ANDERSON WILLIAN ZANELATTO

DESENVOLVIMENTO DE UM SISTEMA DE APOIO A GERÊNCIA DE FALHAS BASEADO EM CASOS - ANTIFAIL

CRICIÚMA, JUNHO DE 2008

ANDERSON WILLIAN ZANELATTO

DESENVOLVIMENTO DE UM SISTEMA DE APOIO A GERÊNCIA DE FALHAS BASEADO EM CASOS - ANTIFAIL

Trabalho de Conclusão de Curso apresentado para obtenção do Grau de Bacharel em Ciência da Computação da Universidade do Extremo Sul Catarinense.

Orientador: Prof. MSc. Paulo João Martins

Co-Orientador: Prof ª. MSc. Merisandra Côrtes de Mattos

CRICIÚMA, JULHO DE 2008

“Eu aprendi que para se crescer como pessoa é preciso me cercar de gente mais inteligente do que eu.”

William Shakespeare

Dedico esta conquista a meus pais, que estiveram sempre presentes incentivando na busca dos meus ideais.

AGRADECIMENTOS

Agradeço esta conquista a Deus, aos meus pais, Aldo Zanelatto e Graciosa Mariot Zanelatto, que sempre presentes, me incentivaram a lutar por meus objetivos. A minha irmã, Taise Fernanda Zanelatto, ao meu cunhado Anderson Joaquim e a minha linda sobrinha Ana Júlia Zanelatto Joaquim, pela alegria e carinho dedicados. Que eu seja um bom exemplo para eles e agradeço a toda família pelo apoio incondicional.

Agradeço também aos meus amigos, que sempre apoiaram e deram a força para continuar na caminhada sem desanimar, em especial ao Pedro Paulo Alexandrino, Ramiro Webber Dimer, João Paulo Mendes e Karina de Campos, companheiros, quase irmãos. À minha namorada, Catia de Campos, pela compreensão e por estar sempre presente nos momentos mais difíceis, acolhendo e incentivando-me.

Ao orientador Paulo João Martins e a co-orientadora Merisandra Côrtes de Mattos, por terem me direcionado e mostrado o caminho para desenvolver esse trabalho

À todos os meus professores, pela educação e por me mostrarem o caminho da busca pelo saber e a todos da comunidade cientifica que me proporcionaram, a obtenção de uma enorme gama de conhecimento.

Agradeço também a todas as pessoas que não mencionei, mas que direta ou indiretamente estiveram envolvidas nessa conquista.

RESUMO

O aumento da utilização das redes de computadores nas organizações vem ganhando importância quanto às práticas de gerenciamento. Com o crescimento da diversidade dos equipamentos que a compõe, torna-se cada vez mais necessária a integração desses diferentes componentes. No entanto, é necessário que se faça o gerenciamento desses equipamentos para evitar problemas na rede. O gerenciamento de redes está dividido em cinco áreas funcionais: falhas, desempenho, configuração, contabilização e segurança. Nesta pesquisa, são apresentadas técnicas para a monitoração e o tratamento das informações acerca do gerenciamento de falhas em redes de computadores no modelo TCP/IP. As técnicas utilizadas para este gerenciamento são fundamentadas na utilização do protocolo SNMP. Além disso, o gerenciamento de redes de computadores pode beneficiar-se com uma das técnicas de Inteligência Artificial, o raciocínio baseado em casos, que por sua vez, possui a característica de relembrar casos ocorridos no passado para auxiliar na resolução de um problema atual. Na implementação desta técnica, o método utilizado para o cálculo da similaridade entre os casos foi o *nearest neighbour* ponderado. Ao associar a técnica de raciocínio baseado em casos com o gerenciamento de falhas, obteve-se o desenvolvimento de um protótipo, que tem a capacidade de emitir alarmes informando a ocorrência de determinados eventos que foram definidos na sua implementação. Este protótipo também permite a recuperação, adaptação e o armazenamento dos problemas ocorridos para serem utilizados futuramente, caso seja necessário.

Palavras-Chave: Gerenciamento de Redes de Computadores, Gerência de Falhas, Protocolo SNMP, Raciocínio Baseado em Casos.

ABSTRACT

The increase in the use of computer networks in organizations has been gaining importance on practices of management. With the increase of the diversity of equipments that compose them, it becomes more necessary the integration of these different components. However, it is a must to manage such equipments in order to avoid problems in the network. The management of networks is divided into five functional areas: fault, performance, configuration, accounting and security. In this research, it is presented techniques for monitoring and treatment of information about error management in computer networking in the model TCP / IP. The techniques used for this management are based on the use of SNMP protocol. Moreover, the management of computer networks may take advantage of one of the techniques of Artificial Intelligence, the reasoning based on cases, which in turn, has the characteristic of recalling cases occurred in the past to assist in resolving a today problem today. On the implementation of this technique, the method used to calculate the similarity between the cases was the nearest neighbour weighted. By associating the technique of reasoning based on cases with the fault management, it was obtained the development of a prototype, which has the ability to sound alarms informing the occurrence of certain events that were defined in its implementation. This prototype also allows the recovery, adaptation and storage of problems for future use if necessary.

Keywords: Management of Computer Networking, Fault Management, SNMP Protocol, Reasoning Based on cases.

LISTA DE ILUSTRAÇÃO

LISTA DE TABELAS

LISTA DE SIGLAS

BC	Base de Casos
BDT	Base de Dados Temporal
IA	Inteligência Artificial
IP	Internet Protocol
ISO	International Organization for Standardization
JDBC	Java Database Connectivity
LAN	Local Area Network
MAN	Metropolitan Area Network
MIB	Management Information Base
OID	Object Identifier
RBC	Raciocínio Baseado em Casos
SE	Sistemas Especialistas
SEP	Sistemas Especialistas Probabilísticos
SGR	Sistema de Gerenciamento de Redes
SGRBC	Sistema de Gerenciamento de Redes Baseado em Conhecimento
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
TTS	Trouble Ticket System
UFC	Universidade Federal do Ceará
UFLA	Universidade Federal de Lavras
UFRGS	Universidade Federal do Rio Grande do Sul
UML	Unified Modeling Language
WAN	Wide Area Network

SUMÁRIO

1 INTRODUÇÃO

O aumento da utilização das redes de computadores e suas aplicações, proporcionou um crescimento elevado das tecnologias deste mesmo domínio. Portanto, para que diferentes tipos de tecnologias funcionem em conjunto, surgiu assim a necessidade de se realizar o gerenciamento dos mesmos, pois juntamente com o aumento da utilização e variedade dessas tecnologias, veio o crescimento do número de problemas causados nestes ambientes de comunicação.

Gerenciar uma rede não é uma tarefa simples de se realizar, pois envolve a parte de *hardware* (meio físico) e de *software* (aplicação), onde os problemas poderão surgir de ambas as partes. Considerando que a tarefa de gerenciar uma rede é complexa, há necessidade da utilização de um técnico especialista para controlar, bem como manter a disponibilidade e qualidade dos serviços da rede por meio do gerenciamento das mesmas.

Devido à ocorrência destes problemas gera-se vários transtornos quanto a disponibilidade dos serviços oferecidos por este tipo de comunicação as organizações que a utilizam. Isso porque em muitas delas, as redes de computadores funcionam como a principal forma de comunicação interna e externa. Por exemplo, uma instituição financeira tem sua comunicação interrompida por algum tipo de falha na rede. Certamente, seus serviços seriam afetados consideravelmente. Como conseqüência, inúmeros clientes estariam deixando de realizar transações financeiras, o que ocasionaria prejuízos para ambas as partes, tanto à instituição financeira quanto aos clientes da mesma.

É por essas e outras razões, que o gerenciamento de rede se faz necessário, sendo que a área de gerência de falhas é a que vem ganhando mais atenção nos últimos

tempos, juntamente com a de segurança.

Com o objetivo de proporcionar o auxílio ao gerenciamento de falhas, de modo a tornar este mais eficaz e preciso com relação a sua administração, surge o conceito dos Sistemas de Gerenciamento de Redes Baseados em Conhecimento (SGRBC).

Dentre as várias técnicas da Inteligência Artificial, os sistemas de SGRBC abordam o paradigma de Raciocínio Baseado em Casos (RBC). Tais sistemas, propõem soluções para um problema atual pela recuperação de situações similares ocorridas anteriormente, conhecidas e denominadas de casos, que podem contribuir para a resolução do problema atual. Esses sistemas têm também como característica o aprendizado com a experiência, possuindo a capacidade de armazenar novas situações solucionadas, que se tornam disponíveis para futuras consultas e aumentam naturalmente o conhecimento presente no sistema conforme forem surgindo os problemas e as respectivas soluções.

1.1 OBJETIVO GERAL

Este trabalho tem por objetivo geral, realizar o desenvolvimento de um protótipo denominado *Antifail*, que consiste em um sistema baseado em casos para o auxílio ao gerenciamento de falhas em um ambiente de redes de computadores, utilizando também o protocolo *Simple Network Management Protocol* (SNMP), para auxiliar o processo de automação na obtenção das informações referentes às falhas ocorridas nos objetos gerenciados.

1.2 OBJETIVOS ESPECÍFICOS

Como parte integrante deste trabalho, pode-se citar como objetivos específicos, as seguintes tarefas:

a) utilizar agentes SNMP para atender as solicitações do gerente SNMP tendo como base de informações uma *Management Information Base* (MIB);
b) desenvolver um gerente SNMP para solicitar aos agentes SNMPs as informações desejadas referente ao gerenciamento de falhas;
c) desenvolver um sistema de RBC para tratar as informações relacionadas às falhas ocorridas na rede gerenciada, e desta forma tentar auxiliar os profissionais responsáveis pelo gerenciamento na resolução destas falhas ocorridas.

1.3 JUSTIFICATIVA

As redes de computadores estão se tornando cada vez mais importantes para a sociedade. Isso porque, estão ganhando proporções cada vez maiores, e conseqüentemente atingindo um número maior de usuários. Além disso, estão tornando-se mais heterogêneas e complexas, o que dificulta o seu gerenciamento por parte dos profissionais da área. No entanto, com o auxílio da técnica de RBC, pode-se então realizar esta tarefa.

Utilizando os benefícios que um sistema de raciocínio baseado em casos pode oferecer, associado às técnicas de monitoramento de redes, forma-se desta maneira uma base de conhecimento que pode se constituir em importante aliada para um

administrador de redes, pois desta forma, o tempo necessário para o descobrimento e correção de problemas em uma estrutura de redes pode ser reduzido significativamente, além de existir também a contribuição direta para manter o desempenho e estabilidade da rede.

Assim, por meio do desenvolvimento de um sistema de apoio a gerência de redes baseado em casos, pode-se obter um modelo de sistema para gerência de redes chamado Sistema de Gerenciamento de Redes Baseado em Conhecimento (SGRBC), que é formado por técnicas de gerenciamento de redes associadas às técnicas de raciocínio baseado em casos da Inteligência Artificial.

