego não *unicast*.);
- 14 - PKTSU (Indica que o valor que está sendo enviado faz parte do tráfego *unicast* que está passando pela rede. Com esse código, a posição 1 do sinal conterá o número da monitoração e o restante conterá o valor do tráfego *unicast*.);
- 15 - ADMIN (Indica que o estado operacional de determinada interface está *down*. Neste caso, a posição 1 do sinal conterá o número da monitoração.);
- 16 - TUTIL (Indica que o valor recebido é referente ao número de octetos utilizados no cálculo da taxa de utilização. Com esse código, a posição 1 do sinal conterá o número da monitoração e o restante conterá o valor que ultrapassou o limite.);
- 21 - ERRO (Indica que houve algum problema na monitoração. Neste caso, a posição 1 do sinal conterá o número da monitoração que ocorreu o problema.).

O protótipo analisa cada código recebido para então tomar a ação apropriada.

O bloco "Trata_Evento", ao identificar que um alerta deve ser criado, consulta a base de registros de problemas para verificar se o mesmo alerta já foi gerado anteriormente e recebe uma resposta através do canal C5. O sinal deste canal é do tipo "*string*" e possui a solução que foi utilizada para resolver o problema diagnosticado. Os alertas para os quais um registro de problemas deve ser criado são enviados pelo canal C6 (sinal "Cria_ticket"). O sinal "Cria_ticket" possui a seguinte estrutura:

```
NEWTYPE ticket
STRUCT
        escal_level,
        tkt_priority,
        extension_resp : integer;
        open_time,
        open_date,
        opened_by,
        contact,
        extension,
        e_mail,
        wstation,
        domain,
        responsible,
        e_mail_resp,
        notifications,
        brief_desc,
        remarks         : string;
ENDNEWTYPE

DCL
Cria_ticket : ticket;
```

O canal C7 é utilizado para enviar os dados que fazem parte do alerta criado para que o *log* seja gerado.

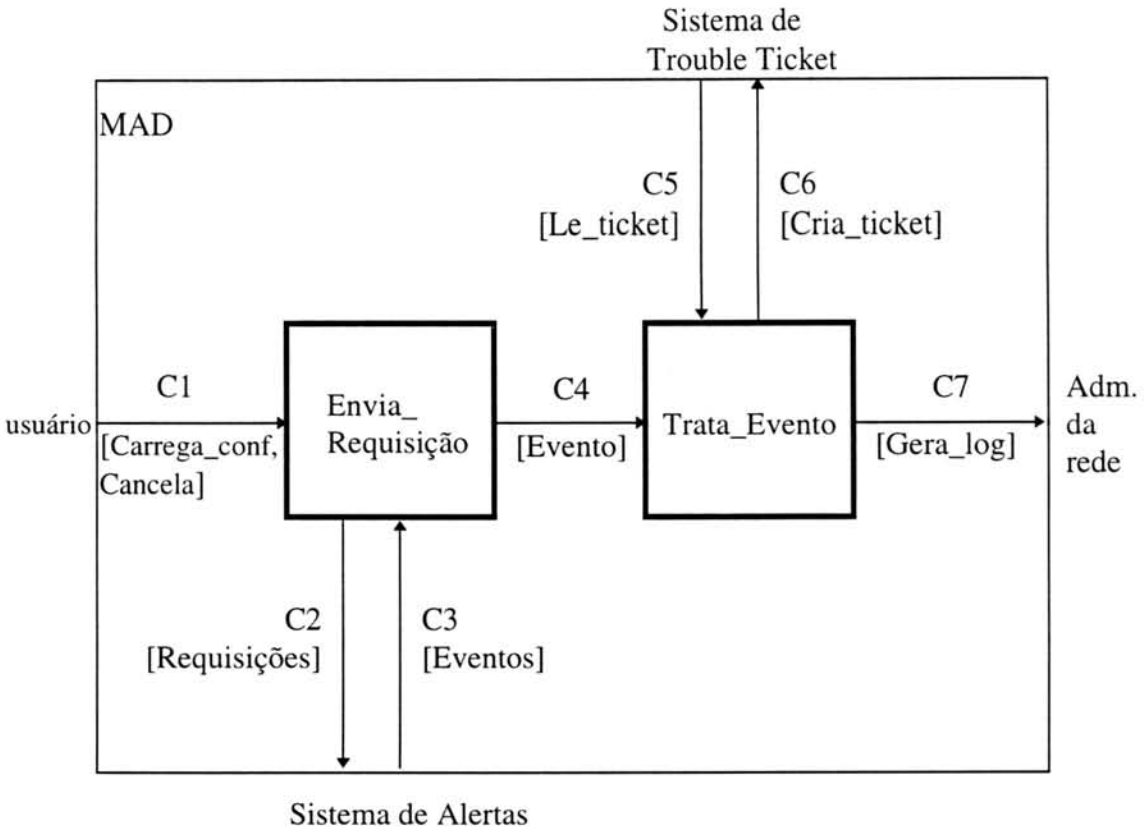

FIGURA 6.1 - Diagrama de blocos do MAD.

Portanto, o protótipo funciona da seguinte forma: primeiramente, o arquivo de configuração é carregado e então suas informações são enviadas ao Sistema de Alertas. O Sistema de Alertas monitora os objetos especificados e, assim que ele verifica que algum limite foi ultrapassado, ele envia um evento para o bloco "Envia_Requisição". Este bloco aguarda a chegada de todos os parâmetros referentes a um evento e envia-os ao bloco "Trata_Evento", o qual analisa o evento recebido. Ao verificar que um alerta deve ser gerado, uma consulta na base de registro de problemas é feita para verificar se aquele problema já ocorreu anteriormente e para obter a solução aplicada. Feito isso, a estrutura contendo os dados do alerta gerado é enviada ao Sistema de *Trouble Ticket* e o evento é armazenado em um *log*.

A figura 6.2 apresenta um fluxograma do funcionamento do MAD de uma forma geral. Inicialmente são feitas algumas inicializações, como por exemplo a inicialização do banco de dados Postgres feito pelo procedimento "inicializa_dbserv()". O procedimento "ativa_handler()" é responsável por indicar a função que deve receber os eventos vindos do Sistema de Alertas. O sistema fica aguardando a chegada de eventos e assim que eles chegam o procedimento "handler()" é ativado, a figura 6.3 apresenta a descrição deste procedimento. O programa só será terminado quando o usuário determinar, por meio de um *kill*, matando o processo. Ao finalizar, um sinal é

enviado ao Sistema de Alertas para que as monitorações requisitadas por aquele cliente sejam encerradas.

O procedimento "handler()", figura 6.3, funciona da seguinte forma: inicialmente ele fica aguardando a chegada de algum dado. Caso o dado recebido seja válido, ele verifica seu código (este código foi descrito anteriormente). Dependendo do tipo de código recebido uma ação diferente é tomada. Sempre que um código do tipo "RELATA" for recebido, é necessário aguardar mais algum tempo pois os valores de tráfego serão recebidos em seguida. O código que representa os valores do tráfego são: PKTSNU e PKTSU. Quando todos os valores necessários forem recebidos, isto é, o valor que ultrapassou o limite e os valores do tráfego para aquele momento, o procedimento "trata_evento()" será chamado.

Todo o evento que chega deve ser analisado para que as ações necessárias sejam tomadas. O procedimento "trata_evento()" é responsável por analisar o tipo que evento recebido e enviá-lo ao procedimento correspondente. A descrição deste procedimento está apresentada na figura 6.4. Para que o sistema não fique esperando o evento ser analisado e possa ficar aguardando novos eventos, um filho para esse processo é criado a cada novo evento recebido. Após a criação do processo filho, o tipo de evento recebido é verificado. Os eventos podem ser dos seguintes tipos:

- estado: indica que o estado operacional de uma interface está *down*;
- taxutil: indica que os valores dos octetos, que foram enviados ou recebidos por uma interface, ultrapassaram seus limites;
- taxerros: indica que a quantidade de erros, que foi recebida ou enviada pela interface, ultrapassou seu limite;
- taxdiscards: indica que a quantidade de pacotes descartados, que foi recebida ou enviada pela interface, ultrapassou seu limite;
- taxbrd: indica que a quantidade de pacotes *broadcast*, que foi recebida ou enviada pela interface, ultrapassou seu limite;
- taxicmp: indica que a quantidade de pacotes ICMP (qualquer objeto do grupo ICMP), que foi recebida ou enviada, ultrapassou seu limite;
- taxip: indica que a quantidade de pacotes IP (qualquer objeto do grupo IP), que foi recebida ou enviada, ultrapassou seu limite;
- taxtcp: indica que a quantidade de pacotes TCP (qualquer objeto do grupo TCP), que foi recebida ou enviada, ultrapassou seu limite.

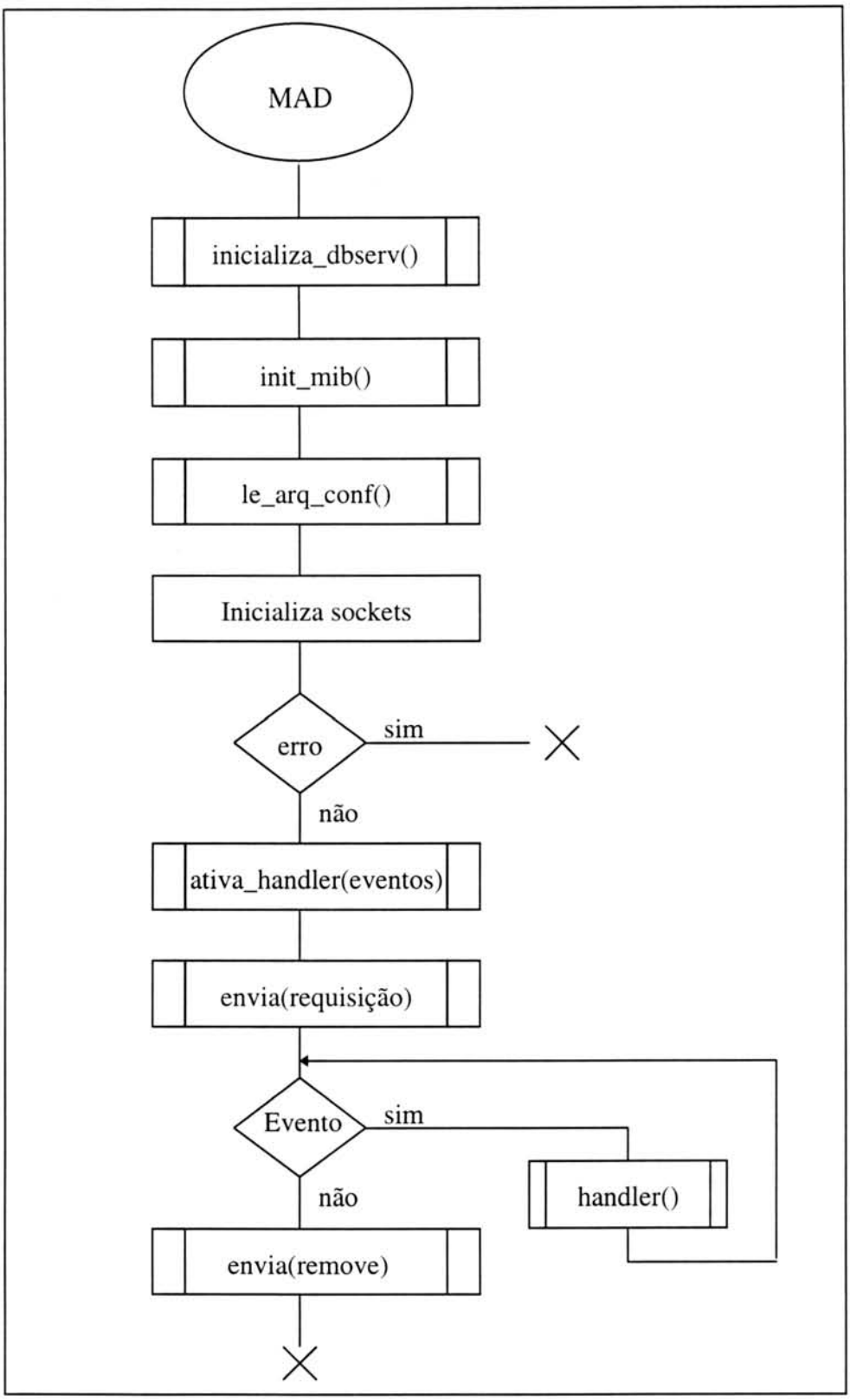

FIGURA 6.2 - Descrição geral do MAD.