Tal modelo pode disponibilizar a princípio, uma vantagem sobre a maioria dos gerenciadores de redes, ou seja, o auxílio ao profissional da área de redes no momento de tomar decisões e proporcionar a adaptação ao surgimento de novos problemas. Com isso, os problemas ocorridos nas redes podem ser resolvidos em tempo menor e a comunicação poderá ser otimizada ou restabelecida rapidamente, proporcionando uma maior agilidade no desenvolvimento das atividades que antes estavam afetadas por algum problema na estrutura da rede. Além disso, o tempo economizado pelo profissional para resolver o(s) problema(s), poderá ser utilizado para desenvolver outras atividades na área.

1.4 ESTRUTURA DO TRABALHO

A presente pesquisa tem como meta demonstrar de maneira prática o gerenciamento de redes de computadores por meio do protocolo SNMP e com a utilização do raciocínio baseado em casos para dar suporte a decisões. Para tanto, o trabalho é dividido em três grandes partes: a primeira abordando os aspectos de

gerenciamento de redes de computadores. A segunda trata dos aspectos da inteligência artificial, mais precisamente do raciocínio baseado em casos, e a terceira a parte prática, que é baseada na fundamentação teórica apresentada, simulando um ambiente real.

O primeiro capítulo aborda o gerenciamento de redes de computadores de um modo geral, explanando sobre a importância de se gerenciar uma rede, aspectos acerca do protocolo SNMP e as áreas funcionais do gerenciamento, abordando de forma mais ampla a de falhas. O capítulo seguinte apresenta o raciocínio baseado em casos, sua estrutura, algumas técnicas específicas e o ciclo de RBC. E completando o trabalho, a parte prática, que demonstra como o sistema foi desenvolvido, além de uma breve introdução sobre sua correta utilização.

2 GERENCIAMENTO DE REDES

O gerenciamento de redes pode ser entendido como o processo de controlar uma rede de computadores de tal modo que seja possível maximizar sua eficiência e produtividade (FREITAS, 2001). Esse processo engloba um conjunto de funções integradas que podem estar em uma máquina ou em várias, dispersas há milhares de quilômetros, em diferentes organizações e residindo em máquinas distintas. É importante observar que, com estas funções, pode-se controlar uma rede de computadores e seus serviços, provendo mecanismos de monitoração, análise e controle dos dispositivos e recursos da mesma (TANENBAUM, 2003).

Uma rede de computadores deve ter a capacidade de suportar as aplicações para qual foi projetada, mantendo uma velocidade satisfatória, alta disponibilidade dos recursos e custos compatíveis com os serviços oferecidos, ou seja, manter um equilíbrio entre custo benefício (TANENBAUM, 1997).

A gerência de redes de computadores permite que se possa utilizar de mecanismos para manter uma corporação sob seu controle, e de forma integrada, com os recursos que compõem a sua infra-estrutura tecnológica para a manipulação das informações (FREITAS, 2001).

Ela compreende a monitoração, análise e resolução de eventuais problemas, dentre outras atividades necessárias para a manutenção de uma rede com qualidade de serviços adequada aos objetivos dos sistemas de informação (MELCHIORS, 1999).

A atividade de gerência cresce em importância e complexidade na proporção em que se diversificam o número de tecnologias de sistemas operacionais, de protocolos de rede e de elementos necessários para interconectar todos estes componentes. Com a constante evolução da tecnologia de redes, aumenta também a

freqüência com que surgem novos elementos agregados à rede corporativa, constituindo-se em novos elementos de rede a serem gerenciados (CASTELLIS, 2006).

A solução para tornar possível a gerência conjunta dos elementos de diferentes tecnologias e fabricantes em uma mesma rede, é por meio da utilização de protocolos não proprietários, ou seja, os chamados protocolos livres. Com isso, um mesmo sistema de gerência pode manipular informações de maneira uniforme e consistente, além de executar operações sobre um determinado elemento na rede, não importando o seu tipo ou seu fabricante. Além disso, poderão ser incorporados a qualquer momento, novos elementos na rede (KUROSE; ROSS, 2006).

Como pode-se observar, o gerenciamento de redes de computadores é muito importante, pois o mesmo é responsável pela gerência conjunta dos diferentes mecanismos de comunicação, sendo de fundamental importância para o bom funcionamento desta tecnologia de comunicação chamada redes de computadores, cuja utilização é de âmbito mundial.

2.1 IMPORTÂNCIA DA NECESSIDADE DE GERENCIAMENTO EM REDES DE COMPUTADORES

A necessidade de gerenciamento das redes de computadores deve-se principalmente ao fato da alta complexidade estrutural de implementação dessas redes (FREITAS, 2001). Isso porque pode-se encontrar diversas *Local Area Network* (LAN) interligadas local ou remotamente por meio de equipamentos de interconexão, como por exemplo os roteadores. Desta forma, pode-se afirmar que a gerência de uma rede é uma atividade bastante complexa porque envolve uma grande quantidade de variáveis associadas a softwares, hardwares e meios de comunicação, ou seja, os meios pelos

quais são transmitidos os dados. A alta complexidade no gerenciamento de redes de computadores pode ser observada sob vários aspectos, onde dois merecem maior destaque. A heterogeneidade e a interoperabilidade entre diversos domínios organizacionais (LOPES; SAUVÉ; NICOLLETTI, 2003).

A heterogeneidade dos componentes de vários sistemas provenientes de diferentes fabricantes dentro de um mesmo domínio corporativo, resulta na convivência de protocolos e formas de gerenciamento diferentes, não operando de uma maneira integrada e homogenia. Conseqüentemente, ocorre o aumento do custo e a falta de eficiência na operação e manutenção do sistema. Entre os diversos aspectos de heterogeneidade nas redes que influenciam diretamente a atividade de gerência, destacam-se a heterogeneidade no nível das arquiteturas dos sistemas interligados e a heterogeneidade no nível dos dados a serem transmitidos (HOLANDA FILHO, 1998).

A heterogeneidade no nível das arquiteturas dos sistemas interligados corresponde às redes interconectadas compostas de sub-redes distribuídas que possuem arquiteturas diferentes, tais como, LAN, *Metropolitan Area Network* (MAN) e *Wide Area Network* (WAN), e que necessitam de mecanismos de gerência específicos para cada uma. Nesses sistemas, os pontos fracos localizam-se nos dispositivos de interconexão. Esses dispositivos, lógicos ou físicos, são compartilhados pelas redes adjacentes que os gerenciam de formas diferentes, necessitando de um tratamento específico para cada tipo de rede (PETERSON; DAVIE, 2004).

Já a heterogeneidade no nível dos dados a serem transmitidos diz respeito às novas tecnologias disponíveis que permitem o transporte de imagem, voz e dados nas redes, cada um exigindo técnicas de gerência bem específicas para cada um desses tipos (HOLANDA FILHO, 1998).

O gerenciamento integrado de sistemas com vários fabricantes diferentes é

portanto, um problema a ser resolvido, pois se de um lado a aquisição de componentes de um único fabricante resolve o problema causado pela heterogeneidade, por outro lado causa o problema de dependência a este fabricante, isto é, a organização limita-se unicamente ao desenvolvimento tecnológico deste fabricante.

A questão da interoperabilidade entre diversos domínios organizacionais diz respeito ao fato de que cada organização possui necessidades e políticas de gerenciamento próprias. É devido a este problema, que os órgãos de pesquisa e normalização, como também os fabricantes de componentes de redes de computadores, procuram adotar estratégias e mecanismos padrões, com o intuito de contornar os problemas provenientes dessa complexidade a nível intra-domínio (heterogeneidade) e de inter-domínio (interoperabilidade) (LOPES; SAUVÉ; NICOLLETTI, 2003).

Todas estas questões mencionadas anteriormente, deverão ser tratadas com ênfase pela gerência de redes, pois é de suma importância para a mesma, que por sua vez, deverá associá-las às suas diferentes áreas funcionais.

2.2 PROTOCOLO SNMP

O gerenciamento de dispositivos em uma rede local se mostra como uma tarefa de importância cada vez maior dentro das organizações. Além disso, a quantidade de equipamentos que podem ser gerenciados tende a aumentar continuamente e, aliada a isto, está a necessidade de simplificar o processo de gerência. Assim, o protocolo SNMP pode ser usado para o gerenciamento dos dispositivos conectados a uma rede local de forma simples e direta (FARREL, 2005).

Por sua vez, os equipamentos tendem a oferecer cada vez mais possibilidades de gerenciamento, de modo a facilitar tarefas como a detecção de falhas,

a visualização de grandezas e as notificações de condições de exceção ou eventos. Portanto, cada vez mais equipamentos oferecem funcionalidades de gerenciamento por SNMP, o que tende a torná-los compatíveis com as redes locais e mais facilmente gerenciáveis (MAURO; SCHMIDT, 2001).

O protocolo SNMP define duas entidades para o gerenciamento, as quais trocam informações entre si por meio de requisições do tipo gerente-agente. O gerente SNMP realiza basicamente duas operações (FARREL, 2005):

a) leitura de valores para o monitoramento do dispositivo gerenciado;

b) escrita onde for possível efetuar a alteração de valores deste dispositivo;

O agente SNMP fica responsável por responder às solicitações do gerente e alterar as informações quando solicitada tal operação, além de notificar o gerente no caso de ocorrer alguma exceção (SANTOS, 2004).

Toda a inteligência do processo fica na estação de gerência permitindo que o agente seja uma aplicação muito simples e com o mínimo de interferência no dispositivo em que está sendo executado. As decisões tomadas na ocorrência de problemas e as funções de relatórios ficam sob responsabilidade do gerente (MAURO; SCHMIDT, 2001).

Tendo o protocolo SNMP devidamente instalado nas estações de gerenciáveis e na estação de gerência, basta agora desenvolver ou simplesmente utilizar os agentes e gerentes.

2.2.1 Agente SNMP

O agente é um processo executado na estação gerenciada, responsável pela manutenção de uma base de dados local com as informações de gerência dessa estação.

Cada estação gerenciada pelo SNMP deve possuir um agente e uma base de informações de gerência. Sendo assim, a estação gerenciada é vista como um conjunto de variáveis que representam informações referentes ao seu estado atual. Essas variáveis ficam disponíveis ao gerente por meio de consultas e podem ser alteradas por ele (STALLINGS, 1999).

Ao disponibilizar essas variáveis à leitura, a estação permite seu monitoramento e, ao receber novos valores do gerente, ela estará sendo controlada (FARREL, 2005).

O agente também é responsável por notificar o gerente no caso da ocorrência de alguma exceção na estação gerenciada. Essas estações gerenciadas podem apresentar falhas ou comportamentos inadequados, e quando o agente identifica que ocorreu um evento significativo ele envia pacotes informativos sobre o ocorrido a todas as estações de gerência de sua lista de distribuição de alarmes. Esta operação é efetuada por meio de interrupções e essas por sua vez, podem ou não informar exatamente os detalhes sobre o que ocorreu, podendo ser necessário que a estação de gerenciamento realize consultas para essa investigação e obtenção de mais detalhes (STALLINGS, 1999).

Há sistemas operacionais que possuem agentes SNMP prontos para serem utilizados, e desta forma pode-se desenvolver apenas gerentes com objetivos específicos que venham a satisfazer determinadas necessidades de gerenciamento.

2.2.2 Gerente SNMP

O gerente é uma aplicação em execução que localiza-se em uma estação de gerenciamento. É possível que exista um ou mais gerentes em execução numa mesma

estação, colaborando entre si para o gerenciamento, e consequentemente todos eles utilizam o protocolo de gerência disponibilizado por essa estação. Essas aplicações são capazes de monitorar os agentes por meio de requisições de informações contidas na base de informações de gerenciamento, e também são capazes de alterar as características das estações gerenciadas, informando novos valores ao agente (RIGANTI, 2005).

Os gerentes são os responsáveis pela implementação da política que será adotada na gerência e também são responsáveis pelo controle de acesso referente às pessoas ou entidades responsáveis pelo gerenciamento da estação. O envio de alarmes por e-mail, chamadas telefônicas, mensagens para telefones celulares ou outras formas de comunicação com o administrador são comuns nestas aplicações (STALLINGS, 1999).

Utilizando o gerente e o agente para obter os dados armazenados na MIB referentes os objetos de gerência, pode-se manipular estes dados conforme a necessidade de gerenciamento implementada no gerente.

2.2.3 Management Information Base

A MIB é a base de informações de gerenciamento. O agente é capaz de responder ao gerente consultas SNMP sobre o conjunto de informações contido na MIB. De fato, em geral é codificado um arquivo, chamado *arquivo de MIB*, no qual são relacionadas informações para que o gerente saiba quais são os dados que podem ser solicitados a um agente e também as informações de alerta que poderão ser enviadas do agente para o gerente (RIGANTI, 2005).

Constituída por uma estrutura em árvore contendo as variáveis de gerência

de um determinado equipamento, a MIB define para cada variável um identificador único denominado *Object Identifier* (OID), formado por um número inteiro não negativo. Em princípio, todos os objetos definidos em todos os padrões oficiais podem ser exclusivamente identificados. Para localizar uma determinada informação, o identificador da variável que será acessada pelo SNMP é representado com o endereço *Internet Protocol* (IP) do equipamento em conjunto com o identificador do objeto na árvore MIB (OID) (STALLINGS, 1999).

A Mib-2 pode ser representada pela Figura 1:

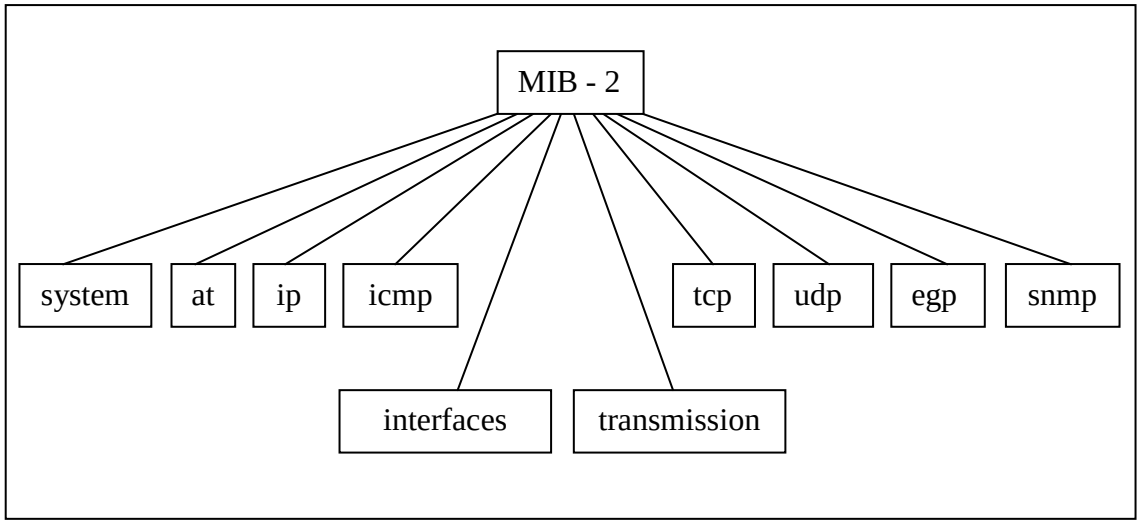

Figura 1. Mib-2
Fonte: STALLINGS, W. (1999)

Com a obtenção dessas informações específicas da MIB, pode-se desenvolver gerentes capazes de tratar particularidades de gerência por área funcional.

2.3 ÁREAS FUNCIONAIS DO GERENCIAMENTO DE REDES

Para especificar a tarefa de gerenciamento de redes de computadores, a *International Organization for Standardization* (ISO) fez uma separação funcional das necessidades no processo da mesma, com o objetivo de especificar as áreas de

gerenciamento, e desta forma facilitar a sua administração (SPECIALSKI, 2000).

Esta divisão foi aprovada e posteriormente adotada pela maioria dos fornecedores de sistemas de gerenciamento de redes. As áreas funcionais de gerência de redes foram divididas em cinco, apresentadas por (SANTOS, 2004):

a) falhas;

b) desempenho;

c) configuração;

d) contabilização;

e) segurança.

2.3.1 Falhas

Corresponde à área funcional que permite a detecção, o isolamento e a correção de operações anormais na rede, como por exemplo, falhas em dispositivos. Os recursos de gerenciamento de falhas mostram ao administrador de rede, o número, tipo, horas de ocorrência e localizações de erros na rede. Quando ocorrem falhas em uma rede, é importante que os seguintes procedimentos sejam seguidos: localizar a falha, isolar a mesma do restante da rede e reparar os componentes que apresentam problemas, de forma a restabelecer o funcionamento da rede ao seu estado normal (TANENBAUM, 2003).

O monitoramento de falhas tem como objetivo determinar a ocorrência de falhas da forma mais rápida possível e identificar as causas dessa falha. Uma vez identificada a causa, ações deverão ser tomadas para solucionar o problema.

Os seguintes problemas são associados à ocorrência de falhas (TANENBAUM, 1997):

a) **falhas não observadas**: algumas falhas são difíceis de serem observadas por meio da observação local. Já outras falhas, podem não serem observadas por causa da incapacidade ou impossibilidade do equipamento registrar a ocorrência da falha;

b) **observações inexatas**: quando observações detalhadas de falhas são possíveis de serem realizadas, podem existir incertezas ou inconsistências associadas a estas observações.

Depois que as falhas são observadas, é necessário que cada falha seja isolada. Para que estas falhas sejam isoladas, surgem alguns problemas, tais como (SANTOS, 2004):

a) **múltiplas fontes**: quando diferentes tecnologias de rede estão envolvidas, os locais e tipos de falhas tendem a aumentar significativamente. Isso dificulta ainda mais a localização da fonte geradora da falha. Informações transmitidas entre cliente e servidor passam por uma rede local, um roteador, um multiplexador, e um sistema de transmissão. Caso a conexão se perca ou a taxa de erros é elevada, o problema pode ter sido gerado em algum desses subsistemas, entretanto não sabe-se exatamente qual subsistema que gerou o problema;

b) **várias observações relacionadas**: ocorre quando uma falha no meio de transmissão afeta toda a comunicação entre redes de diferentes tipos. Com isso, esta falha pode gerar várias outras que por conseqüência estão relacionadas a esta;

c) **interferência de procedimentos de reparação local em diagnose**: procedimentos de correção local podem causar a perda de importantes evidências relacionada à natureza da falha, tornado impossível o

diagnóstico;

d) **ausência de ferramentas de teste automatizadas**: testes para isolar falhas são difíceis e complexos de serem realizados, por isso custam caro para o proprietário da rede.

Mediante os aspectos relatados anteriormente correspondente as falhas em redes, pode-se prever que haverá também uma certa complexidade de se realizar diagnósticos destas falhas, porém, existem técnicas que tornam o diagnóstico de falhas possível de ser realizado.

2.3.1.1 Diagnósticos de Falhas

A forma pela qual a falha é detectada consiste, geralmente, na comparação entre um comportamento esperado e o apresentado pela rede. As diferenças de comportamentos indicam que o sistema está com problemas. Confirmado o problema, deve-se determinar suas causas, ou seja, um diagnóstico inicial. Com isso, o objetivo principal deste diagnóstico é determinar os elementos responsáveis pela má conduta do sistema. A diferença entre o comportamento esperado e o observado é utilizada para guiar a pesquisa pelo diagnóstico mais exato possível. Existem dois tipos de diagnósticos: baseado em modelo e heurístico (TANENBAUM, 1997).

2.3.1.1.1 Diagnóstico Baseado em Modelo

O princípio básico consiste em uma interação entre o comportamento esperado para o sistema que está sendo diagnosticado e a observação do sistema no estado atual (TANENBAUM, 2003).

Este modelo permite definir o comportamento esperado para o sistema. As observações sobre o sistema informam como este sistema atualmente está se comportando que é também chamado de comportamento observado. Diferenças entre o comportamento observado e o esperado indicam que o sistema não está se comportando como esperado (SANTOS, 2004).

Um fator fundamental nos diagnósticos baseados em modelo deve-se ao fato de que os modelos devem ser completamente corretos. Caso contrário, o modelo adotado para ser utilizado como referência irá trabalhar com uma quantidade de hipóteses simplificadas e aproximações que não serão capazes de reproduzir a real situação do sistema implementado com precisão. Em um modo geral, se a aproximação for boa o suficiente, a abordagem baseada em modelo terá se mostrado como uma boa técnica de diagnóstico (HOLANDA FILHO, 1998).

A pesquisa em diagnósticos baseados em modelos tem trabalhado em dois segmentos (SANTOS, 2004):

a) **modelo de comportamento correto**: define como normalmente trabalha o sistema;
b) **modelo de comportamento falho**: define como o sistema trabalha se determinadas falhas ocorrem.

Resumidamente, o diagnóstico baseado em modelo segue os seguintes passos (SANTOS, 2004):

a) descreve o comportamento esperado de um sistema modelo;
b) observa um comportamento real de um sistema que está em conflito com o esperado (detecção das diferenças entre o sistema atual e o sistema modelo);
c) determina os componentes do sistema que em hipótese de falha explicam

tal diferença de comportamento (diagnóstico).

O diagnóstico baseado em modelo utiliza um formalismo apropriado para determinar o comportamento esperado do sistema de interesse. As formas de se fazer diagnósticos baseados em modelo, depende do conhecimento que se obtém do comportamento do sistema, e se classificam em (HOLANDA FILHO, 1998):

a) **diagnóstico baseado em consistência**: é baseado em um modelo que descreve o comportamento esperado do sistema;
b) **diagnóstico baseado em abdução**: é baseado em um modelo que descreve o comportamento falho do sistema.

2.3.1.1.2 Diagnóstico Heurístico

Em alguns problemas, a solução por meio de procedimentos exatos simplesmente não existe ou são inviáveis computacionalmente. Uma alternativa para se resolver esses tipos de problemas está na utilização de procedimentos que oferecem soluções consideradas boas, mas que em alguns casos pode não ser a melhor solução. A este método dá-se o nome de heurística (LOPES; SAUVÉ; NICOLLETTI, 2003).

Uma classe mais geral ao método de heurística é chamada de metaheurística e algumas têm sido propostas e especialmente projetadas para tentar evitar que o procedimento fique preso em situações de ótimos locais (HOLANDA FILHO, 1998).

O diagnóstico heurístico usa o conhecimento de especialistas e o conhecimento obtido por meio da observação de uma grande quantidade de dados. Geralmente, este conhecimento pode ser representado por regras, associando sintomas com as falhas observadas (SANTOS, 2004).