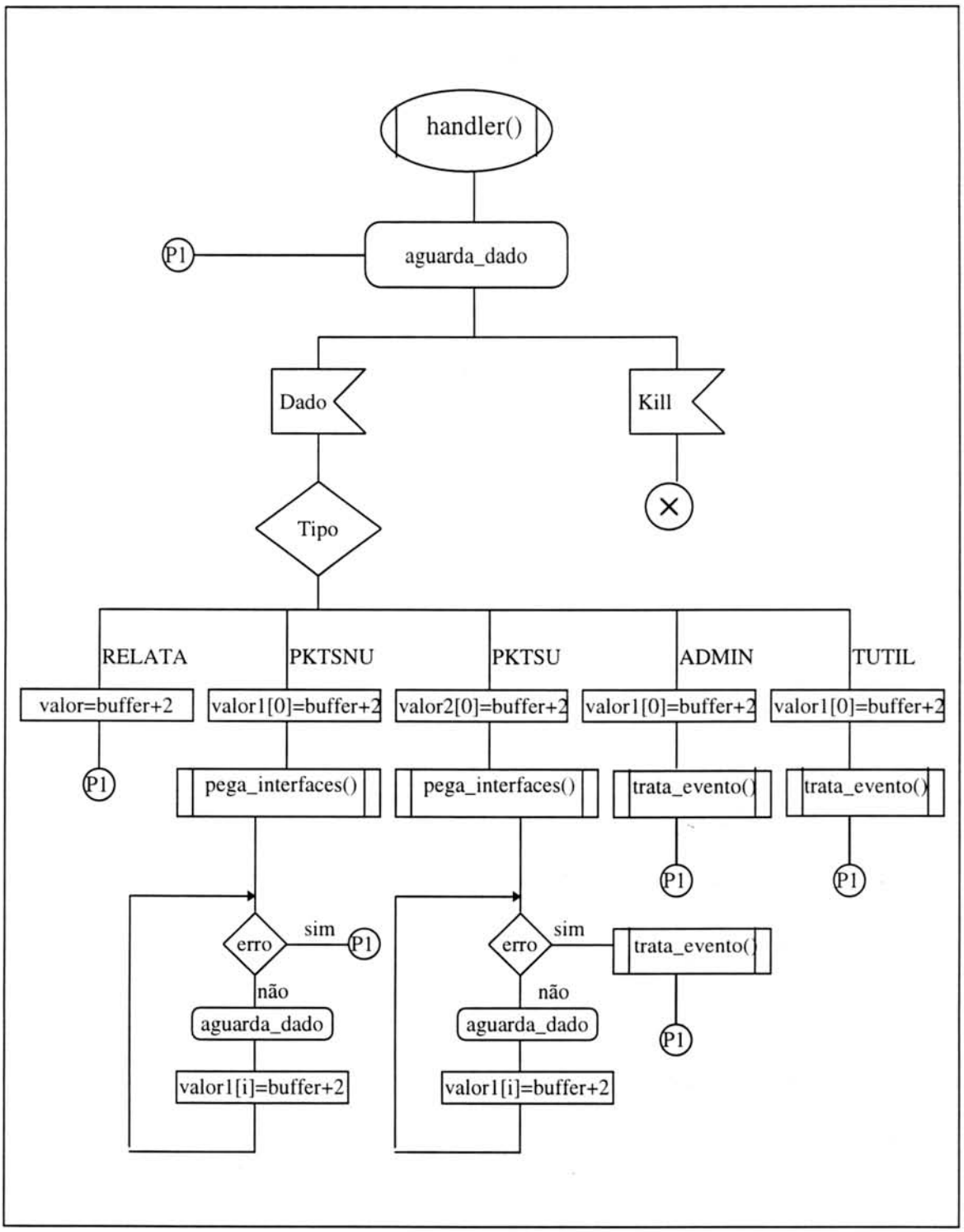

FIGURA 6.3 - Descrição do procedimento *handler()*.

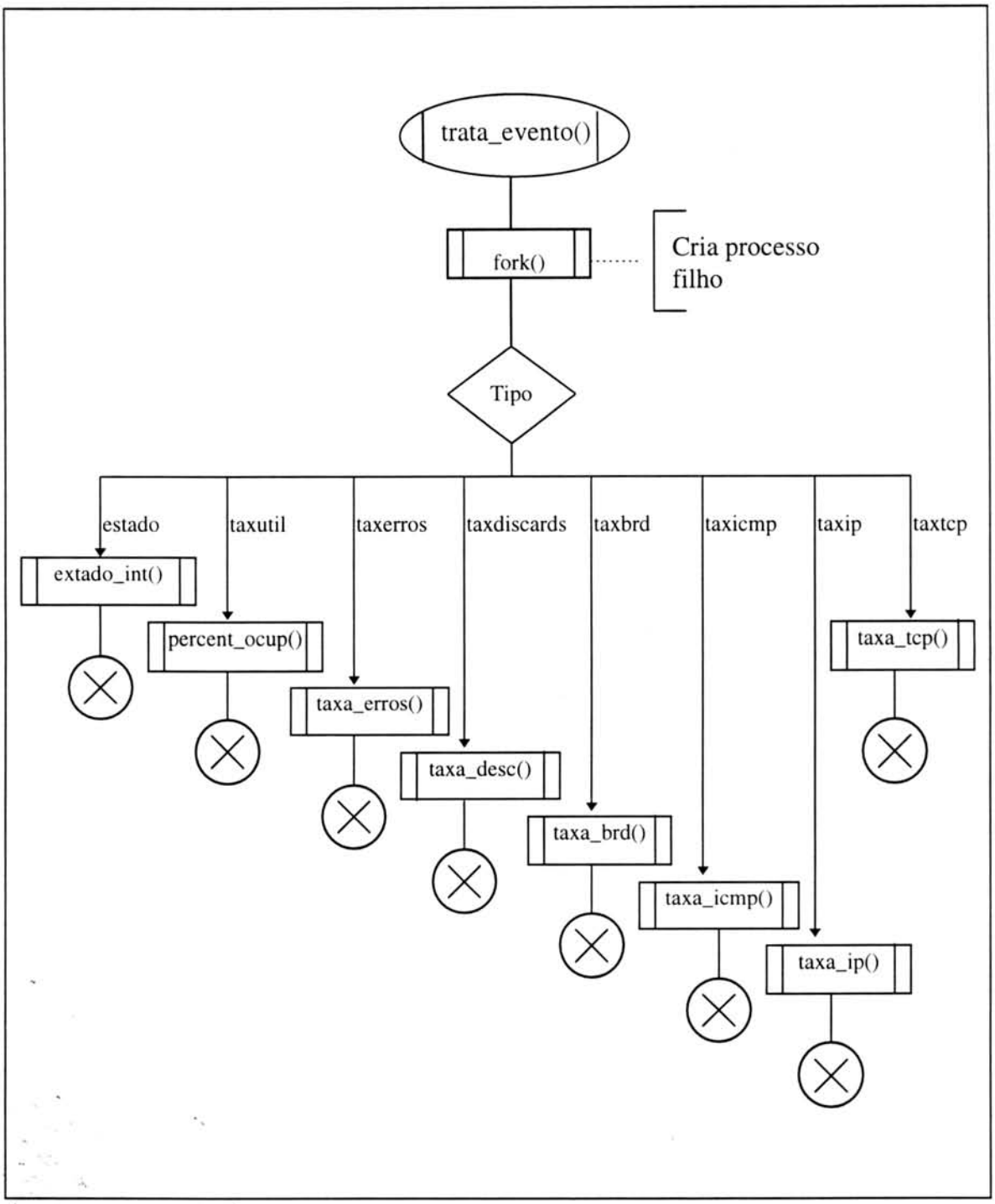

FIGURA 6.4 - Descrição do procedimento *trata_evento()*.

Com o objetivo de dar uma idéia melhor de como cada regra funciona, o procedimento "taxa_erros()" foi especificado abaixo, figura 6.5. Através desse

procedimento é possível verificar como é feita a comparação com os percentuais apropriados e a abertura de registros de problemas.

Antes de um registro de problema ser aberto é necessário verificar se o mesmo já ocorreu anteriormente e qual a solução que foi aplicada. A figura 6.6 descreve o procedimento que possui tal função. Este procedimento é chamado pelo procedimento "Open_TT()".

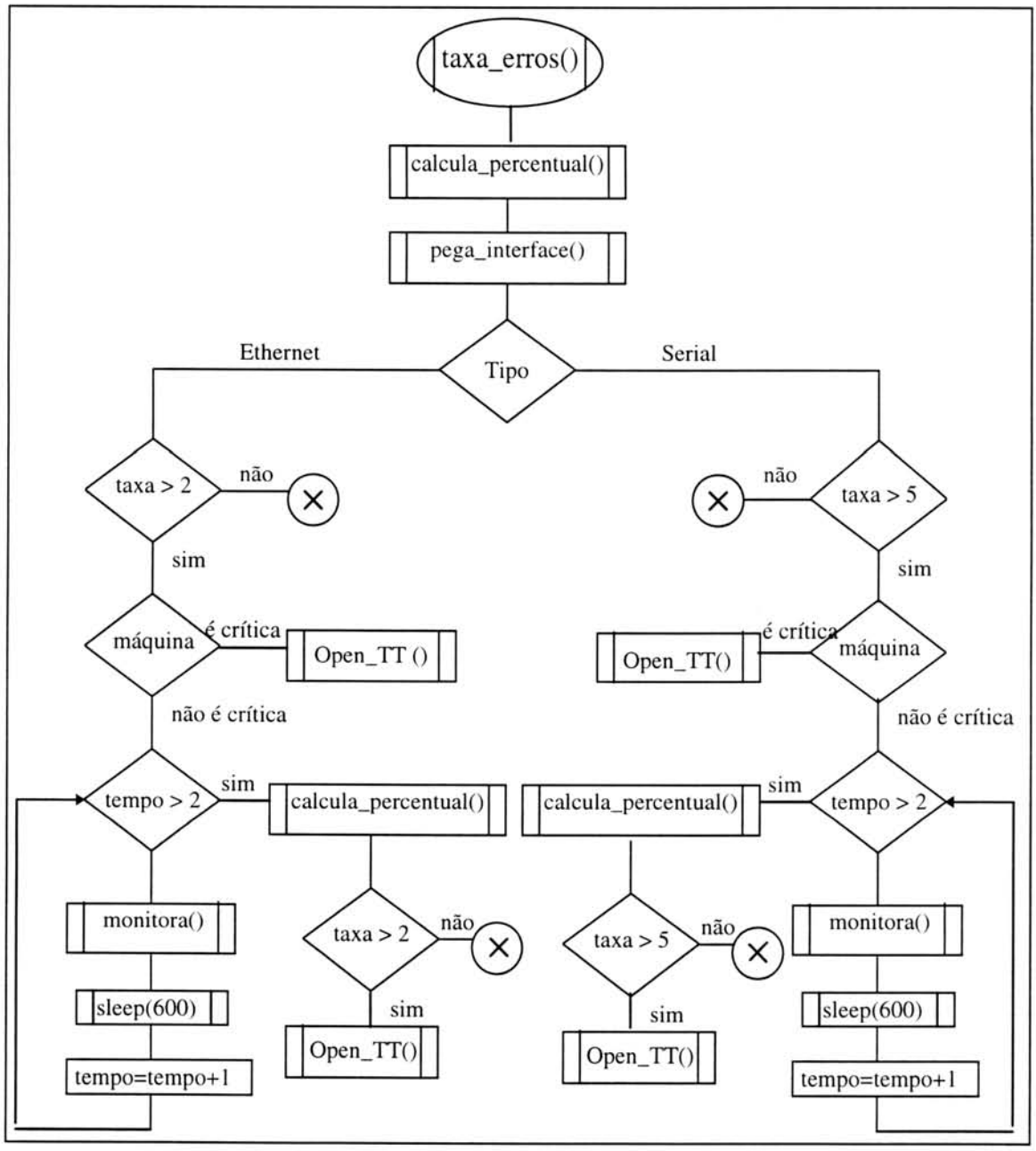

FIGURA 6.5 - Descrição do procedimento *taxa_erros()*.