Muitos problemas podem ser identificados quando se usa uma abordagem

heurística (HOLANDA FILHO, 1998):

a) a aquisição do conhecimento de especialistas humanos é uma tarefa difícil e consome muito tempo, pois há a necessidade de se elaborar entrevistas ou um outro tipo de método;
b) o conhecimento é muito dependente de um ambiente específico e não é reutilizável, isto é, não há uma generalização para este conhecimento, pois o mesmo é específico;
c) a manutenção de uma grande base de regras é complexa;
d) apenas o conhecimento sobre o comportamento do sistema até a data atual pode ser utilizado e, portanto, alguns tipos de falhas raras podem não ser diagnosticados porque ocorreram depois da data de armazenamento do conhecimento.

Após ter sido realizado o diagnóstico do problema, é conveniente relatar a ocorrência do mesmo ao responsável que terá a missão de tratá-lo. Este aviso poderá ser efetuado por meio de uma notificação.

2.3.1.2 Registro de Alarmes

Notificações são mensagens emitidas por objetos gerenciados. Os alarmes fazem parte de um subconjunto de notificações e são gerados quando ocorrem condições não usuais. Eles podem ser gerados por condições anormais que foram detectadas, como por exemplo, quando ocorre uma degradação de um determinado serviço e conseqüentemente ultrapassa um determinado valor limite (FARREL, 2005).

Alarmes podem ser gerados por várias razões, e, para se isolar as fontes, os alarmes devem ser correlacionados. Desses alarmes correlacionados, a fonte da

condição de alarme deve ser identificada. Esses alarmes devem ser relacionados de uma forma padrão, e devem conter informações para identificar a natureza e a fonte do problema. Se alguns problemas ocorrem com alguma freqüência, informações adicionais também devem ser utilizadas para se analisar e estudar as tendências (TANENBAUM, 2003).

A função de registrar alarmes deve considerar principalmente as necessidades primordiais de serviços utilizados pelos usuários, os protocolos necessários para suportar tais serviços e também os parâmetros utilizados nos alarmes, para que desta forma, esses possam estar sendo empacotados no serviço de registro de alarme, sendo que o mesmo está contido nos agentes e no gerente (SANTOS, 2004).

Os alarmes são de fundamental importância para a determinação de problemas. As informações geradas pelos alarmes contêm não apenas uma ajuda para a determinação da fonte do problema, mas alguns desses alarmes podem indicar os passos para que o diagnóstico possa ser iniciado (TANENBAUM, 2003).

Notificações emitidas pelos objetos gerenciados devem ser manipuladas seletivamente para se escolher qual delas devem ser enviadas para um ou mais gerentes. Também a freqüência do envio de notificações para o gerente deve ser flexível (HOLANDA FILHO, 1998).

Há também a possibilidade de o próprio *software* gerente efetuar as notificações por meio da comparação dos valores de objetos gerenciados, filtrando somente o que lhe interessa.

2.3.1.3 Controle de Log

O controle de *logs* nada mais é do que requerimentos de usuários com

relação a serviços oferecidos e o protocolo suportado para rodar esses serviços. Já os eventos e as notificações que são recebidas pelo sistema deverão ser registradas no controle de *logs* para que a posteriori, possam ser utilizadas na análise de problemas (HOLANDA FILHO, 1998).

Ainda com relação aos *logs*, pode-se utilizar objetos gerenciados que emitem notificações por meio de algum tipo de procedimento, já que eles podem gerar registros de *log*. Esses registros deverão ser enviados para os arquivos de *log*, logo após serem tratados e filtrados corretamente por meio de filtros que possuem um conjunto de regras que estipulam e determinam quais registros de *log* serão armazenados (SANTOS, 2004).

Por meio dos filtros, pode-se obter notificações específicas para cada área funcional de gerenciamento.

2.3.2 Desempenho

Consiste em gerenciar constantemente os elementos que compõem a rede, para avaliar o seu comportamento mediante a sua operação. As informações tratada pelo gerenciamento de desempenho tem a função de servirem como base para planejamento e controle da qualidade dos serviços suportados na rede, sendo utilizadas por meio de estatísticas de desempenho. Consequentemente, estas estatísticas poderão ser utilizadas para promover ações em relação a prevenção de problemas que venham a ocorrer devido ao aumento dos tempos de resposta, que por sua vez são gerados por problemas ou por saturação de capacidade dos equipamentos ou serviços na rede (SPECIALSKI, 2000).

2.3.3 Configuração

Corresponde a um conjunto de variáveis que tratam das questões de instalação, inicialização, modificação e registro de parâmetros de configuração da rede, com a finalidade de prover uma rede eficiente, veloz e segura de acordo com suas limitações. É importante salientar que a gerência de rede deve conhecer amplamente os equipamentos que a compõe, como a localização, suas especificações técnicas e as configurações, para que no caso de apresentarem problemas ou serem trocados de lugar, os responsáveis pela gerência possam estar devidamente atualizados com relação a estas informações (TANENBAUM, 2003).

2.3.4 Contabilização

Consiste em realizar registros sobre informações relevantes a utilização dos recursos oferecidos pela rede, para que desta forma os responsáveis possam estar quantificando variáveis como a distribuição de custos, de tarifação, de planejamento de capacidade e verificação de cotas de utilização da mesma (SPECIALSKI, 2000).

2.3.5 Segurança

Com a mesma importância das demais áreas funcionais da gerência de redes, o gerenciamento de segurança se apresenta com a função de disponibilizar um conjunto de funções responsáveis pela criação de mecanismos de segurança para a proteção da rede. A proteção da rede de computadores de uma organização, por exemplo, consiste em limitar ou liberar usuários e aplicações para que acessem a mesma

de forma confiável e segura (SPECIALSKI, 2000).

Já as informações manipuladas por estes usuários e aplicações também deverão ser mantidas de forma segura, para que não sofram nenhum tipo de violação (HOLANDA FILHO, 1998).

2.4 SISTEMAS DE REGISTROS DE PROBLEMAS

O Sistema de Registro de Problemas (*Trouble Ticket System* - TTS), é utilizado para monitorar os problemas em uma rede de computadores, com o intuito de manter o rastro do ciclo de vida de um problema, ou seja, desde seu inicio até sua extinção. O TTS deve manter um completo histórico dos problemas ocorridos, e disponibilizá-lo de uma maneira pela qual, qualquer operador da rede possa analisá-lo e posteriormente realizar alguma iniciativa sem ter que consultar outro operador ou o administrador do sistema (SANTOS, 2004).

O TTS também pode ser utilizado como uma ferramenta de referência para a busca de soluções pelo sistema especialista, para o gerenciamento das falhas ocorridas em uma rede de computadores (MELCHIORS, 1999).

São atribuídos a estes sistemas, várias funções e características, tais como (SANTOS, 2004):

a) utilizar-se de um escalonamento de problemas atribuindo prioridades aos mesmos. Desta forma, os técnicos responsáveis poderão tomar decisões baseadas nas necessidades mais importantes do sistema. Além disso, poderia também permitir que a prioridade dos registros mudassem de acordo com o horário ou em relação à alarmes de tempo;
b) se o TTS for integrado a um sistema de correio eletrônico, permitirá que

alguns registros de problemas sejam enviados diretamente ao responsável;

c) atribuir temporizadores para cada registro de problema. Isto poderá ser aplicado caso o problema não seja resolvido em um determinado tempo, automaticamente será acionado um alarme lembrando sobre o problema. Também poderá ser utilizado um escalonamento baseado no tempo de espera, no tipo de rede e na importância do problema;

d) na hipótese de que a empresa opere em mais de um centro de operação, seria interessante enviar relatórios eletronicamente contendo resumos dos problemas associados a essa rede, para que os representantes de cada rede controlada pelo domínio de gerência, fiquem informados sobre o estado corrente de cada ocorrência ainda não solucionada;

e) fornecer meios para que se possa obter estatísticas como por exemplo, tempo médio entre a ocorrência da falha e tempo levado para corrigi-la. Com isso, pode-se analisar estas informações estatísticas de forma que se possa tomar medidas preventivas a possíveis falhas em dispositivos do sistema;

f) atuar como filtro dos alertas que estão relacionados a um registro de problema ainda não resolvido;

g) permitir que os usuários e administradores da rede possam visualizar as atividades desenvolvidas pelo centro de operações de gerência para a resolução de falhas, e posteriormente estarem cientes dos esforços empregados para a resolução de cada falha.

Outro propósito pelo qual um sistema de registro de problemas pode ser utilizado, é com o intuito de permitir uma interação entre os vários domínios envolvidos

em um problema. É importante ressaltar que a gerência de redes em um ambiente de processamento distribuído, admite o surgimento de ilhas de gerência, ou seja, cabe ao pessoal da rede local a responsabilidade pela administração da rede (MELCHIORS, 1999).

Se por um lado, a divisão da responsabilidade de administração facilita o diagnóstico dos problemas, isso porque os administradores locais possuem grande conhecimento daquele segmento da rede. Já por outro lado, a possibilidade de surgir problemas em sub-redes em função de anomalias de outra, leva à necessidade de se estabelecer algum mecanismo de apoio à interação e cooperação entre os responsáveis pelas diversas sub-redes. Desta maneira, os sistemas de registro de problemas poderão ser utilizados para compartilhar informações a respeito das soluções adotadas para a resolução dos diversos problemas, permitindo a colaboração dos especialistas das diversas sub-redes envolvidas no diagnóstico dos problemas (TANENBAUM, 2003).

Um TTS cria para cada problema informado um novo registro, atribuindo a este um número identificador que funciona como um índice, e registra os dados sobre o problema e ações realizadas no decorrer do mesmo, desde a sua criação até o seu encerramento. Os registros podem ser criados automaticamente, a partir de alarmes, ou manualmente, por usuários ou gerentes da rede. A partir do momento em que o problema é registrado, o TTS interage com sua base de dados de modo a preencher automaticamente as informações solicitadas pelo registro que ele tem condições de responder (SANTOS, 2004).

Todo problema que vir à ser registrado, deverá ser associado automaticamente ou manualmente pela pessoa responsável pela gerência à uma categoria de problemas, tais como: falha no enlace, falha em equipamento da rede, vulnerabilidade de segurança, erro de configuração, problema de performance e questão

de contabilização. Também podem existir outros diferentes tipos de registros para diferentes problemas encontrados em uma rede, variando o formato dos registros principalmente nos campos fixos. Com isso, esta metodologia poderá futuramente auxiliar na identificação dos problemas que ocorrem com mais freqüência (MELCHIORS, 1999).

O histórico dos problemas ocorridos pode ser armazenado por meio de campos fixos ou de texto em forma livre. Os campos fixos possuem a vantagem de serem utilizados com mais facilidade para a busca e ter sua consistência verificada com maior exatidão. Esta metodologia de armazenamento é apropriada para dados que são fornecidos pelo sistema automaticamente. Isso porque, possuem a vantagem de tornar os dados mais consistentes e confiáveis e seu uso é aconselhado para ambientes de resolução de problemas bem compreendidos e específicos. Em contrapartida, os campos fixos têm a desvantagem de forçar os usuários a escolherem entre valores preparados e permitidos que nem sempre representam a situação com precisão (SANTOS, 2004).

A estrutura de um registro de problema para redes de computadores, é formada de três partes: cabeçalho, atualizações e dados da resolução (TANENBAUM, 2003).

O cabeçalho é responsável pelas informações de abertura do problema, que incluem (MELCHIORS, 1999):

a) hora e data do início do problema;
b) identificação do usuário que abriu o registro;
c) severidade do problema;
d) descrição do problema;
e) quem relatou o problema;
f) quais os equipamentos envolvidos;

g) qual a rede envolvida (quando o Centro de Gerenciamento é responsável por várias redes);

h) endereço da máquina do usuário;

i) endereço da máquina destino;

j) próxima ação;

k) hora e data para o alarme associado ao problema;

l) para quem enviar o registro;

m) responsável pelo registro.

Neste cabeçalho, os quatro primeiros itens citados anteriormente são sugeridos para todos os sistemas. Já os itens restantes são específicos para o armazenamento de informações associadas aos diferentes tipos de problemas. Para permitir uma flexibilidade maior no sistema, um TTS pode ser desenvolvido para que apresente características chaves em forma de campo fixo e que, em determinados campos, permita uma maior flexibilidade ao usuário que está registrando o problema, dando a ele a possibilidade de redigir sobre o ocorrido (SANTOS, 2004).

As informações de atualização representam as ações e diagnósticos realizados ao longo do ciclo de vida do problema. A primeira atualização pode representar uma descrição do problema, pois quando o problema é aberto, sua natureza exata é geralmente desconhecida e a descrição fornecida pode ser imprecisa e complexa. Sugere-se que exista no mínimo um campo de texto livre nesse estágio do problema, para esse tipo de informação. Os demais campos poderão ser bem simples e armazenados tanto em campos fixos quanto em campos de texto livre. Ainda é necessário que haja sempre uma indicação da próxima ação associada ao registro, que novamente, poderá ser implementada como um campo fixo especial ou de texto livre (TANENBAUM, 2003).

Os dados da resolução dos problemas representam as informações que o resumem para análises estatísticas futuras e poderão também ser utilizados como um guia de referência para resolução de problemas similares. Os campos que são definidos como úteis para esta etapa são (MELCHIORS, 1999):

a) hora e data da resolução do problema;

b) duração;

c) uma linha descrevendo o ocorrido (para registro no relatório);

d) descrição da resolução do problema;

e) componentes afetados;

f) quem verificou o problema depois que este foi resolvido;

g) quem foi consultado para auxílio na resolução do problema;

h) estado corrente do problema;

i) usuários afetados;

j) prováveis causas do problema.

Os usuários de um TTS dependerão do nível de sofisticação do sistema de registro de problemas. Caso este sistema tiver um mecanismo de ajuda orientado por alguma técnica de Inteligência Artificial, boa parte dos registros poderão ser feitos automaticamente, e conseqüentemente, qualquer usuário, incluindo o usuário final, poderá ter seu trabalho facilitado pelo sistema (SANTOS, 2004).

Deve-se também levar em consideração, a utilização de mecanismos de segurança como por exemplo, a geração de *logs* e as senhas de acesso ao TTS, para que se possa ter um bom e correto funcionamento do mesmo. Se o TTS vir a ser de uma arquitetura de operação um pouco mais complexa, este provavelmente será utilizado somente pelas pessoas que possuem um conhecimento mais avançado do sistema. Sendo que este não é o objetivo, é importante que o TTS esteja disponível de forma

simples ao usuário final, porque assim diminui a burocracia na solução dos problemas e aumenta a usabilidade do mesmo (SANTOS, 2004).

Contudo, há outros métodos para tratamento de problemas, como por exemplo os sistemas especialistas para gerência de redes que fazem uso da técnica de raciocínio baseados em casos, da Inteligência Artificial.

3 SISTEMAS ESPECIALISTAS PARA GERÊNCIA DE REDES

Os sistemas especialistas são desenvolvidos para atender a uma determinada aplicação. São capazes de auxiliar na tomada de decisões, baseada em conhecimento justificado por especialistas da área, por meio de uma base de informações (RUSSELL; NORVING, 2004).

A partir do momento em que se compreende o funcionamento da gerência de redes de computadores, tem-se uma noção mais realista da complexidade e da funcionalidade de um sistema especialista para automação de redes, que deve implementar dois módulos que a compõe: a monitoração e o controle (MELCHIORS, 1999).

A tarefa de monitoração requer uma atenção especial, pois é por meio dela que os problemas serão detectados ou previstos. A monitoração consiste na coleta e na avaliação em tempo real dos dados coletados.

Geralmente existem três maneiras de se obter as informações. A primeira delas é a forma mais comum e também mais utilizada, onde as mesmas devem ser analisadas constantemente. Com isso, o dispositivo da rede envia os dados periodicamente para o monitor da rede, realizando *polling*[1]. A segunda maneira é quando não existe a necessidade de grandes informações de controle, ou seja, somente quando alguma exceção ocorre. Neste caso, o dispositivo da rede irá enviar os dados ao monitor somente em situações em que ocorrerem exceções, onde geralmente precisa ser efetuada alguma medida preventiva ou reparadora com uma certa urgência. Por último, existe a forma de obtenção dos dados em situações especiais, onde o próprio monitor da rede solicita ao dispositivo as informações necessárias (SANTOS, 2004).

[1] Solicitação de determinada informação feita pelo gerente ao agente, e esta por sua vez retorna ao gerente (STALLINGS, 1999).

Existem várias razões pela qual deve-se efetuar o controle, tais como (SANTOS, 2004):

a) reparar dispositivos que apresentam falhas;

b) reconfigurar a rede;

c) efetuar a manutenção de *software*;

d) realizar testes com novos *softwares* e *hardwares*.

Quando algum serviço da rede que está sendo monitorado apresentar um problema ou o monitor alertar para um eventual e provável problema, o gerente da rede precisa antes, localizar a falha de modo específico, para que depois disso possa realizar os devidos reparos. O gerenciador da rede precisa ter a capacidade de analisar e posteriormente fazer as reconfigurações necessárias. Isto quer dizer que, se necessário, o gerenciador deverá incluir inserção ou remoção de módulos de software e habilitação ou desabilitação de interfaces. Resumindo, o gerenciador da rede deverá ser capaz de habilitar a efetuar a manutenção da maior parte possível dos problemas ocorridos na rede.

O sistema de gerenciamento deve permitir que a sua operação possa ser realizada pelo gerente da rede por meio de dispositivos de entrada, como o teclado ou mouse, e desta forma o mesmo possa ter a opção de selecionar os itens de algum menu ou clicar em algum objeto gerenciado para poder alterá-lo sobre suas condições. Os dados obtidos anteriormente devem ser integrados juntamente com os novos dados para que o sistema foque as atenções aos itens que o gerente está tratando.

Para que a uma rede de computadores esteja livre de erros, ela deve conter a flexibilidade de *hardware*, redundância e funções de diagnósticos inteligentes. O sistema de gerenciamento da rede deve sempre estar monitorando o tráfego e os ajustes dos componentes da rede, para que desta forma ocorra um controle automatizado da

rede (TANENBAUM, 2003).

Os sistemas baseados em conhecimento podem e devem ser desenvolvidos de maneira independente, isto é, cada um deles deverá conter seu próprio conhecimento, e com isso permitir melhorias nos níveis de desempenho e disponibilidade do seu próprio conhecimento.

Os elementos citados são necessários para o entendimento das funcionalidades de cada módulo da arquitetura (HOLANDA FILHO, 1998):

a) **contadores**: armazenam o número de ocorrência de erros ou de outros eventos relevantes na rede, como exceções, por exemplo;
b) **eventos**: representam ocorrências significativas na rede;
c) **diagnóstico**: corresponde a um esquema que descreve um evento na rede, diagnosticado com mais detalhes;
d) **dado temporal**: constitui uma estrutura que representa eventos e diagnósticos que podem ser armazenados na base de dados temporal, ou seja, uma base de dados temporária;
e) **imagem da rede**: corresponde a um conjunto de esquemas que descreve a rede e seus elementos constituintes, como por exemplo os dispositivos que a compõe, tais como, pontes, roteadores, entre outros.

Os principais componentes da arquitetura dos Sistemas de Gerenciamento de Redes Baseados em Conhecimento podem ser apresentados conforme abaixo (HOLANDA FILHO, 1998):

a) **analisador de tendências:** identifica as anomalias de comportamento na operação da rede, que são indicadas pela alteração excessiva nos valores de contadores. Estes algoritmos utilizam métodos baseados em análises estatísticas que determinam quando há uma alteração no valor de um

contador dentro de um período já estabelecido ou tem uma alteração estatística, como por exemplo, variações repentinas ou de longa duração em valores de um contador. Se o resultado da análise estatística estiver dentro de uma faixa de valores aceitáveis, nenhuma ação precisa ser realizada, caso contrário, um esquema é criado para descrever as ocorrências do evento que causou esta alteração. Este evento é posteriormente encaminhado para a base de dados temporal;

b) **base de dados temporal** (BDT): corresponde à implementação da base central de dados do sistema e possui as informações sobre os intervalos de tempo. A BDT também tem uma função que corresponde à simulação, onde esta é usada para prever possíveis eventos que por ventura venham a ocorrer. Responsável também pelo processamento de consultas provenientes de outros componentes do sistema;

c) **base de conhecimento**: armazena as regras que especificam os eventos. Estas especificações podem ser o diagnóstico de um novo problema ou uma explicação de um evento baseado em diagnósticos anteriores. A única entrada para a base de conhecimento são os eventos vindos da BDT. Quando um novo evento é gerado, a base de conhecimento tenta explicar o evento que podem ser explicados pelo diagnóstico de um problema, e um esquema de diagnóstico é criado e encaminhado para a BDT;

d) **gerador de objetivos**: decide quais ações deverão ser tomadas para identificar eventos críticos. Monitora a BDT com o propósito de encontrar problemas que precisam ser solucionados. Essa atividade corresponde a conclusão de diagnósticos parciais, a determinação de

como se reconfigurar a rede após um evento crítico, a solução de falhas e a geração de relatórios para que o administrador da rede possa analisá-lo. Já os objetivos são identificados, colocados em fila de prioridade[2], e enviados para a BDT;

e) **planejador**: cria planos para ações a serem tomadas pelo SGRBC, como por exemplo, a reconfiguração da rede após uma falha e a localização e correção de problemas. O planejador busca os objetivos do gerador armazenados na BDT, e quando um novo objetivo é armazenado, o planejador gera um plano para tentar tornar o mesmo viável dentro de um intervalo de tempo específico. Após a geração deste plano, o mesmo é enviado para a BDT, onde é executado pelo executor;

f) **executor**: tem a finalidade de executar os planos gerados pelo planejador. Para isto, é necessário a geração das ações de gerenciamento a serem encaminhadas para o Sistema de Gerenciamento de Redes (SGR). Neste momento, há necessidade de um completo monitoramento do plano, para que desta forma possa se assegurar que ele funcione perfeitamente. O Executor monitora a BDT, aguardando que um plano esteja pronto para ser executado. Quando isto ocorre, o plano é lido na BDT e cada ação é enviada para o SGR para que esta seja executada.