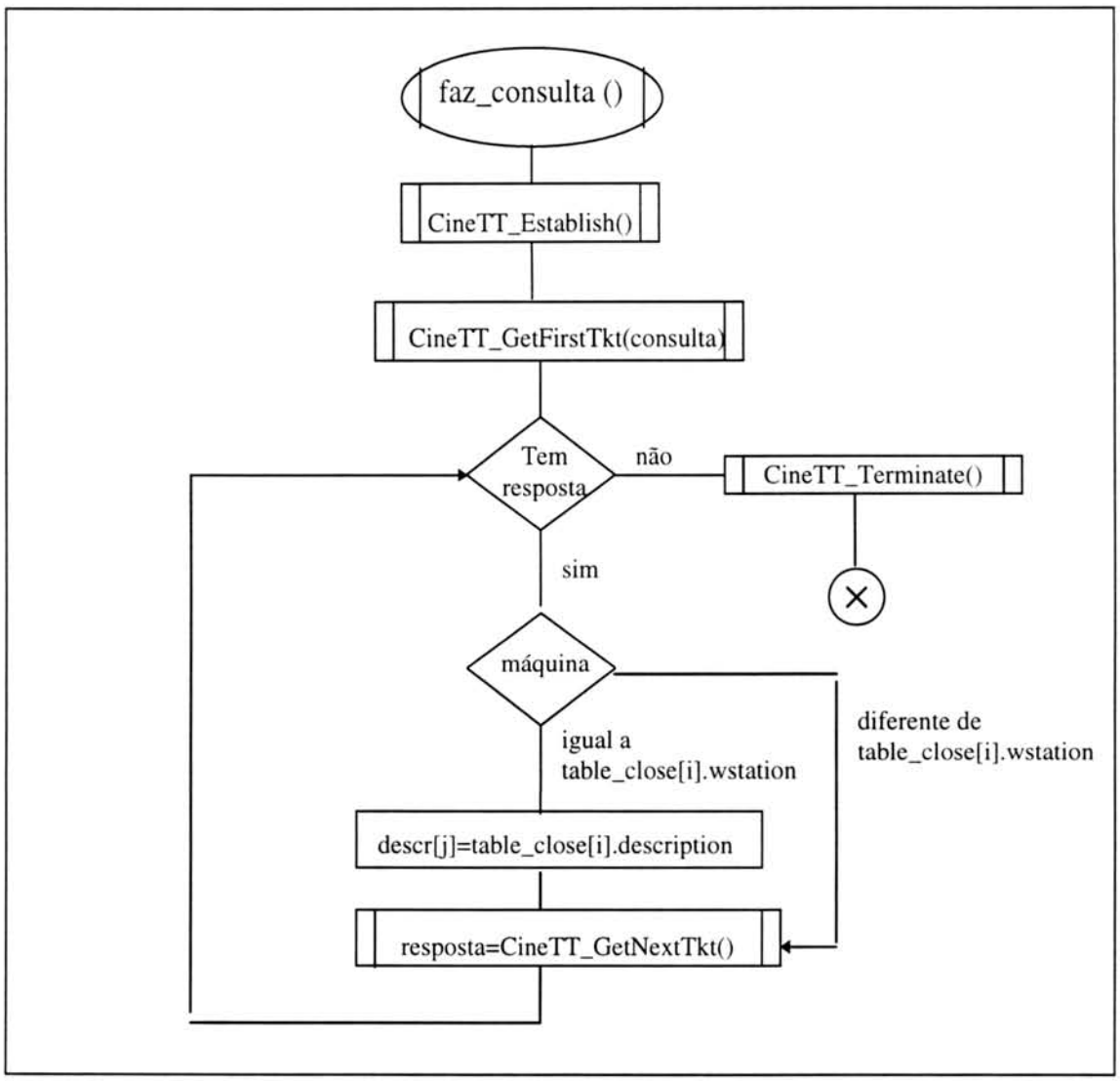

FIGURA 6.6 - Descrição do procedimento *faz_consulta()*.

6.1 Implementação do Protótipo

Para validar o paradigma proposto foi implementado um protótipo contendo as características relevantes para o seu funcionamento. Este protótipo foi desenvolvido em estações de trabalho, utilizando uma plataforma UNIX. A linguagem "C" foi utilizada em sua implementação. Além disso, a API apresentada na seção 4.3.1.3 foi utilizada na comunicação entre o protótipo e o Sistema de *Trouble Ticket*. Para manter a compatibilidade com o último sistema, que utiliza o sistema de banco de dados Postgres [STO 90] para armazenar os registros de problemas, este mesmo sistema de banco de dados foi utilizado para armazenar os dados coletados da rede que formam a *baseline*.

Não houve a necessidade de se desenvolver uma interface gráfica para o protótipo, uma vez que seus resultados são mostrados através da criação de registros de problemas e na forma de um *mail* enviado para a pessoa responsável.

Antes de inicializar o protótipo é preciso preencher um arquivo de configuração informando os dados necessários para a correta monitoração da rede. Cada linha deste arquivo deve ser preenchida com as seguintes informações:

- Número da Monitoração: cada monitoração deve ser numerada em ordem, começando pelo número "1". Este número serve como identificação da monitoração.
- Entidade Monitorada: componente da rede (roteador, *bridge*, estação de trabalho, etc) que constitui-se no alvo da monitoração.
- Objeto: nome completo do objeto que será monitorado (exemplo: *ip.ipInReceives.0*).
- Identificador de Expressão: o valor "0" indica que o objeto a ser monitorado é um objeto simples (exemplo: *ipInReceives*). O valor "1" indica que uma expressão deve ser monitorada (exemplo: taxa de pacotes descartados).
- Intervalo: espaço de tempo entre amostragens, em segundos;
- Número de *Pollings*: quantidade de vezes que um objeto deve ser monitorado.
- Tipo de Amostragem: uma das três opções - (0) as instâncias dos objetos são analisadas na forma em que foram amostradas (amostragem absoluta), ou (1) são consideradas as variações entre os valores a cada novo intervalo de amostragem (amostragem tipo delta), ou (2) são consideradas as variações entre os valores a cada segundo (amostragem tipo delta por segundo). Apenas a amostragem tipo "delta" foi implementada no protótipo do sistema.
- Janela de Amostragem: espaço de tempo que determina quais são os valores amostrados como instâncias de objetos que serão utilizados para o cálculo automático dos limites.
- Limite Inferior e Superior: valores dos limites inferior e superior respectivamente.
- Fator Inferior e Superior: valores dos fatores inferior e superior respectivamente.

Os valores informados para a janela de amostragem, limite inferior e superior e fator inferior e superior só serão utilizados caso o objeto monitorado não se encontrar na *baseline* da rede.

As figuras abaixo mostram um exemplo de um registro de problemas (figura 6.7) e de um *mail* recebido pelo CINEMA informando sobre o registro criado (figura 6.8).

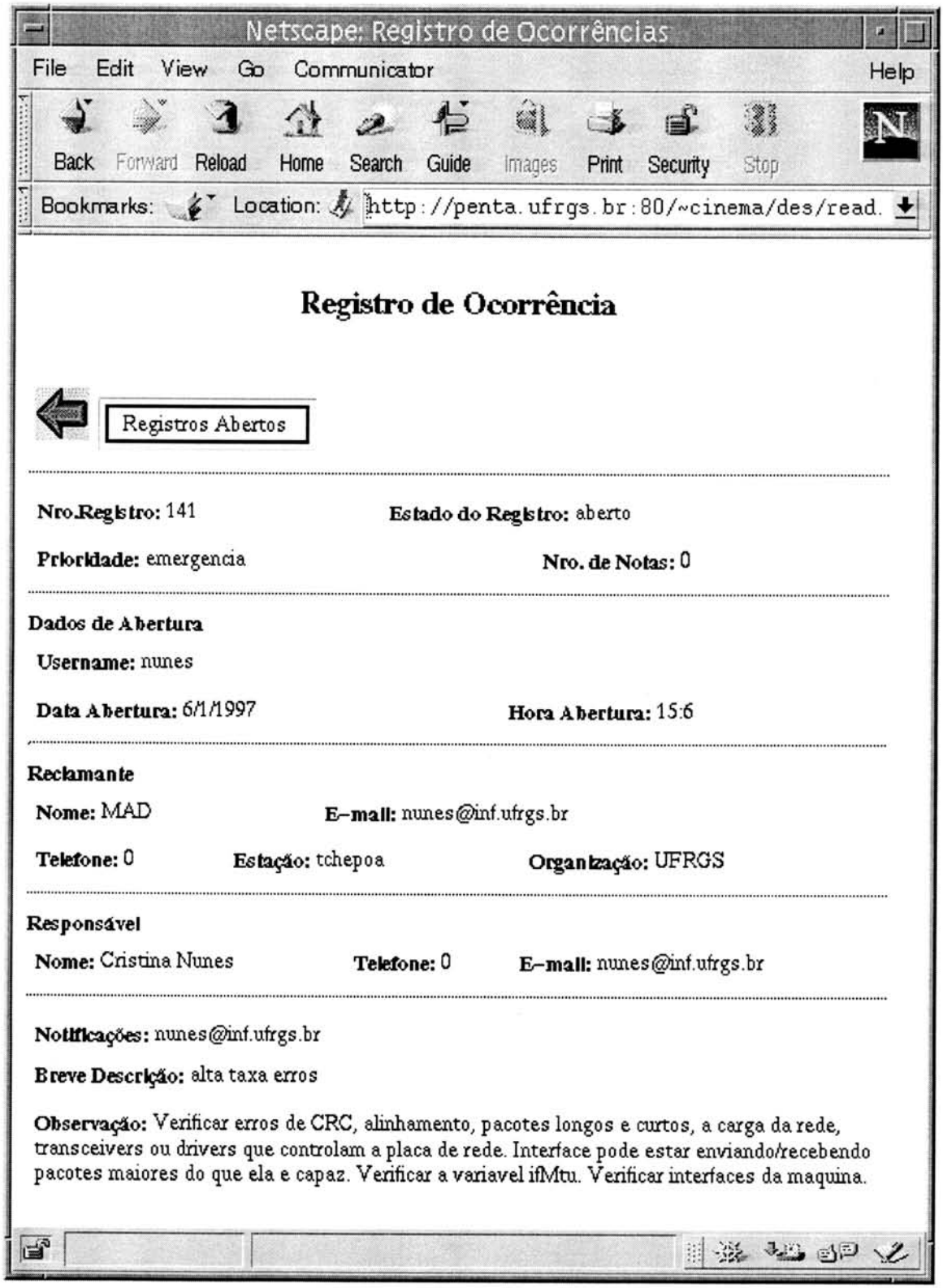

FIGURA 6.7 - Exemplo de um registro de problemas criado pelo MAD.

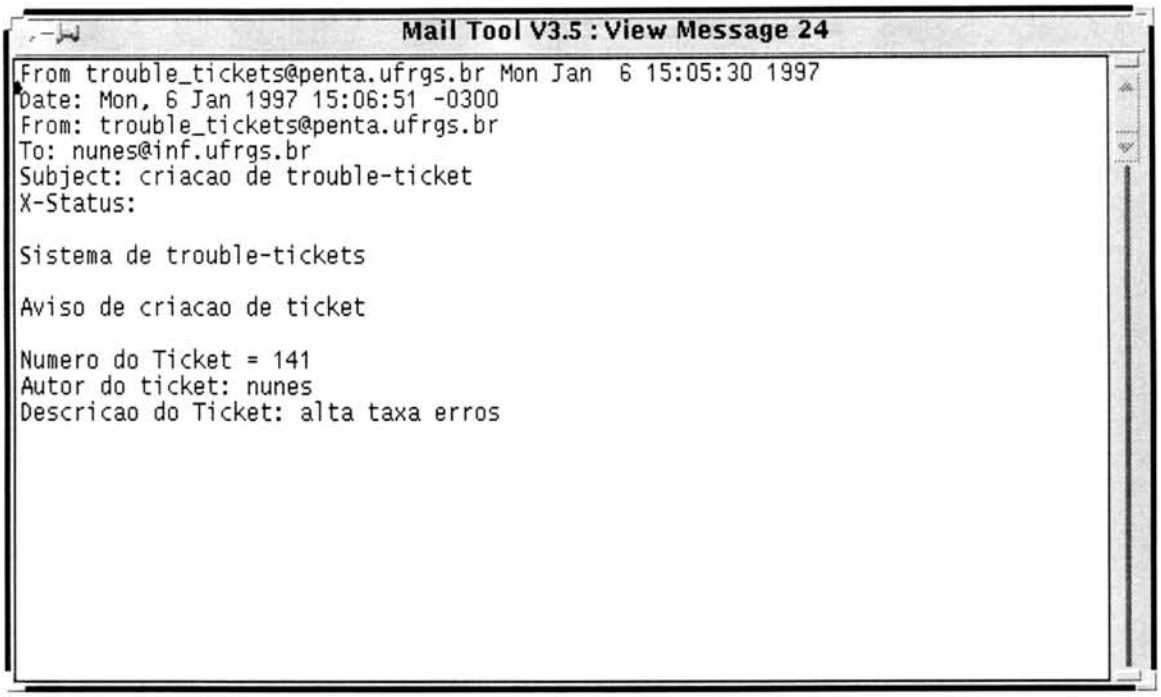

FIGURA 6.8 - Exemplo de um *mail* enviado pelo CINEMA.

6.2 Avaliação do Protótipo

Para avaliar o protótipo desenvolvido foi necessário deixá-lo rodando durante um período de tempo. O protótipo ficou executando na máquina "penta", localizada no CPD da universidade, e ocupa menos de um por cento de CPU da máquina. Portanto, não houve sobrecarga em função de sua execução. A descrição de sua utilização encontra-se no Anexo A-2.

No período de testes, vários eventos foram gerados, mas a geração de alertas não ocorreu com tanta freqüência. Além disso, pode-se observar que raramente os percentuais foram atingidos, principalmente aqueles correspondentes a mensagens ICMP e segmentos TCP. Aparentemente o comportamento da rede se manteve estável.

Neste período uma série de eventos foram gerados, mas como eles não levam em consideração o tráfego da rede, na maioria dos casos não houve necessidade do sistema gerar um alerta. Para a geração de alertas, o tráfego total que está passando pelas interfaces da máquina é levado em consideração, então a maioria dos eventos recebidos não foi transformada em alertas por seu percentual não ser significativo.

Neste período de avaliação foram monitoradas as seguintes máquinas: "routcv", "routcc", "penta", "caracol", "bb2.pop-rs.rnp.br" e "tchepoa". As monitorações foram realizadas de uma em uma hora. Abaixo estão listados alguns eventos que foram recebidos pelo sistema mas que não foram transformados em alertas por não ultrapassarem os percentuais designados pelas regras:

Objeto monitorado: *ifInNUcastPkts*.1 (taxbrd) - Interface Ethernet
Máquina: routcv
Dia: 12/11/96
Hora: 12:00
Valor que ultrapassou o limite: 3361
Média: 2805
Desvio padrão: 159
Tráfego: 199803 pacotes
Percentual: 1.68%

Objeto monitorado: *ipOutRequests* (taxip)
Máquina: routcc
Dia: 12/11/96
Hora: 12:00
Valor que ultrapassou o limite: 4578
Média: 986
Desvio padrão: 633
Tráfego: 352683 pacotes
Percentual: 1.30 %

Objeto monitorado: *ifInErrors*.1 (taxerros) - Interface Ethernet
Máquina: routcc
Dia: 13/11/96
Hora: 22:00
Valor que ultrapassou o limite: 2
Média: 0
Desvio padrão: 0
Tráfego: 395992 pacotes
Percentual: 0.0005 %

Objeto monitorado: *icmpInMsgs* (taxicmp)
Máquina: routcv
Dia: 13/11/96
Hora: 22:00
Valor que ultrapassou o limite: 120
Média: 63
Desvio padrão: 21
Tráfego: 235079 pacotes
Percentual: 0.051%

Objeto monitorado: *icmpInMsgs* (taxicmp)
Máquina: routcc
Dia: 19/11/96
Hora: 22:00
Valor que ultrapassou o limite: 103
Média: 61
Desvio padrão: 10
Tráfego: 376083 pacotes
Percentual: 0.0271%

Objeto monitorado: *ipOutRequests* (taxip)
Máquina: routcc
Dia: 20/11/96
Hora: 12:00
Valor que ultrapassou o limite: 4079
Média: 986
Desvio padrão: 633
Tráfego: 610168 pacotes
Percentual: 0.0067%

Objeto monitorado: *ifInOctets*.1 e *ifOutOctets*.1 (taxutil) - Interface Ethernet
Máquina: routcv
Dia: 20/11/96
Hora: 17:00
Valor que ultrapassou o limite: 283833000
Média: 99828750
Desvio padrão: 60793742
Tráfego: 1218166 pacotes
Percentual: 12.18%

Objeto monitorado: *icmpInMsgs* (taxicmp)
Máquina: caracol
Dia: 22/11/96
Hora: 16:00
Valor que ultrapassou o limite: 5622
Média: 1705
Desvio padrão: 1479
Tráfego: 171074 pacotes
Percentual: 3.29%

Objeto monitorado: *ifInNUcastPkts*.1 (taxbrd) - Interface Ethernet
Máquina: tchepoa
Dia: 04/12/96
Hora: 20:00
Valor que ultrapassou o limite: 1698
Média: 1461
Desvio padrão: 5
Tráfego: 220456 pacotes
Percentual: 0.77%

Objeto monitorado: *ifInOctets*.1 e *ifOutOctets*.1 (taxutil) - Interface Ethernet
Máquina: tchepoa
Dia: 12/12/96
Hora: 5:00
Valor que ultrapassou o limite: 44114689
Média: 8992948
Desvio padrão: 3248327
Tráfego: 25600 pacotes
Percentual: 0.26%

Objeto monitorado: *ifInOctets*.4 e *ifOutOctets*.4 (taxutil) - Interface Serial
Máquina: tchepoa
Dia: 13/12/96
Hora: 10:00
Valor que ultrapassou o limite: 10397824
Média: 6676124
Desvio padrão: 1065089
Tráfego: 23106 pacotes
Percentual: 36.10%

Objeto monitorado: *ifInErrors*.2 (taxerros) - Interface Ethernet
Máquina: bb2.pop-rs.rnp.br
Dia: 13/12/96
Hora: 10:00
Valor que ultrapassou o limite: 1108
Média: 792
Desvio padrão: 118
Tráfego: 355956 pacotes
Percentual: 0.31%

Objeto monitorado: *ifInOctets*.4 e *ifOutOctets*.4 (taxutil) - Interface Serial
Máquina: bb2.pop-rs.rnp.br
Dia: 13/12/96
Hora: 15:00
Valor que ultrapassou o limite: 368933917
Média: 213312014
Desvio padrão: 2134567
Tráfego: 819853 pacotes
Percentual: 40.03%

Objeto monitorado: *ifInOctets*.2 e *ifOutOctets*.2 (taxutil) - Interface Ethernet
Máquina: bb2.pop-rs.rnp.br
Dia: 17/12/96
Hora: 23:00
Valor que ultrapassou o limite: 87702507
Média: 12062606
Desvio padrão: 1206268
Tráfego: 1169366 pacotes
Percentual: 11.69%

A "média" e o "desvio padrão" indicados acima foram retirados da *baseline* da rede e o "valor que ultrapassou o limite" foi o valor coletado. Como pode-se observar, de todos os eventos listados nenhum ultrapassou o percentual indicado em sua regra. Caso esse sistema não fosse utilizado, esses e muitos outros eventos que não foram listados seriam apresentados ao administrador da rede desnecessariamente, pois eles não seriam suficientes para indicar que havia um problema na rede.

Contudo, ao analisar alguns eventos, o protótipo verificou que havia necessidade de se gerar um alerta para aquele evento. A lista abaixo apresenta alguns alertas que foram gerados pelo MAD:

Objeto monitorado: *ifInErrors*.5 (taxerros) - Interface Serial
Máquina: tchepoa
Dia: 11/12/96
Hora: 21:00
Valor que ultrapassou o limite: 268
Média: 50
Desvio padrão: 20
Tráfego: 5145 pacotes
Percentual: 5.21%
Descrição da interface: Interface com a URCAMP (linha urbana não especializada com modem banda base de 14400bps).

Objeto monitorado: *ifInOctets*.2 e *ifOutOctets*.2 (taxutil) - Interface Serial
Máquina: tchepoa
Dia: 13/11/96
Hora: 16:00
Valor que ultrapassou o limite: 7731907
Média: 1619919
Desvio padrão: 53369
Tráfego: 17182 pacotes
Percentual: 90,43%
Descrição da interface: Interface com a Universidade de Rio Grande (linha com velocidade de 19200bps). Geralmente apresenta muitos erros e excesso de tráfego.

Objeto monitorado: *ifInErrors*.4 (taxerros) - Interface Serial
Máquina: tchepoa
Dia: 13/11/96
Hora: 16:00
Valor que ultrapassou o limite: 6350
Média: 1351
Desvio padrão: 1099
Tráfego: 106200 pacotes
Percentual: 5.98%
Descrição da interface: Interface com a Universidade Federal de Santa Maria (linha com velocidade de 64kbps). Geralmente apresenta muitos erros.

Objeto monitorado: *ifInOctets*.2 e *ifOutOctets*.2 (taxutil) - Interface Serial
Máquina: tchepoa
Dia: 19/11/96
Hora: 10:00
Valor que ultrapassou o limite: 6878914
Média: 4451125
Desvio padrão: 742820
Tráfego: 15286 pacotes
Percentual: 80,45%
Descrição da interface: Interface com a Universidade de Rio Grande (linha com velocidade de 19200bps). Geralmente apresenta muitos erros excesso de tráfego.

Objeto monitorado: *ifInErrors*.5 (taxerros) - Interface Serial
Máquina: tchepoa
Dia: 04/12/96
Hora: 23:00
Valor que ultrapassou o limite: 285
Média: 48

Desvio padrão: 21
Tráfego: 1780 pacotes
Percentual: 16.01%
Descrição da interface: Interface com a URCAMP (linha urbana não especializada com modem banda base de 14400bps).

Objeto monitorado: *ifInOctets*.2 e *ifOutOctets*.2 (taxutil) - Interface Serial
Máquina: tchepoa
Dia: 13/12/96
Hora: 21:00
Valor que ultrapassou o limite: 7028308
Média: 5811535
Desvio padrão: 264852
Tráfego: 15618 pacotes
Percentual: 82.20%
Descrição da interface: Interface com a Universidade de Rio Grande (linha com velocidade de 19200bps). Geralmente apresenta muitos erros excesso de tráfego.

Objeto monitorado: *ifInErrors*.4 (taxerros) - Interface Serial
Máquina: tchepoa
Dia: 13/12/96
Hora: 21:00
Valor que ultrapassou o limite: 2057
Média: 293
Desvio padrão: 224
Tráfego: 7854 pacotes
Percentual: 26.19%
Descrição da interface: Interface com a Universidade Federal de Santa Maria (linha com velocidade de 64kbps). Geralmente apresenta muitos erros.

Objeto monitorado: *ifInErrors*.5 (taxerros) - Interface Serial
Máquina: tchepoa
Dia: 14/12/96
Hora: 9:00
Valor que ultrapassou o limite: 280
Média: 83
Desvio padrão: 30
Tráfego: 2357 pacotes
Percentual: 23.57%
Descrição da interface: Interface com a URCAMP (linha urbana não especializada com modem banda base de 14400bps).

Objeto monitorado: *ifInErrors*.2 (taxerros) - Interface Serial
Máquina: tchepoa
Dia: 14/12/96
Hora: 10:00
Valor que ultrapassou o limite: 510
Média: 261
Desvio padrão: 99
Tráfego: 6824 pacotes
Percentual: 7.47%
Descrição da interface: Interface com a Universidade de Rio Grande (linha com velocidade de 19200bps). Geralmente apresenta muitos erros excesso de tráfego.

Objeto monitorado: *ifInOctets*.5 e *ifOutOctets*.5 (taxutil) - Interface Serial
Máquina: bb2.pop-rs.rnp.br
Dia: 17/12/96
Hora: 23:00
Valor que ultrapassou o limite: 852001452
Média: 92783662
Desvio padrão: 34982720
Tráfego: 1893336 pacotes
Percentual: 92,44%
Descrição da interface: Interface com a URCAMP (linha urbana não especializada com modem banda base de 14400bps).

Observa-se que os problemas encontrados situaram-se nas conexões WAN. As taxas de erros elevadas que episodicamente acontecem derivam da má qualidade das linhas de comunicação de dados existentes na país. Quando uma situação deste tipo emerge, investiga-se a perenidade da mesma antes de acionar a concessionária. Se após algum tempo (uma a duas horas) a taxa de erros retorna ao normal, a chamada é fechada pelo técnico local. Se a taxa de erros persiste, á aberto um chamado junto à concessionária (CRT).

No que tange ao outro tipo de problema observado com maior freqüência que foi o de enlace com excesso de tráfego, a providência cabível depende da intensidade com que tal problema ocorre. Se é episódico, pode ser simplesmente contado. Se o excesso de tráfego se mantém, cabe determinar providências para expansão da capacidade da transmissão instalada.

7 Conclusões e trabalhos futuros

O trabalho realizado teve como objetivo principal a definição de um sistema que fosse inteligente o bastante para transformar eventos, recebidos do Sistema de Alertas, em alertas ao administrador da rede, caso esses eventos fossem considerados graves e poderiam de alguma forma prejudicar o desempenho da rede. Para cada alerta gerado, uma recomendação deve ser fornecida informando cursos de ações para se chegar na solução do problema. Este trabalho foi realizado para formar um dos módulos do ambiente CINEMA, atuando como um integrador entre o Sistema de *Trouble Ticket* e o Sistema de Alertas, sendo responsável pela gerência de falhas e desempenho da rede.

Inicialmente definiu-se o conjunto de objetos da MIB II que foi julgado importante de ser gerenciado e o qual pode ser aplicado na gerência de falhas e desempenho. Após essa análise, fez-se um estudo sobre quais limiares seriam considerados normais para determinado objeto. A maioria desses limiares foram definidos a partir de monitorações na rede. Esses limiares foram então usados para compor as regras que fariam parte do sistema. A monitoração dos objetos foi feita nas máquinas especificadas na seção 5.4 e os valores obtidos foram sendo armazenados em base de dados do Sistema de Banco de Dados Postgres.

Para que o sistema pudesse efetuar a detecção automática de problemas foi preciso usar técnicas de inteligência artificial. Para tanto, fez-se um estudo sobre as formas de representação do conhecimento e verificou-se que a utilização de regras de produção seria bastante apropriada para ser utilizada no sistema em questão. Redes semânticas também foram utilizadas, uma vez que com elas pode-se ter uma visão global sobre os problemas que o sistema pode identificar.

Uma vez definida todas as regras do sistema, construiu-se um protótipo com o objetivo de avaliá-las. Esse protótipo, conforme comentado no capítulo 6, foi desenvolvido na linguagem C e utilizou-se o pacote de software da CMU para realizar as monitorações nos objetos da MIB II. Essas monitorações foram feitas com a utilização de operações de *get* do protocolo SNMP.

No início encontrou-se dificuldades em relação à utilização do Sistema de Banco de Dados Postgres, devido à pouca documentação disponível. Uma outra dificuldade encontrada foi em relação aos problemas que podem ser diagnosticados pela monitoração dos objetos da MIB II. Atualmente existe pouca documentação que indique aspectos característicos de determinados problemas e formas de remediá-los ou solucioná-los. Além disso, outra dificuldade encontrada foi em relação aos testes a serem feitos no protótipo. Seria interessante que os testes fossem feitos sem muitas interrupções, mas como eles estavam sendo realizados no Instituto de Informática e diariamente ocorria falta de luz ou algum problema que isolava a sub-rede de onde partiam os testes, sua avaliação estava sendo prejudicada. Para remediar este problema as bases de dados contendo a *baseline* da rede, bem como a implementação do

protótipo, foram transferidas para outra máquina que localiza-se no CPD da UFRGS, a máquina "penta", onde essa fase de teste pode ser feita sem tantas interrupções.