Um Sistema de Gerenciamento de Redes Baseado em Conhecimento (SGRBC) pode ser representado conforme a Figura 2 (SANTOS, 2004):

[2] Qualidade do que vem em primeiro lugar (MICHAELIS, 1998);

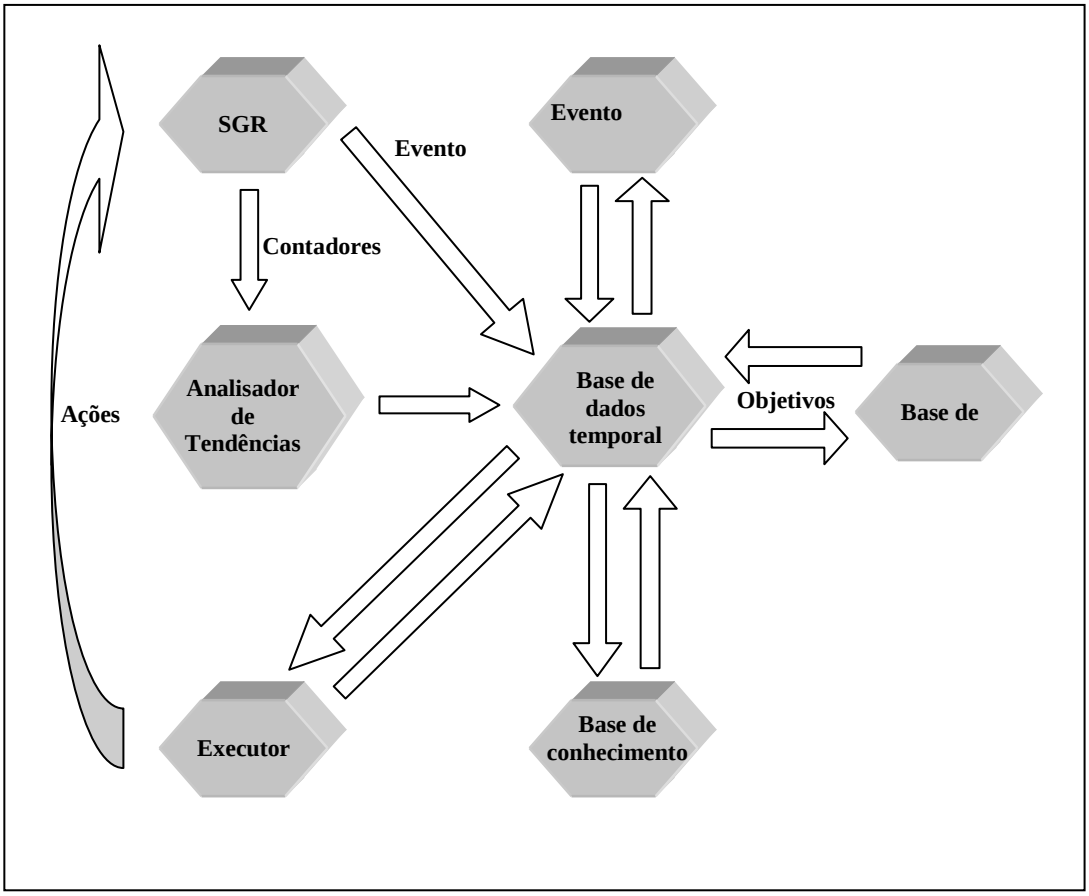

Figura 2. SGRBC
Fonte: HOLANDA FILHO, R. (1998)

Os sistemas especialistas para gerência de redes de computadores podem ser utilizados para diversas finalidades. Para a área de gerenciamento de configuração, os SGRBCs podem auxiliar no planejamento das redes. Para isso, é necessário que *a priori*, informe ao sistema as informações sobre a topologia, os componentes físicos, a lógica da rede e os mapas de roteamento (SANTOS, 2004).

Na área de gerenciamento de falhas os SGRBCs podem ser utilizados para diagnosticar eventuais problemas ou eventos e realizar a manutenção das redes. As ferramentas para diagnósticos efetuam a coleta de dados e análise das falhas da rede e seus impactos, com o objetivo de identificar as prováveis causas e estabelecer os devidos reparos para resolver o problema.

Os benefícios dos sistemas especialistas para diagnósticos incluem:

diminuição do tempo para detectar as causas do problema, sugerindo aos gerentes de redes ações para resolução do problema; e se possível automatizar a resolução dos mesmos pela intervenção direta, resultando em comandos corretivos para uma rede inteligente. Porém, o objetivo deste trabalho não é de forma alguma substituir o trabalho do gerente de redes, mas sim oferecer apoio ao gerenciamento da rede. Por isso, usa-se uma outra aplicação de sistemas especialistas para gerenciamento de falhas, que trabalha com o paradigma de controle da rede. Os benefícios deste tipo de aplicação são o aumento da precisão e eficiência da intervenção do operador, maior facilidade no momento de tomada de decisão e redução na quantidade de tempo necessária para restaurar ou alterar a rede (MELCHIORS, 1999).

Além do diagnóstico e controle, os SGRBCs podem ser aplicados para a interpretação de eventos, disponibilizando mensagens de acordo com a ordem ou a propriedade que são estabelecidos (SANTOS, 2004).

Já as áreas de gerenciamento de performance, contabilização e segurança também pode-se utilizar os benefícios de um SGRBC. Por exemplo, uma aplicação na área de gerenciamento de segurança pode se beneficiar do conhecimento obtido sobre o sistema alvo, o perfil da história das atividades dos usuários e heurísticas de detecção de intrusão, com a finalidade de detectar violações específicas que ocorrem no computador alvo (MELCHIORS, 1999).

Os sistemas especialistas baseados em conhecimento para gerenciamento de redes de computadores pode beneficiar-se com uma das técnicas de Inteligência Artificial, o raciocínio baseado em casos, que será visto no capítulo seguinte.

4 RACIOCÍNIO BASEADO EM CASOS

O Raciocínio Baseado em Casos (RBC) é uma área de conhecimento da Inteligência Artificial (IA) que se baseia em uma das formas de raciocínio humano, a memória. Consiste em uma metodologia de resolução de problemas que, em alguns aspectos, diferencia-se de outras técnicas da IA, pois ao invés de conter somente um conhecimento geral do domínio do problema ou fazer associações de relacionamento generalizadas entre descrição e conclusões de um caso, utiliza o conhecimento específico de uma experiência passada, para resolver uma situação atual. Assim, um novo problema é resolvido por meio da busca por um caso similar passado e a solução poderá ser adaptada para este (SILVA, 2000).

A diferença entre RBC e outras técnicas de IA, está na resolução de um problema que inclui processo de armazenamento do mesmo numa base de casos para que posteriormente, possa ser utilizado na solução de futuros problemas (REZENDE, 2005).

Existem quatro elementos básicos que compõem os sistemas de RBC (VON WANGENHEIM; VON WANGENHEIM, 2003) :

a) **representação do conhecimento**: nos sistemas de RBC o conhecimento pode ser representado de várias maneiras, como por exemplo, em forma de casos abstratos e generalizados, tipos de dados, modelos de objetos usados como informação, entre outros. Porém, na maioria desses sistemas, o conhecimento é representado por meio de casos que representam experiências concretas;
b) **medida de similaridade**: é a capacidade que se deve ter de encontrar um caso similar que está armazenado na base de casos para o problema atual

e também responder à pergunta quando um caso relembrado for igual ou parecido com um novo problema;

c) **adaptação:** na maioria das vezes, as situações anteriores representadas como casos serão iguais as do problema atual. Os sistemas de RBC mais avançados, possuem mecanismos e conhecimento suficientes para adaptar completamente os casos recuperados e posteriormente verificar se os mesmos satisfazem as características da situação atual;

d) **aprendizado**: corresponde à capacidade que um sistema de RBC possui de se manter atualizado e evoluir continuamente, pois quando o mesmo resolve problemas com sucesso, deverá armazená-los com as respectivas soluções, para que futuramente possa lembrar dessa situação e assim servir de base para resolver um novo caso que seja similar.

Como observa-se, o RBC é uma técnica que pode ser utilizada para manipular o conhecimento acerca de vários domínios específicos, basta que os represente corretamente.

4.1 REPRESENTAÇÃO DE CASOS

A forma como se representa o conhecimento em um sistema de RBC é de fundamental importância, pois a principal maneira de representá-lo é por meio de *frames*[3], redes semânticas e outros (SILVA, 2000). Um caso corresponde a um objeto de conhecimento tratado de forma contextualizada, que por sua vez armazena um problema ou situação problemática que foi resolvido totalmente ou parcialmente (PAL; SHIU, 2004).

[3] Técnica da inteligência artificial utilizada para representar casos (REZENDE, 2005).

Normalmente, um caso representa a descrição de um problema juntamente com a solução adquirida durante a sua resolução, isto é, corresponde a uma associação entre dois conjuntos de informações conhecidas como a descrição do problema e a sua solução (MELCHIORS, 1999).

Os casos na maioria das vezes são utilizados para representar experiências concretas vivenciadas por alguém, mas também podem representar mesmo que raramente algo abstrato, isto é, quando há a possibilidade de algo vir a se concretizar, porém não tendo certeza que irá ocorrer.

4.1.1 Casos que Representam Experiências Concretas

A experiência concreta representa casos verídicos, que podem por exemplo, apresentar as seguintes situações (VON WANGENHEIM; VON WANGENHEIM, 2003):

a) um conjunto dos sintomas de um paciente e os procedimentos adequados para o respectivo tratamento médico a ser aplicado;
b) a descrição das falhas técnicas apresentada por um equipamento qualquer, como por exemplo, um microcomputador e os passos corretos a serem aplicados para o seu conserto;
c) as deficiências técnicas de um atleta e o treinamento adequado para corrigi-las;
d) os requisitos para a construção de uma obra e o esboço gráfico da planta para futura construção.

Um caso também pode constituir outros itens, como por exemplo, os resultados da aplicação de uma solução, a própria justificativa ou até mesmo a sua

explicação. Também pode-se acrescentar ao caso informações de origem administrativa, como o número do caso, a data de sua criação, o nome da pessoa responsável que o cadastrou na base de conhecimento, entre outras informações (LEAKE, 1996).

As maiorias dos casos constituem-se essencialmente de experiências concretas, vivenciadas em uma situação específica. Apesar disso, também pode-se criar casos abstratos, que representam situações que ainda não se concretizaram, mas que posteriormente possam vir a se realizar baseando-se em experiências adquiridas por um conjunto de situações observadas anteriormente (VON WANGENHEIM; VON WANGENHEIM, 2003).

Porém, não adianta somente adquirir e processar os casos, é necessário que estes sejam armazenados de alguma forma para serem reutilizados futuramente.

4.1.2 Armazenamento dos Casos

A disponibilidade dos casos para serem reutilizados se concretiza a partir do momento que os mesmos estejam organizados e armazenados em uma base de casos (BC). Geralmente, possui experiências que deram certo e descrevem métodos de solução que ajudaram na resolução do problema descrito, de forma que possam ser reutilizados futuramente. Já as experiências que não deram certo, representando tentativas frustradas de solução de um problema, também podem ser armazenadas na BC com o propósito de indicar problemas futuros, e conseqüentemente prevenir a reincidência de erros já ocorridos no passado (REZENDE, 2005).