Como uma seqüência desse trabalho seria interessante que o sistema aprendesse as soluções utilizadas pelos administradores para resolver os problemas e incluísse-as na sua base de regras. Atualmente, quando algum alerta deve ser gerado, faz-se uma pesquisa na base de registros de problemas para verificar se o mesmo problema já ocorreu anteriormente e apresentar a solução encontrada como sendo uma sugestão do novo registro a ser criado. Além disso, também seria interessante que o sistema tentasse corrigir automaticamente um problema antes que um alerta fosse gerado. Pode-se também pensar em incluir outras MIBs, como por exemplo a MIB da Cisco, quando se estiver gerenciando um roteador Cisco, sendo que uma nova base de regras deve ser construída para a MIB a ser incluída.

Atualmente, está em desenvolvimento um trabalho de Dissertação de Mestrado no qual será feita uma análise nos registros de problemas existentes para que possam ser criadas novas regras. Essa análise será efetuada nos problemas diagnosticados e nas soluções utilizadas para resolvê-los. As novas regras serão então utilizadas para tentar, através de probabilidades, apresentar automaticamente uma solução para um novo problema encontrado. Caso a solução apresentada não seja satisfatória, uma nova solução deverá ser apresentada.

Por fim, os objetivos desse trabalho foram cumpridos. Durante a avaliação do protótipo vários eventos foram criados, mas a grande maioria não foi transformada em alertas. Normalmente, quando a quantidade de tráfego era levada em consideração verificava-se que não havia um percentual significativo para transformá-lo em um alerta. Isso mostra a utilidade deste sistema, pois sem ele todos os eventos gerados seriam enviados ao administrador desnecessariamente, sobrecarregando-o. Além disso, como esse sistema realiza uma vigilância contínua sobre a rede, ele fornece uma visão geral da situação a técnicos e usuários e também apoia análise estatística e estudo de tendências da rede.

Dos alertas gerados durante o período de avaliação não foi tentado verificar a causa que poderia estar elevando o percentual de determinada regra. Como um alerta só é gerado depois de ser transformado em um evento e passar por uma base de regras, que verifica se determinado percentual está realmente elevado, todos os alertas gerados foram considerados válidos e dão uma idéia de como está o comportamento da rede naquele momento.

O sistema trabalha com o conhecimento equivalente a mais de um especialista humano, uma vez que seu conhecimento pode ser adquirido de várias pessoas. Além disso, esse conhecimento pode ser aumentado à medida que novas regras são adicionadas "manualmente" no sistema. Outro fator importante que pode ser observado é que o sistema pode ficar trabalhando um dia inteiro sem precisar de interrupções, como aconteceria se fosse feito diretamente por seres humanos. Por esses motivos, considera-se o protótipo desenvolvido de grande valia para a administração de uma rede no que tange o gerenciamento de falhas e desempenho.

Anexo A-1 *Baseline* da Rede

A *baseline* da rede da UFRGS foi formada a partir da monitoração de determinados objetos da MIB II em máquinas específicas, apresentadas na seção 5.2. Através de operações de *get* do protocolo SNMP, os dados foram coletados para então serem armazenados em bases de dados do Postgres. Por se tratar de objetos contadores, cada valor coletado foi subtraído de seu antecessor. Objetos relacionados com gerência de desempenho e de falhas, descritos nas tabelas a seguir, formam selecionados para compor esta *baseline*.

A monitoração do objeto *sysUpTime* é muito útil no gerenciamento de falhas. Sua monitoração pode determinar se a entidade foi reinicializada: se o valor desse objeto aumenta ao longo do tempo, significa que a entidade está normal, se o valor de *sysUpTime* é menor que o valor coletado anteriormente, então a entidade foi reinicializada. Isto pode significar alguma falha interna na entidade sendo monitorada. O objeto *sysUpTime* faz parte do grupo *system* e é um dos objetos que compõe a *baseline* da rede.

A tabela A-1.1 apresenta os objetos que formam monitorados do grupo interfaces. Com a monitoração desses objetos é possível descobrir taxas importantes para o gerenciamento de desempenho da rede, como por exemplo, a taxa de utilização de determinada interface, a taxa de pacotes com erro que estão entrando por uma interface, entre outras. Os objetos do grupo interfaces oferecem dados sobre cada interface específica de determinado equipamento da rede.

Os objetos selecionados dos grupos IP, ICMP e TCP, mostrados nas tabelas A-1.2, A-1.3 e A-1.4, respectivamente, também são utilizados no gerenciamento de desempenho da rede. Com a monitoração nesses objetos pode-se verificar se os percentuais desses tipos de pacotes estão muito elevados.

TABELA A-1.1 - Objetos monitorados do grupo interfaces.

Objeto	Descrição
ifInOctets	octetos recebidos por uma interface.
ifOutOctets	octetos transmitidos de uma interface.
ifInUcastPkts	pacotes *unicast* recebidos por uma interface.
ifOutUcastPkts	pacotes *unicast* transmitidos de uma interface.
ifInNUcastPkts	pacotes não-*unicast*, isto é, pacotes *broadcast* ou *multicast* recebidos por uma interface.

ifOutNUcastPkts	pacotes não-*unicast*, isto é, pacotes *broadcast* ou *multicast* transmitidos de uma interface.
ifInErrors	pacotes que entraram por uma interface possuindo algum erro.
ifOutErrors	pacotes que não puderam ser transmitidos por uma interface pois possuíam algum erro.
ifInDiscards	pacotes que entraram por uma interface mas foram descartados, mesmo não havendo erro.
ifOutDiscards	pacotes que foram descartados antes de serem entregues, mesmo não havendo erro.

TABELA A-1.2 - Objetos monitorados do grupo IP.

Objeto	Descrição
ipInReceives	número total de datagramas IP recebidos, incluindo aqueles com erros.
ipOutRequests	total de datagramas IP transmitidos.
ipInDiscards	datagramas recebidos sem erros mas que foram descartados.
ipOutDiscards	datagramas sem erros para serem transmitidos mas que foram descartados.
ipInHdrErrors	datagramas descartados devido a erros nos seus cabeçalhos IP.
ipInAddrErrors	datagramas descartados devido a erros nos seus endereços IP.
ipInUnknownProtos	datagramas com protocolo de nível superior desconhecido.
ipOutNoRoutes	número de vezes que a entidade não teve uma rota válida para o datagrama.
ipReasmFails	número de falhas detectadas pelo algoritmo de remontagem.
ipFragFails	datagramas com flag *Don't Fragment* em *on*.

TABELA A-1.3 - Objetos monitorados do grupo ICMP.

Objeto	Descrição
icmpInMsgs	mensagens ICMP recebidas, incluindo *icmpInErrors*.
icmpOutMsgs	mensagens ICMP que a entidade tentou enviar, incluindo *icmpOutErrors*.

icmpInDestUnreachs	mensagens de destino não alcançável recebidas.
icmpOutDestUnreachs	mensagens de destino não alcançável enviadas.
icmpInTimeExcds	mensagens de tempo excedido recebidas.
icmpOutTimeExcds	mensagens de tempo excedido enviadas.
icmpInParmProbs	mensagens indicando problemas de parâmetros recebidas.
icmpOutParmProbs	mensagens indicando problemas de parâmetros enviadas.
icmpInSrcQuenchs	mensagens *Source Quench* recebidas.
icmpOutSrcQuenchs	mensagens *Source Quench* enviadas.
icmpInRedirects	mensagens de *Redirects* recebidas.
icmpOutRedirects	mensagens de *Redirects* enviadas.

TABELA A-1.4 - Objetos monitorados do grupo TCP.

Objeto	**Descrição**
tcpInSegs	número total de segmentos recebidos.
tcpOutSegs	número total de segmentos enviados.
tcpAttemptFails	conexões que não foram estabelecidas.
tcpRetransSegs	número total de segmentos retransmitidos.
tcpCurrEstab	número total de conexões estabelecidas.
tcpMaxConn	número máximo de conexões que podem estar abertas.

Após um mês de monitoração de todos os objetos, calculou-se a média e o desvio padrão para cada um deles, para cada máquina, dependendo da hora do dia. Um exemplo de um objeto que foi monitorado e encontra-se na *baseline* pode ser observado na tabela A-1.5. Esta tabela apresenta a *baseline* do objeto *ifInOctets* para a interface 1, interface Ethernet, da máquina "routcc" (roteador do Campus Centro). A segunda coluna corresponde à média de todos os valores amostrados e a terceira corresponde ao desvio padrão.

TABELA A-1.5 - *Baseline* do objeto *ifInOctets* para a máquina "routcc".

hora	Média	Desvio Padrão
01-00	964235315	1634539744
02-01	841037202	1467104338
03-02	1248724536	1856835833
04-03	683793603	1432969916
05-04	688437083	1470498404
06-05	731405446	1292994103
07-06	631845447	1217202541
08-07	1019128429	1677971977
09-08	552182835	1156448019
10-09	1185758413	1653648168
11-10	674414898	1307336594
12-11	1111009781	1647659125
13-12	1186012227	1736990433
14-13	545014805	1157954578
15-14	1639716986	1820199868
16-15	971554902	1644642602
17-16	1028547971	1643751765
18-17	1254121437	1787894376
19-18	1118405871	1651693788
20-19	614507802	1178890928
21-20	884996702	1533317930
22-21	1138162727	1782469425
23-22	574964397	1353866860

A seguir alguns gráficos apresentam o percentual de utilização do roteador "bb2.pop-rs.rnp.br". Através da figura A-1.1 pode-se observar que o percentual de 30% não foi atingido. Portanto, a partir dessa análise e dos percentuais de outros roteadores verificou-se que este é um limite razoável para ser utilizado pela regra quando a interface monitorada for do tipo Ethernet.

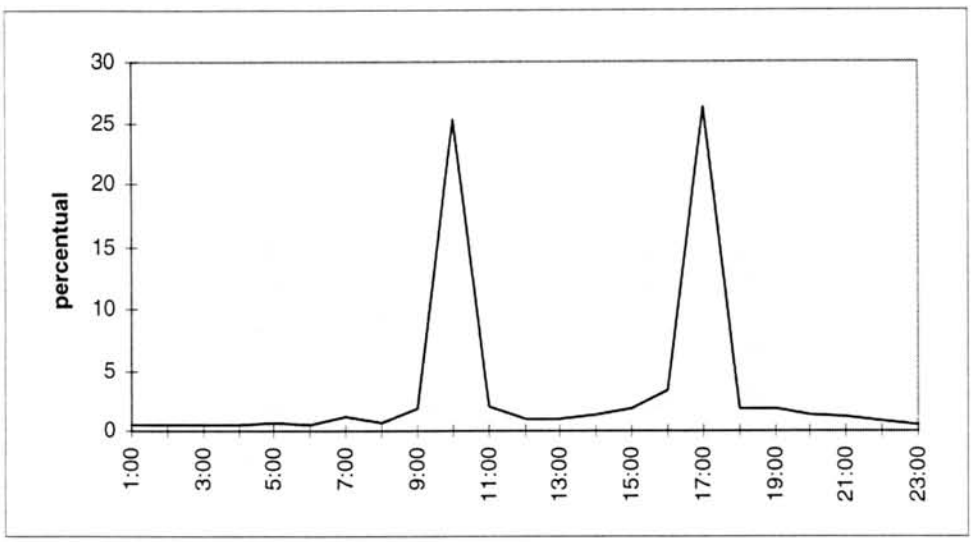

FIGURA A-1.1 - Percentual de utilização de uma interface Ethernet do "bb2.pop-rs.rnp.br".

A figura A-1.2 apresenta o gráfico do percentual de ocupação para uma interface Serial do roteador "bb2.pop-rs.rnp.br". Neste gráfico pode-se observar que a partir das nove horas a utilização começa a crescer. Analisando-se os percentuais de outros roteadores verificou-se que na maior parte do tempo este percentual não ultrapassa 50%. Portanto, este é o percentual utilizado para a regra de percentual de utilização quando a interface é Serial.

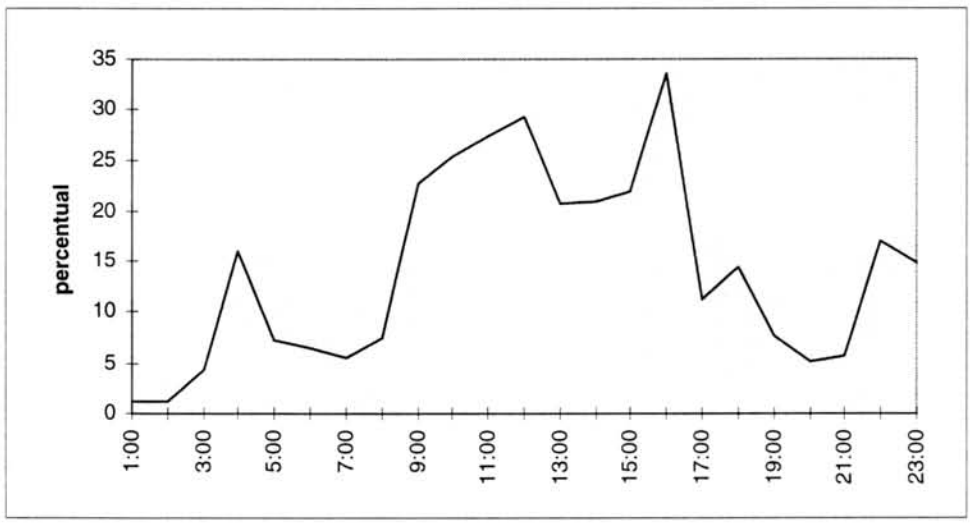

FIGURA A-1.2 - Percentual de utilização de uma interface Serial do "bb2.pop-rs.rnp.br".

As figuras A-1.3 e A-1.4 apresentam o percentual de erros dos roteadores "routcc" e "routcv" para uma interface Ethernet. Como pode-se observar nas figuras abaixo, o percentual de erros para cada roteador não ultrapassou 2% do tráfego total.

Para o roteador "routcc", o percentual de entrada mais alto encontra-se entre onze e treze horas e para o roteador "routcv" este mesmo percentual encontra-se entre nove e treze horas. Mas na maior parte do tempo ele se mantém estável. O mesmo foi observado para o roteador "bb2.pop-rs.rnp.br", onde o percentual de erros para uma interface Serial também não ultrapassou 1% do tráfego total.

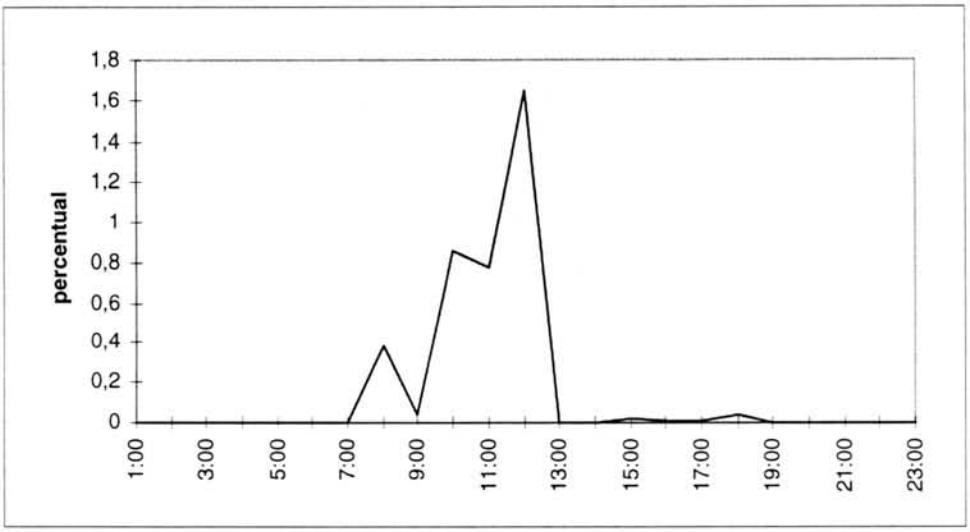

FIGURA A-1.3 - Percentual de erros de uma interface Ethernet do "routcc".

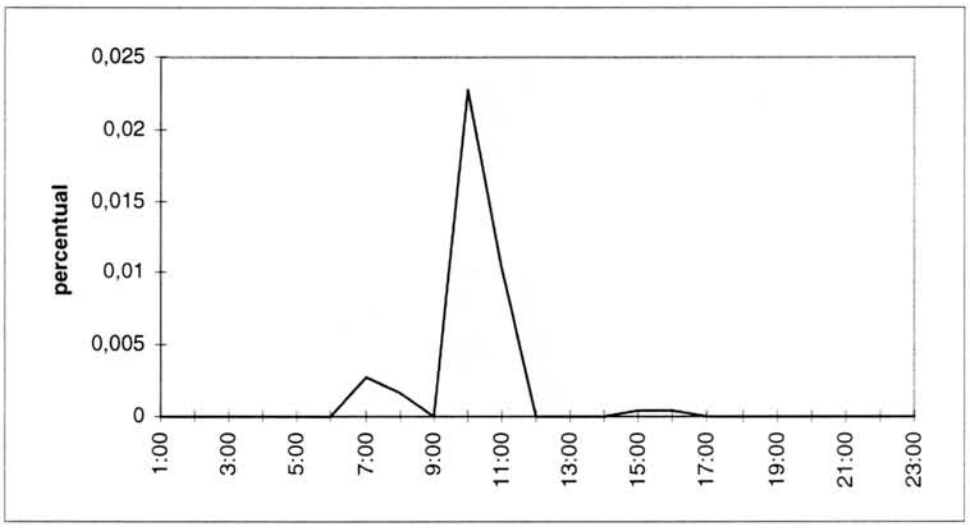

FIGURA A-1.4 - Percentual de erros de uma interface Ethernet do "routcv".

A figura A-1.5 apresenta um gráfico com o percentual de erros para uma interface Serial do roteador "tchepoa". Observando-se este gráfico é possível verificar que o percentual de erros em uma interface Serial é maior do que para uma interface Ethernet. Portanto, o percentual de erros utilizado em sua respectiva regra foi estipulado

em 5% quando a interface for Serial. Um alerta somente será gerado quando o percentual ultrapassar este valor.

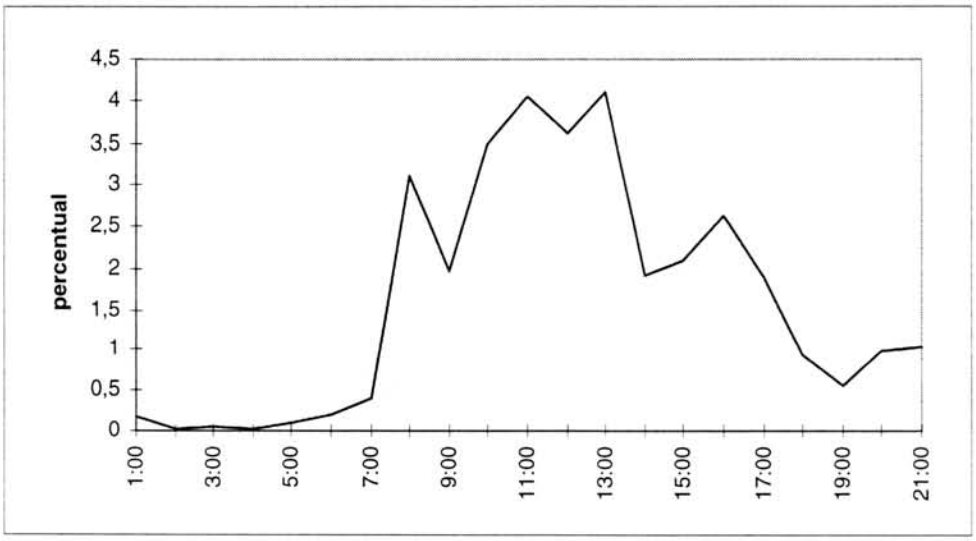

FIGURA A-1.5 - Percentual de erros de uma interface Serial do "tchepoa".

Anexo A-2 Utilização do Protótipo

O primeiro passo a ser feito quando se for utilizar o protótipo implementado é preencher o arquivo de configuração com os parâmetros descritos no capítulo 6. O arquivo de configuração deve ser preenchido dando um espaço entre uma informação e outra.

Quando alguma equação precisar ser monitorada, deve-se colocar antes do nome do objeto o nome da equação seguido por dois pontos ":" e em seguida colocar o objeto que deve compor a parte superior da equação, como por exemplo, para a equação que calcula o percentual de erros de uma interface coloca-se: "*taxerros:interfaces.ifEntry.ifTable.ifInErrors.1*". Os objetos para o cálculo do tráfego (*ifInUcastPkts* e *ifInNUcastPkts*, para esse caso) não devem ser informados, pois eles são monitorados automaticamente pelo sistema quando este tipo de equação for solicitado.

As equações que o sistema reconhece são rotuladas da seguinte forma:

- **taxutil** - taxa de utilização de uma interface

Para interface Ethernet:

$$utilização = \frac{(8 * (\Delta(ifInOctets_{t_1}, ifInOctets_{t_0}) + \Delta(ifOutOctets_{t_1}, ifOutOctets_{t_0}))) / (t_1 - t_0)}{ifSpeed}$$

Para interface serial:

$$utilização = \frac{(8 * \max(\Delta(ifInOctets_{t_1}, ifInOctets_{t_0}) + \Delta(ifOutOctets_{t_1}, ifOutOctets_{t_0}))) / (t_1 - t_0)}{ifSpeed}$$

- **taxerros** - taxa de erros de uma interface

Taxa de erros de entrada:

$$taxa = \frac{ifInErrors}{ifInUcastPkts + ifInNUcastPkts}$$

Taxa de erros de saída:

$$taxa = \frac{ifOutErrors}{ifOutUcastPkts + ifOutNUcastPkts}$$

- **taxdiscards** - taxa de pacotes descartados por uma interface

Taxa de descartes de entrada:

$$taxa = \frac{ifInDiscards}{ifInUcastPkts + ifInNUcastPkts}$$

Taxa de descartes de saída:

$$taxa = \frac{ifOutDiscards}{ifOutUcastPkts + ifOutNUcastPkts}$$

- **taxip** - esta taxa é composta da seguinte forma: um objeto do grupo IP dividido pelo somatório de *ifInUcastPkts* e *ifInNUcastPkts* ou *ifOutUcastPkts* e *ifOutNUcastPkts* para cada interface. Exemplo:

$$taxa = \frac{ipInDiscards}{\sum_{i=0}^{n} (ifInUcastPkts_i + ifInNUcastPkts_i)}$$

onde: n ⇒ número de interfaces da máquina

- **taxicmp** - esta taxa é composta da seguinte forma: um objeto do grupo ICMP dividido pelo somatório de *ifInUcastPkts* e *ifInNUcastPkts* ou *ifOutUcastPkts* e *ifOutNUcastPkts* para cada interface. Exemplo:

$$taxa = \frac{ipOutMsgs}{\sum_{i=0}^{n} (ifOutUcastPkts_i + ifOutNUcastPkts_i)}$$

onde: n ⇒ número de interfaces da máquina

- **taxtcp** - esta taxa é composta da seguinte forma: um objeto do grupo TCP dividido pelo somatório de *ifInUcastPkts* e *ifInNUcastPkts* ou *ifOutUcastPkts* e *ifOutNUcastPkts* para cada interface. Exemplo:

$$taxa = \frac{tcpInSegs}{\sum_{i=0}^{n} (ifInUcastPkts_i + ifInNUcastPkts_i)}$$

onde: n ⇒ número de interfaces da máquina

Os rótulos "taxip", "taxicmp" e "taxtcp" podem ser utilizados para qualquer objeto do respectivo grupo que for solicitado monitoração. Por exemplo, caso queira-se monitorar a taxa de mensagens com tempo excedido que está entrado na entidade, utiliza-se o rótulo "taxicmp" seguido do objeto "*icmp.icmpInDestUnreachs.0*"

Um exemplo de uma entrada para o arquivo de configuração, utilizando-se equações, deve ser feita da seguinte forma:

"*1 routcv taxerros:interfaces.ifEntry.ifTable.ifInErrors.1 1 3600 48 1 0 0 0*"

Neste caso, o sistema irá monitorar a taxa de erros que está entrando pela interface 1 do roteador "routcv" durante dois dias no intervalo de uma hora.

Uma exceção para essa forma é dada quando for utilizado o rótulo "taxutil". Neste caso, deve-se informar os dois objetos que compõem a expressão (*ifInOctets* e *ifOutOctets*) com a respectiva interface que deve ser monitorada. Exemplo:

"*taxutil:interfaces.ifEntry.ifTable.ifInOctets.1,interfaces.ifEntry.ifTable.ifOutOctets.1*".

Após o arquivo de configuração estar preenchido, é necessário verificar se o programa que simula o Sistema de Alertas está no ar. Para ativá-lo basta chamá-lo da seguinte forma: "salerta &". Feito isso, pode-se executar o protótipo com o comando: "mad config.txt &", onde config.txt é o nome do arquivo de configuração criado pelo usuário. Para finalizar a execução do MAD deve-se utilizar o comando "kill" do sistema operacional.

Caso o usuário deseje monitorar um objeto que não se encontra na *baseline* da rede é necessário que ele informe o valor dos limites superior e inferior para aquele objeto. Como o objeto não está na *baseline* não há uma regra para ele, portanto caso um evento seja gerado ele será transformado imediatamente em um alerta e enviado ao administrador da rede. Neste caso, uma nova regra deveria ser incluída para que o evento pudesse ser tratado adequadamente.