Muitas vezes, também é necessário armazenar não somente os casos, mas o conhecimento em um âmbito mais generalizado acerca do mesmo, que de algum modo possa contribuir para a resolução de futuros problemas.

4.1.3 Repositórios de Conhecimento

Juntamente com os casos, um sistema de RBC também pode incluir o conhecimento geral sobre o seu domínio de aplicação. Existem quatro repositórios diferentes, onde um sistema de RBC poderá armazenar o conhecimento acerca do domínio de aplicação, chamados de repositórios de conhecimento, são eles (VON WANGENHEIM; VON WANGENHEIM, 2003):

a) **vocabulário**: é utilizado para descrever o conhecimento geral sobre o domínio de aplicação, cuja utilização se faz necessária durante os diferentes estágios do processo de RBC. Por exemplo, um sistema de ajuda via Internet de um fabricante de monitores, onde o cliente poderá pesquisar somente por questões já formuladas, necessitando ter um vocabulário técnico, com termos acerca do domínio de aplicação, como por exemplo, as palavras v*ídeo*, *configuração*, *resolução*, entre outros;

b) **casos**: são experiências concretas vividas no passado e armazenadas em uma base de casos, conforme citado anteriormente. Representam, por exemplo, um conjunto de perguntas feitas por telefone ou e-mail nos últimos três meses e as respectivas respostas dadas pelos técnicos da empresa;

c) **conhecimento sobre como identificar casos**: é utilizado na identificação de casos que podem ser úteis para resolução do problema atual por meio da similaridade entre as descrições dos mesmos e os casos armazenados na BC. Pode-se citar como exemplo, as perguntas de clientes que anteriormente foram relacionadas ao mesmo modelo de monitor, sendo que há um grande número de palavras-chave iguais. O

nível da similaridade entre dois casos é relevante, pois depende do domínio acerca da aplicação. Neste caso, pode-se afirmar que a similaridade deverá ser modelada de forma explícita em um sistema de RBC;

d) **conhecimento sobre como adaptar casos recuperados**: serve para adaptar os casos recuperados de forma a satisfazer completamente as necessidades da atual situação. Por exemplo, o problema em questão corresponde ao mau funcionamento do botão ligar de um modelo “X”, enquanto o caso mais similar refere-se a um problema de mau funcionamento do botão ligar de um modelo “Y”. Neste caso, a solução adotada no passado (trocar o botão de ligar do monitor “Y”) deverá ser adaptada à situação atual, sugerindo trocar o botão de ligar do monitor “X”.

Dependendo das características em que o sistema de RBC será aplicado, o foco principal da representação do conhecimento poderá variar de um repositório de conhecimento para outro.

Após o armazenamento dos casos e do conhecimento, há necessidade de tratar estas informações de modo que possam vir a ser reutilizadas, sendo necessário comparar estas informações (casos) com outras, baseando-se no quão parecidas elas são, ou seja, avaliando seu nível de similaridade.

4.2 SIMILARIDADE DE CASOS

Paralelamente a análise e caracterização empírica dos critérios de similaridade, os modelos formais ocupam espaço nos sistemas de RBC, pois estes

tentam aproximar o processo da determinação da similaridade com relação ao realizado pelo ser humano. Desta maneira, as suposições determinadas nestes podem ser verificadas de forma empírica, sendo que há uma série de aspectos básicos referentes ao julgamento de similaridade (BITTENCOURT, 2006).

Pode-se observar que há uma definição genérica para o conceito de similaridade, mas no que se refere ao conceito da determinação de similaridade, não se pode aplicar uma definição genérica, pois os casos são específicos. Isso ocorre devido aos vários aspectos que necessitam ser levados em consideração juntamente com a incerteza que está associada aos casos. Conseqüentemente, não se pode considerar que o processo de seleção de casos em um sistema de RBC é realizado de forma correta ou errada, mas sim avaliá-lo como melhor ou pior no que se refere ao domínio de aplicação específica.

4.2.1 Similaridade Global

A similaridade entre dois objetos caracteriza-se pela comparação e avaliação do quão estes são parecidos, em RBC dá-se o nome de similaridade global. Para que se possa determinar a utilidade de um caso em relação a outro, esta similaridade entre ambos deverá ser determinada (PAL; SHIU, 2004).

A similaridade global pode ser calculada por meio de diferentes técnicas, tais como (VON WANGENHEIM; VON WANGENHEIM, 2003):

a) *nearest neighbour*: conhecido também como Vizinho mais Próximo, é uma técnica bem simples de ser aplicada. Nesta técnica pode-se representar os casos como pontos distribuídos em um espaço geométrico e visualizar a distância entre eles, e quanto mais próximo um ponto

estiver de outro, maior a similaridade entre os mesmos é maior. Pode ser representado pela seguinte função matemática:

$S_1 = X_1 + Y_1 + ... + Z_1.$ (1)

$S_2 = X_2 + Y_2 + ... + Z_2.$

Onde:

- S é o valor da similaridade;

- X, Y e Z são os valores de atributos diferentes.

b) *nearest neighbour* ponderado: é uma técnica muito parecida com o *Nearest Neighbour,* porém, cada atributo pode ter uma importância diferente em relação aos demais para o cálculo da similaridade. Representa-se matematicamente pelo seguinte função:

$S_1 = ((X_1 * V_1) + (Y_1 * V_2) + ... + (Z_1 * V_3)) / (V_1 + V_2 + V_3).$ (2)

$S_2 = ((X_2 * V_1) + (Y_2 * V_2) + ... + (Z_2 * V_3)) / (V_1 + V_2 + V_3).$

Onde:

- S é o valor da similaridade;

- X, Y e Z são os valores de atributos diferentes;

- V é o grau de importância para cada atributo.

Há também outras técnicas similares que podem ser utilizadas, como por exemplo a distância euclidiana[4], a distância euclidiana ponderada, a métrica do quarteirão ou distância de Manhattan[5], a distância Hamming[6], entre outras.

Dentre as várias técnicas citadas anteriormente, nota-se que apenas duas foram detalhadas. Isso se deve ao fato de que *Nearest Neighbour* representa a técnica

[4] Método decorrente do Teorema de Pitágoras da Geometria Euclidiana utilizado para realizar o cálculo da similaridade entre casos no RBC (VON WANGENHEIM; VON WANGENHEIM, 2003).
[5] Conhecido também como Métrica do Quarteirão, é um método utilizado para realizar o cálculo da similaridade entre casos no RBC (VON WANGENHEIM; VON WANGENHEIM, 2003).
[6] Método utilizado para encontrar a dissimilaridade entre casos no RBC (VON WANGENHEIM; VON WANGENHEIM, 2003).

que deu origem à *Nearest Neighbour* Ponderada, e esta por sua vez foi utilizada para o cálculo da similaridade global nesta pesquisa.

4.2.2 Similaridade Local

Busca determinar o grau de similaridade entre o caso a ser resolvido e os armazenados na base de casos, observando-se que a similaridade local entre atributos específicos de ambos podem ser considerados no momento em que se processa a similaridade global. Conseqüentemente, quando houver casos com valores de atributo diferentes, mas que podem ainda ser similares ao procurado, não serão distinguidos dos outros, cujos valores são completamente diferentes (VON WANGENHEIM; VON WANGENHEIM, 2003).

A similaridade local pode ser calculada por meio de diferentes técnicas, tais como (VON WANGENHEIM; VON WANGENHEIM, 2003):

a) função escada: fundamenta-se no princípio de que o valor de um atributo é totalmente igual ou diferente ao mesmo de um outro caso. Esta técnica utiliza os limiares 1.0 para representar a igualdade total ou 0.0 para a desigualdade. Sendo assim, não existem valores intermediários, o que caracteriza uma função binária;
b) contagem de palavras: consiste em contar o número de palavras idênticas entre dois casos, sendo representada pela função:

$$S = B / A. \quad (3)$$

Onde:

- A é o número de palavras do caso que possui mais palavras;
- B é o número de palavras idênticas entre os dois casos;

- S é o valor da similaridade local.

Desta forma, se todas as palavras de ambos os casos forem idênticas, o valor de S será 1.0. Se todas forem diferentes, S será 0.0, mas se existirem algumas palavras iguais, diferentemente da função escada, S irá assumir o valor intermediário entre os limiares 1.0 e 0.0.

Como pôde-se observar, deve-se considerar a questão da similaridade local, pois permite a integração das similaridades entre atributos isolados ao cálculo da similaridade global, tornando o sistema de RBC mais sensitivo, além de auxiliar na recuperação dos casos mais similares ao atual.

4.3 RECUPERAÇÃO DE CASOS

O objetivo desta etapa é recuperar os casos da base que contenham uma solução mais próxima para um problema atual. A recuperação é feita usando as características do novo caso que são relevantes na solução de um problema. A tarefa de recuperação inicia com a descrição de um problema e termina quando um caso mais similar é encontrado (BRAGA JÚNIOR, 2001).

Para determinar se um caso é útil para a solução de um problema ou uma situação específica, é necessário que se encontre sua similaridade na resolução com o caso já resolvido anteriormente, para constatar se os mesmos diferem de acordo com o domínio e o propósito da aplicação. Pode-se dizer então, que a semelhança entre casos está na similaridade das características que representam o conteúdo e o contexto das experiências em questão. Um caso pode ser considerado similar ao problema atual, se a solução do caso pode ser reutilizada para resolver o problema atual. Em RBC assume-se que problemas ou situações similares requerem soluções similares (VON

WANGENHEIM; VON WANGENHEIM, 2003).

A partir da definição do conceito de similaridade e a forma de determinação da mesma, é necessário avaliar se o caso escolhido para a solução do problema atual é útil o suficiente. A solução descrita em um caso escolhido pode ser aceita para a resolução do problema atual, caso ela satisfaça os seguintes requisitos (PAL; SHIU, 2004):

a) permita a solução do problema atual de alguma forma;
b) evite que ocorra a repetição de um erro passado;
c) permita que se realize uma solução eficiente do problema, sem que haja a utilização de uma heurística de passo a passo para calcular uma solução;
d) ofereça a melhor solução para o problema conforme os critérios estabelecidos no sistema de RBC;
e) forneça uma solução ao usuário, sendo que utilize uma lógica que possa ser compreendida por ele.

A partir do momento que se pode associar idéias diferentes utilizando o próprio conceito de utilidade, ou seja, verificar se um caso da base de casos é útil para a resolução do problema atual, se faz necessário definir primeiro a maneira com que os objetivos do processo de escolha de casos irão interagir no sistema de RBC (LEAKE, 1996). No momento de se determinar a similaridade existe uma dependência dos objetivos a serem atingidos pelo processo de solução do problema, e com isso, essa dependência mostra-se fundamental na hora de avaliar os tipos de características de um caso relevante ao objetivo da aplicação (VON WANGENHEIM; VON WANGENHEIM, 2003).

Resumindo, a etapa de recuperação consiste em localizar na base os casos mais similares ao atual, para serem reutilizados na solução do mesmo. Porém, é

necessário utilizar um critério de indexação eficiente entre os casos da base, pois se a mesma estender-se em grandes proporções, o processo de recuperação pode ser prejudicado em relação ao tempo de resposta.