Como sua implementação foi desenvolvida na linguagem "C", a base de regras do sistema encontra-se dentro do código fonte do programa. Portanto, a inclusão de novas regras não é um processo muito simples. Em primeiro lugar é preciso criar uma função para essa nova regra, incluindo sua chamada juntamente com as demais na função *trata_exp* mostrada abaixo. Nessa função são colocadas as chamadas das funções que implementam as regras utilizadas pelo MAD. Caso a regra a ser incluída seja relacionada com uma expressão, sua chamada deve ser incluída dentro dessa função, como as chamadas abaixo:

```
/* Função que envia os eventos que chegam à função apropriada. */
trata_exp(indice, valor, valor1, valor2)
int indice;
unsigned long valor;
unsigned long *valor1;
unsigned long *valor2;
{
   char buffer[200];
   int est=0, status_wait;
   pid_t childpid;

   if((childpid = fork()) < 0)        /* Criação do processo filho */
        err_dump("MAD: erro no fork.\n");
   else
        if(childpid == 0)
        {                                  /* Chamada das funções */
                if(!strcmp(nomeexp[indice], "estado"))
                        estado_int(valor, valor1[0], indice, &est, buffer);
                if(!strcmp(nomeexp[indice], "taxutil"))
                        percent_ocup(valor, valor1[0], indice);
                if(!strcmp(nomeexp[indice], "taxerros"))
                        taxa_erros(valor, valor1[0], valor2[0], indice);
```

```
                    if(!strcmp(nomeexp[indice], "taxdiscards"))
                            taxa_descartados(valor, valor1[0], valor2[0], indice);
                    if(!strcmp(nomeexp[indice], "taxbrd"))
                            taxa_broadcast(valor, valor2[0], indice);
                    if(!strcmp(nomeexp[indice], "taxicmp"))
                            taxa_icmp(valor, valor1, valor2, indice);
                     if(!strcmp(nomeexp[indice], "taxip"))
                            taxa_ip(valor, valor1, valor2, indice);
                     if(!strcmp(nomeexp[indice], "taxtcp"))
                            taxa_tcp(valor, valor1, valor2, indice);
                    exit(0);
            }
      waitpid(childpid, &status_wait, WNOHANG);
}
```

A nova função pode ser colocada em um arquivo ".c" separado, mas deve ser "linkada" juntamente com os demais arquivos ".c" que compõem o protótipo.

Quando um registro de problemas precisar ser criado por uma nova regra, a seguinte chamada de função deve ser incluída:

open_TT(máquina, alerta, recomendação);

onde:

máquina ⇒ nome da máquina para a qual o evento foi gerado;

alerta ⇒ problema diagnosticado;

recomendação ⇒ recomendação ou comentário sobre como resolver o problema.

A função *open_TT* pesquisará na base de registros de problemas se o mesmo problema ocorreu anteriormente buscando a solução empregada para ser incluída com as recomendações definidas pela nova regra. Além disso, essa função se comunica com o Sistema de *Trouble Ticket*, criando um novo registro de problemas e enviando um *mail* para a pessoa responsável.

Anexo A-3 Topologia da Rede da UFRGS

A rede da UFRGS está dividida geograficamente por três campi, o Campus Centro (CC), o Campus Saúde (CS) e o Campus do Vale (CV). A figura A-3.1 ilustra como é feita atualmente a interligação entre esses três campi.

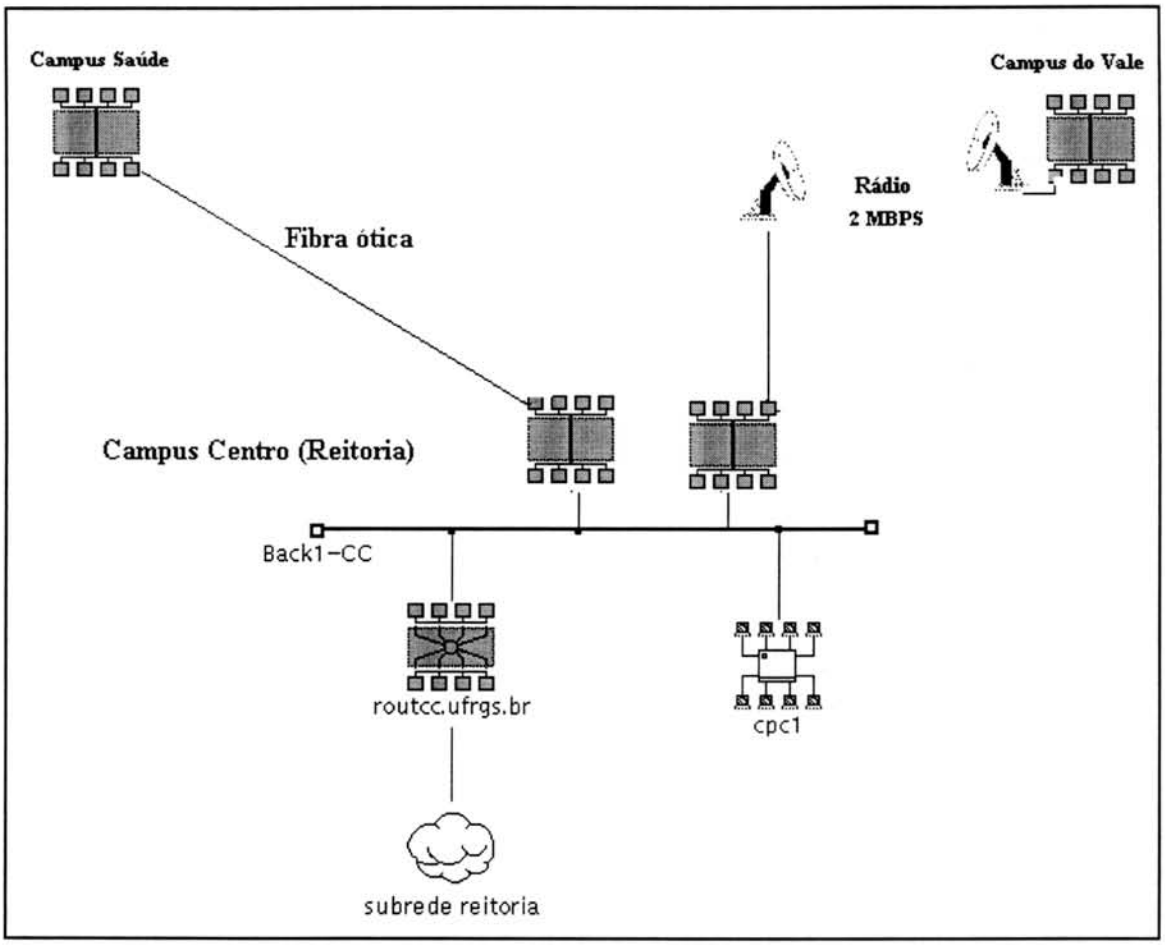

FIGURA A-3.1 - A interligação entre os três campi.

As figuras A-3.2, A-3.3 e A-3.4 ilustram como são feitas as ligações de cada curso da universidade com seu respectivo campus. A princípio, cada curso da universidade se localiza em um prédio diferente. Todos os prédios em um mesmo campus estão interligados por cabo de fibra ótica e em cada prédio existe uma rede local interna.

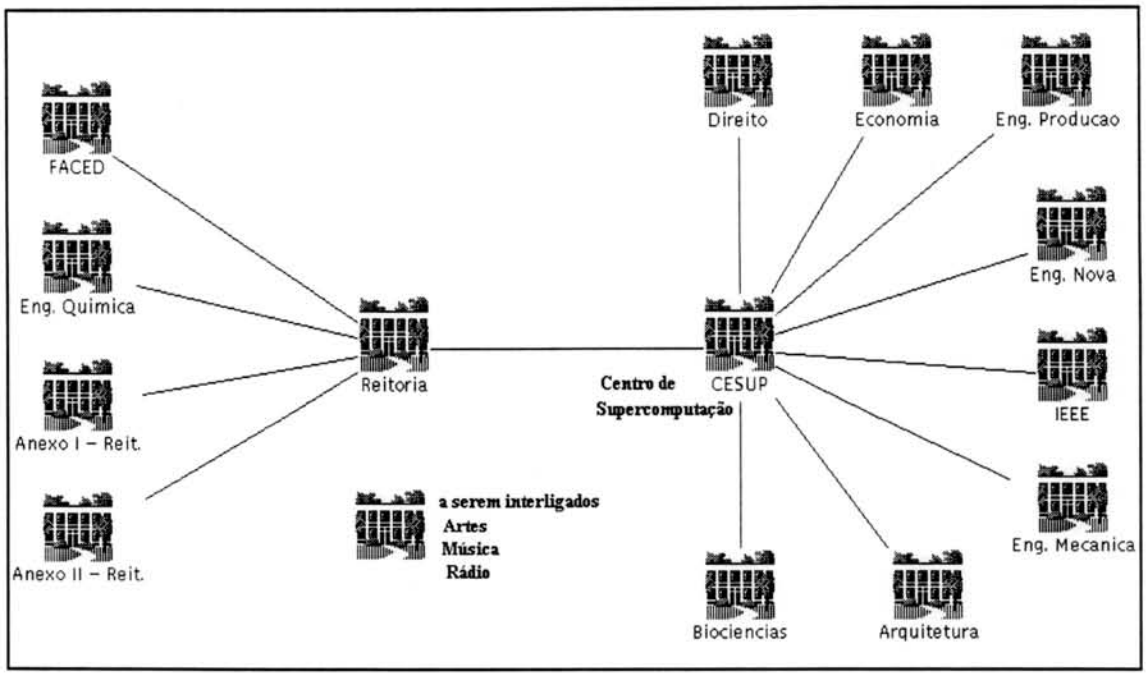

FIGURA A-3.2 - Campus Centro.

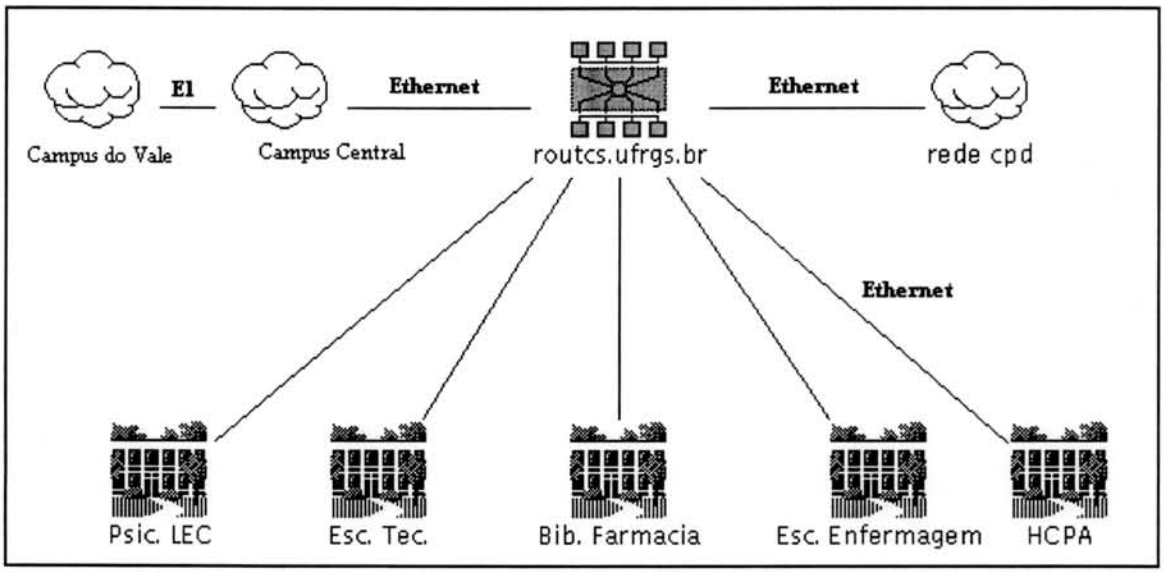

FIGURA A-3.3 - Campus Saúde.

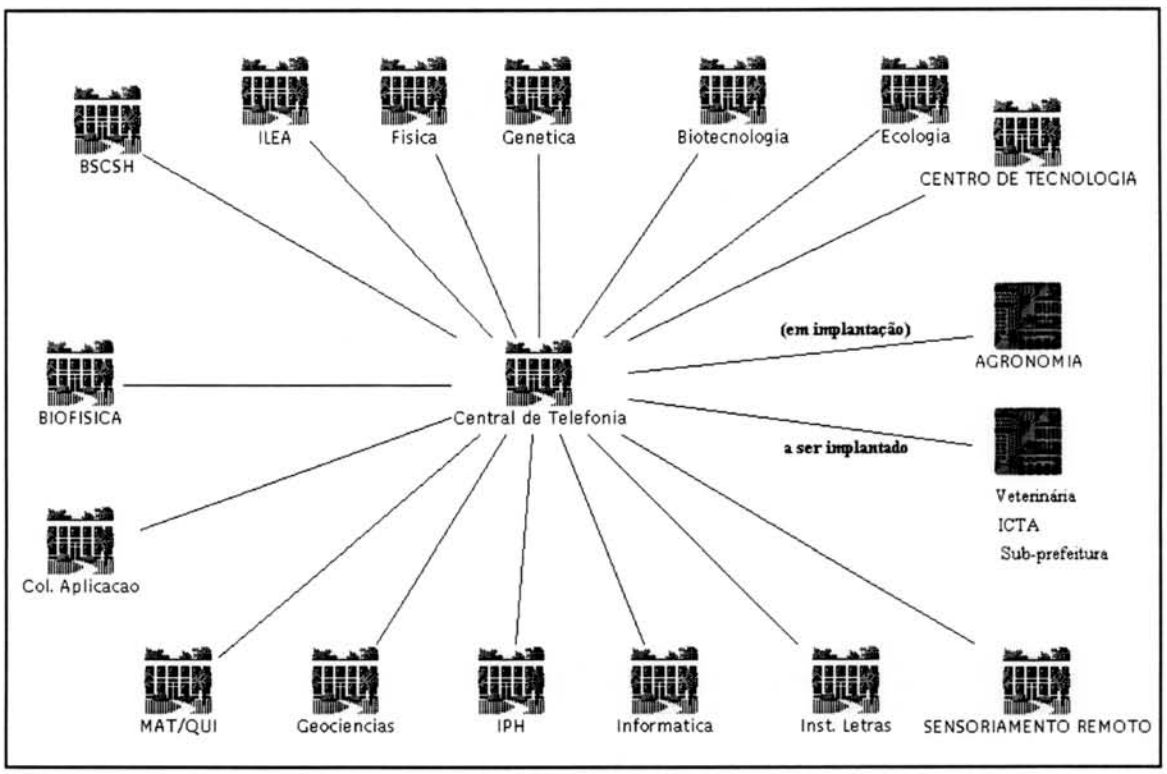

FIGURA A-3.4 - Campus do Vale.

A figura A-3.5 apresenta a interligação do POP/RS da RNP e o POP da rede TCHÊ com a rede da UFRGS.

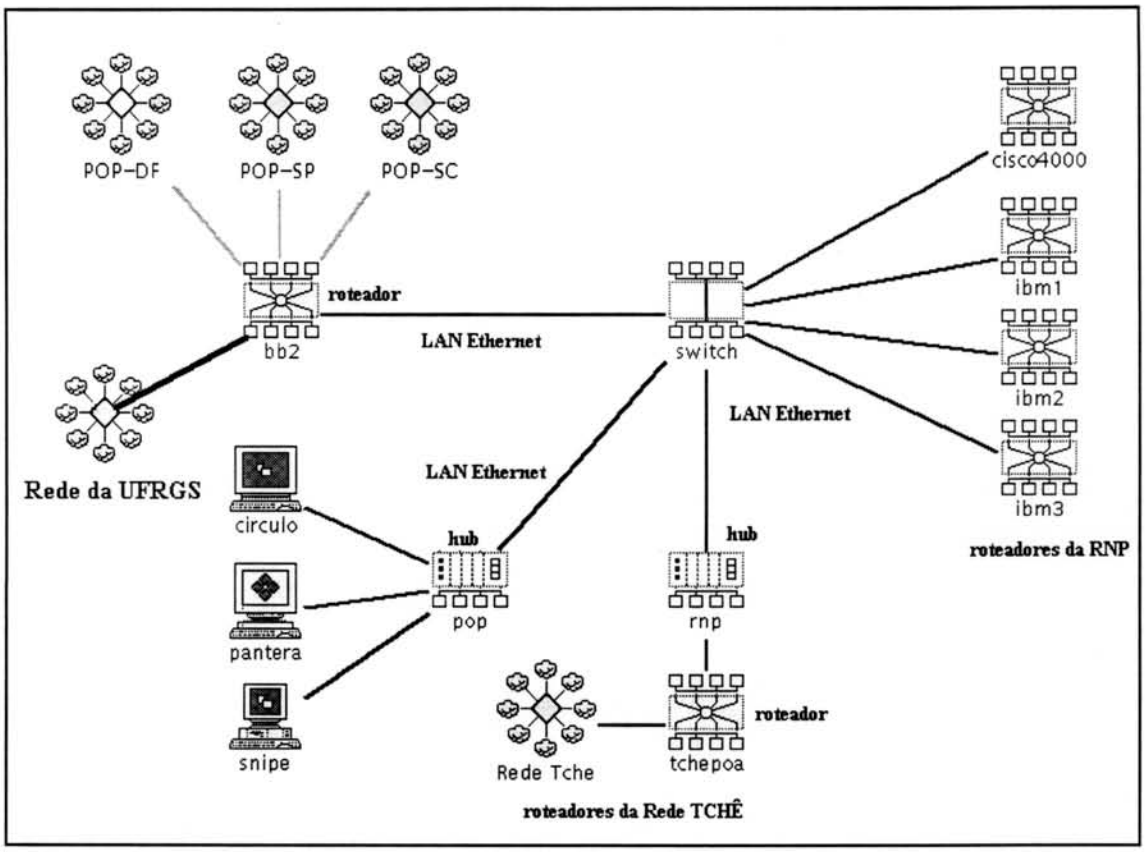

FIGURA A-3.5 - POP/RS, POP-Rede TCHÊ e interligação com a rede da UFRGS.

Bibliografia

[ART 96] ARTOLA, Esmilda S. **OLHO VIVO Sistema Inteligente para Gerência Pró-ativa Remota**. Porto Alegre: CPGCC da UFRGS, 1996. Dissertação de Mestrado.

[BRI 93] BRISA SOCIEDADE BRASILEIRA PARA INTERCONEXÃO DE SISTEMAS ABERTOS. **Gerenciamento de Redes:** Uma Abordagem de Sistemas Abertos. São Paulo: Makron Books do Brasil, 1993.

[BRO 93] BRONZATTI, Reges A. **Um Modelo de Gerência de Rede Baseado no Protocolo SNMP**. Porto Alegre: CPGCC da UFRGS, 1993. Dissertação de Mestrado.

[CAS 90] CASE, J. et al. **A Simple Network Management Protocol (SNMP)**. Request for Comments 1157, SNMP Research, Performance Systems International, MIT Laboratory for Computer Science, May, 1990. Disponível por FTP anônimo em nic.ddn.mil, no arquivo rfc/rfc1157.txt. (mar. 1995).

[COM 91] COMER, Douglas E. **Internetworking with TCP/IP**. 2. ed. Englewood Cliffs, NJ, EUA: Prentice-Hall, 1991. v.1.

[CRO 88] CRONK, Robert N.; CALLAHAN, P. H.; BERNSTEIN, L. Rule-Based Expert Systems for Network Management and Operations: An Introduction. **IEEE Network**, New York, v.2, n.5, p.7-21, Sept. 1988.

[GRI 93] GRILLO, P; WALDBUSSER, S. **Host Resources MIB**. Request for Comments 1514, Network Innovations, Carnegie Mellon University, Sept, 1993. Disponível por FTP anônimo em nic.ddn.mil, no arquivo rfc/rfc1514.txt. (mar. 1996).

[HAR 88] HARMON, Paul; DAVID, King. **Sistemas Especialistas**: A Inteligência Artificial chega ao Mercado. Rio de Janeiro: Campus, 1988.

[HAR 97] HARTMANN, Lisiane. **Gerência de Roteamento em Redes Interconectadas**. Porto Alegre: CPGCC da UFRGS, 1997. Dissertação de Mestrado.

[JAC 86] JACKSON, Peter. **Introduction to Expert Systems**. Workinghaw: Addison-Wesley, 1986.

[JAN 96] JANDER, Mary. Distributed Net Management: In Search of Solutions. **Data Communications**, New York, v.25, n.2, p.101-112, Feb. 1996.

[JOH 92] JOHNSON, D. **NOC Internal Integrated Trouble Ticket System: Functional Specification Wishlist**. Request for Comments 1297, Merit Network, Jan. 1992. Disponível por FTP anônimo em nic.ddn.mil, no arquivo rfc/rfc1297.txt. (mar. 1995).

[LEI 93] LEINWAND, Allan; FANG, Karen. **Network Management:** a practical perspective. Menlo Park, CA, EUA: Addison-Wesley, 1993. 222p.

[MAD 94] MADRUGA, Ewerton L. **Ferramentas de Apoio à Gerência de Falhas e Desempenho em Contexto Distribuído**. Porto Alegre: CPGCC da UFRGS, 1994. Dissertação de Mestrado.

[MAD 94] MADRUGA, E. L; TAROUCO, L.M.R. Fault Management Tools for a Cooperative and Decentralized Network Operations Environment. **IEEE Journal on Selected Areas in Communications**, New York, v.12, n.6, p.1121-1130, Aug. 1994.

[McC 91] McCLOGHRIE, K.; ROSE, M. **Management Information Base for Network Management of TCP/IP-based Internet: MIB-II**. Request for Comments 1213, Hughes LAN Systems, Mar. 1991. Disponível por FTP anônimo em nic.ddn.mil, no arquivo rfc/rfc1213.txt. (mar. 1995).

[McM 93] McMASTER, D.; McCloghrie, K. **Definitions of Managed Objects for IEEE 802.3 Repeater Devices**. Request for Comments 1516, SynOptics Communications, Hughes LAN Systems, Sept. 1993. Disponível por FTP anônimo em nic.ddn.mil, no arquivo rfc/rfc1516.txt. (abr. 1996).

[MED 96] MEDINA, Roseclea D. **SAFO - Sistema Agregador de Ferramentas de Operação de Rede**. Porto Alegre: CPGCC da UFRGS, 1996. Dissertação de Mestrado.

[MIL 89] MILLER, Mark A. **LAN Troubleshooting Handbook**. San Mateo, CA: M&T Books, 1989.

[MIL 91] MILLER, Mark A. **Troubleshooting Internetworks - Tools, Techniques, and Protocols**. San Mateo, CA: M&T Books, 1991.

[MUL 90] MULLER, N. J.; DAVIDSON, R. P. **LANs to WANs:** Network Management in the 1990s. Norwood, MA: Artech House, 1990.

[NAS 94] NASSER, Dan. **Network Optimization and Troubleshooting:** achieve maximum network performance. Indianapolis, Indiana: New Riders Publishing, 1994.

[NUN 95] NUNES, Cristina M. **Ferramentas de Monitoração de Rede**. Porto Alegre: CPGCC da UFRGS, 1995. (Trabalho Individual).

[NMT 96] NMT - Network Management Technologies. Documento disponível em http://www.ozemail.com.au/~mhammett/. (jul. 1996).

[PAS 86] PASQUALE, Joseph. **Knowledge-Based Distributed Systems Management**. Berkeley: University of California, 1986. (Relatório N° UCB/CSD 86/295).

[PIC 95] PICOTO, Carlos; VEIGA, Pedro. Management of a WWW Server using SNMP. In: JOINT EUROPEAN NETWORKING CONFERENCE, JENC6., 1995, Tel Aviv, Israel. **Proceedings....** Disponível em http://www.terema.nl/jenc6/413.ps. (jul. 1996).

[POS 81] POSTEL, J. **Internet Control Message Protocol**. Request Comments 0792, ISI, Darpa Internet Program Protocol Specification, Sept. 1981. Disponível por FTP anônimo em nic.ddn.mil, no arquivo rfc/rfc0792.txt. (abr. 1996).

[ROS 90] ROSE, Marshall. **The Simple Book:** An Introduction on to Management of TCP/IP - based Internets. Englewood Cliffs, NJ: Prentice-Hall, 1990.

[ROS 95] ROSE, Marshall; McCLOGHRIE, Keith. **How to Manage Your Network Using SNMP.** Englewood Cliffs, NJ: PTR Prentice-Hall, 1995.

[SIL 95] SILVA, Ana Cristina B. da. **Uma Proposta para Gerência de Correio Eletrônico**. Porto Alegre: CPGCC da UFRGS, 1995. Dissertação de Mestrado.

[SNG 90] SNG, D.C.H. **Network Monitoring and Fault Detection on the University of Illinois at Urbana-Champaign Campus Computer Network**. Urbana-Champaign, IL, EUA: DCS/UIUC, 1990. (Relatório Técnico UIUCDCS-R-90-1595).

[STO 90] STONEBRAKER, M. **POSTGRES reference manual**. Berkeley, CA, EUA: University of California, 1990. 175p.

[SOU 95] SOUZA, José Neuman de; OLIVEIRA, A. Mauro B. de. **Curso Avançado sobre Gerenciamento de Redes**. Fortaleza, CE: LAR - Laboratório Multiinstitucional de Redes e Sistemas Distribuídos, ETFCE - UFC - UECE, 1995.

[TAN 89] TANEMBAUM, Andrew S. **Computer Network**. 2.ed. Amsterdam: Prentice-Hall, 1989.

[TAR 90] TAROUCO, Liane M. R. **Inteligência Artificial Aplicada ao Gerenciamento de Redes de Computadores**. São Paulo: Escola Politécnica da USP, 1990. Tese de Doutorado.

[TAR 96] TAROUCO, Liane M. R. et al. **Um ambiente para gerenciamento integrado e cooperativo**. Fortaleza: UFC, 1996. Trabalho apresentado no Segundo Workshop sobre Administração e Integração de Sistemas, 1996, Fortaleza.

[TRI 92] TRINDADE, Rodrigo S. **Um estudo da linguagem SDL para especificação e teste de protocolos**. Porto Alegre: CPGCC da UFRGS, 1992. 87p. (Trabalho Individual).

[WAL 95] WALDBUSSER, S. **Remote Network Monitoring Management Information Base**. Request for Comments 1757, Carnegie Mellon University, Feb. 1995. Disponível por FTP anônimo em nic.ddn.mil, no arquivo rfc1757.txt. (maio. 1995).

Informática  UFRGS

CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Um Discriminador Inteligente de Eventos de Rede para o Ambiente CINEMA.

por

Cristina Moreira Nunes

Dissertação apresentada aos Senhores:

Profa. Dra. Tereza Cristina Melo de Brito Carvalho (EPUSP)

Profa. Dra. Maria Janilce Bosquiroli Almeida

Prof. Dr. Raul Fernando Weber

Prof. Juergen Rochol

Vista e permitida a impressão.
Porto Alegre, 04 / 06 / 97.

Profa. Dra. Liane Margarida Rockenbach Tarouco,
Orientador.

Profa. Carla Maria Dal Sasso Freitas
Coordenadora Substituta do Curso de
Pós-Graduação em Ciência da Computação
Instituto de Informática - UFRGS