4.4 INDEXAÇÃO

A tarefa de encontrar casos similares na base de casos do sistema de RBC para um problema a ser resolvido, consiste em definir quais atributos serão utilizados para realizar a comparação entre um caso antigo e a situação atual a ser resolvida. A estes atributos de indexação dá-se o nome de índices (LEE, 1998).

Os índices podem ser determinados por meio de métodos manuais, onde a escolha começa com a análise dos casos para a identificação da utilidade que poderia ter o mesmo, e sob que circunstâncias. Essas informações devem ser, então, traduzidas para representações que o sistema pode usar, definindo um conjunto de descritores, que são trabalhados de modo a garantir que os índices sejam aplicáveis em âmbito geral e que possam ser reconhecidos no máximo de situações possíveis. Além dos métodos manuais, existem, também, métodos de indexação automáticos, que são apresentados por muitas das ferramentas de RBC disponíveis no mercado (SANTOS, 2004).

Na definição dos índices a difícil tarefa é prever que tipos de situação de consultas irão surgir e que tipos de informação serão necessárias para recuperar casos em situações futuras. Muitos esforços foram feitos para estabelecer regras gerais de vocabulário de índices em classes particulares de RBC, mas esta acaba ainda sendo desenvolvida para atender os objetivos específicos da recuperação de cada aplicativo que use RBC (MELCHIORS, 1999).

O processo de indexação é uma oportunidade de superar a deficiência de

experiências mal descritas e torná-las úteis e valiosas na realização da tarefa do sistema. Esta meta é conduzida pela correta interpretação da experiência a partir da perspectiva do especialista, permitindo a identificação do significado e da correlação entre as entidades ativas participantes na experiência. Uma forma de buscar tais relações é tentar representar as correspondências entre as causas e conseqüências, razões e soluções (PAL; SHIU, 2004).

A indexação pode representar um afunilamento no desenvolvimento de sistemas de RBC (BARRETO, 2001), como é o exemplo da necessidade de indexação automática para viabilizar o sistema. Esta indexação pode ser necessária em sistemas que comportam mecanismos de aprendizagem automática. Além disso, há domínios em que o conhecimento está disponível somente em formato textual o que exige grandes bases de casos. Os domínios do Direito, Economia e Medicina são exemplos nos quais um mecanismo de indexação automática é plenamente justificado (MELCHIORS, 1999).

A atribuição de índices aos casos depende da compreensão do conteúdo e das informações que um caso pode fornecer. Deve permitir que sejam reconhecidas as similaridades entre a situação corrente e os casos armazenados que podem contribuir para atingir os objetivos do caso corrente. Assim, para a escolha de um bom índice, é necessária a compreensão da situação problema, de forma que essas similaridades possam ser corretamente identificadas (BARRETO, 2001).

Os índices devem ser preditivos, ou seja, podendo prever a utilização da informação presente nos casos para diferentes situações de problema. Por isso, os índices devem (VON WANGENHEIM; VON WANGENHEIM, 2003):

a) endereçar os propósitos em que o caso pode ser usado;

b) ser suficientemente abstratos para permitir que um caso seja útil em uma

variedade de diferentes situações;

c) devem ser suficientemente concretos para que possam ser facilmente reconhecidos em situações futuras.

A indexação de um caso nada mais é do que os seus atributos de entrada devidamente valorados, onde por meio do cálculo de similaridade poderão ser recuperados para serem reutilizados ou adaptados para a solução de um caso futuro.

4.5 REUTILIZAÇÃO E ADAPTAÇÃO DE CASOS

Quando um caso adequado é recuperado da base, a solução sugerida é objeto de uma tentativa de reutilização para a solução do problema a ser resolvido. Neste momento, acontece uma reutilização de conhecimento por meio da transferência do caso anterior que foi recuperado para o atual (VON WANGENHEIM; VON WANGENHEIM, 2003).

4.5.1 Reutilização

A reutilização consiste principalmente da adaptação da solução do caso anterior ao atual, que necessita ser resolvido (COSTA; SIMÕES, 2004). É responsável pela recuperação de casos similares, a partir da descrição de um problema ou situação de casos similares ao problema corrente que seja útil para a identificação da sua solução (BRAGA JÚNIOR, 2001).

A busca pelos casos similares não deve considerar, porém, apenas a descoberta de algumas dimensões da descrição do problema similares à situação. Na identificação da similaridade entre os casos, alguns atributos são mais importantes que

outros no julgamento da mesma e esta valoração pode variar de acordo com os objetivos do sistema de RBC (PAL; SHIU, 2004).

Assim, a recuperação deverá considerar que os casos similares ao problema corrente são aqueles que são parecidos nas dimensões que auxiliam o sistema a realizar suas tarefas ou atingir os objetivos desejados (BRAGA JÚNIOR, 2001).

A recuperação dos casos úteis à situação corrente envolve várias etapas, cada uma possuindo diferentes pontos que devem ser analisados. Inicialmente, precisam ser identificadas quais as características ou dimensões do caso corrente que devem ser utilizadas para julgar a similaridade dos casos armazenados. Isso é determinado levando-se em conta os propósitos para os quais os casos estão sendo recuperados e as dimensões que foram relevantes no passado para determinar o resultado do ambiente para as soluções aplicadas.

Após a reutilização dos casos é necessário passar por um processo de adaptação dos mesmos, caso haja necessidade..

4.5.2 Adaptação

Há situações no RBC em que os casos recuperados podem apresentar uma solução aproximada para a atual, exigindo algumas modificações para melhor ajustá-la na resolução deste. Estas modificações são chamadas de adaptação e podem ser feitas por meio da utilização de conceitos específicos da técnica de RBC, utilizando regras que representam um conhecimento adicional sobre o domínio do problema ou até mesmo por meio de interações com o usuário (SILVA, 2000).

Esta etapa do RBC possui um papel fundamental na flexibilidade deste sistema, ou seja, a capacidade de resolver novos problemas adaptando os casos

recuperados de acordo com as novas circunstâncias do caso a ser resolvido. A maior dificuldade surge no momento em que define-se como realizar a adaptação. Há muitas maneiras, porém se depende do conhecimento sobre as possíveis modificações válidas, isto é, obter maneiras de selecionar quais serão as modificações mais apropriadas para a situação atual (RAMOS, 2000).

Resumindo, a solução proposta é adaptada de maneira a satisfazer por completo os requisitos da situação atual. Além disso, é necessário verificar se a solução adaptada é satisfatória ou não, por meio do processo de revisão.

4.6 REVISÃO

Quando a solução sugerida pela fase de reutilização está incorreta, existe uma oportunidade para que o sistema de RBC possa aprender com as falhas encontradas na mesma (RAMOS, 2000). A esta fase do RBC dá-se o nome de revisão do caso. (BITTENCOURT, 2006).

O processo de revisão pode ser compreendido pela Figura 3:

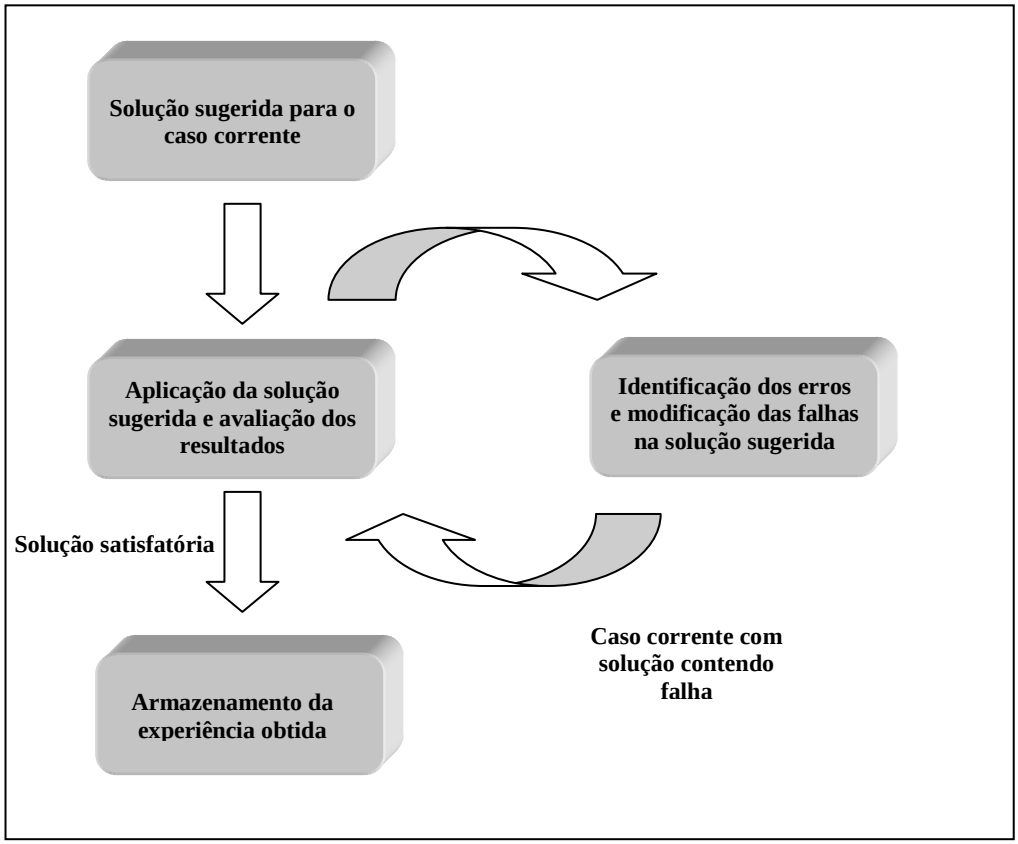

Figura 3. Processo de Revisão
Fonte: AAMODT, A.; PLAZA, E. (1994)

A revisão consiste nas seguintes tarefas (VON WANGENHEIM; VON WANGENHEIM, 2003):

a) identificação da solução sugerida;

b) avaliação da solução gerada para ser reutilizada;

c) correção da solução utilizando o conhecimento específico acerca do domínio.

d) armazenamento da nova solução.

Conforme visto anteriormente, o processo de revisão tem o objetivo de avaliar se a adaptação do caso recuperado atende ao atual, verificando a solução e os erros.

4.6.1 Avaliação da Solução

A tarefa de avaliação da solução leva em consideração o resultado da solução em uma aplicação de ambiente real, por meio da consulta a um especialista ou da aplicação de regras que possam validar a solução sugerida anteriormente.

4.6.2 Reparação de Falhas

A função de reparar um caso consiste em encontrar os erros da solução proposta e apresentar as explicações para a atualização desta solução sugerida. Alguns sistemas utilizam um conhecimento casual para gerar uma explicação do por que certas partes da solução não foram alcançadas. Estes sistemas aprendem as situações gerais que causaram as falhas usando uma técnica de aprendizagem baseada em explicação. Isto é incluído na memória de falhas que é empregada na fase de reutilização, fornecendo atalhos para a adaptação, oferecendo a vantagem de se detectar possíveis erros na fase de adaptação (PAL; SHIU, 2004).

Descobrindo as falhas da solução proposta, tem-se a necessidade de aplicar a reparação. Esta tarefa usa uma falha explicada para modificar a solução de tal forma que a mesma não ocorra. Para modificá-la, um módulo de reparação, que possui conhecimento do domínio, assegura que a causa dos erros não irá ocorrer. Após este processo a solução poderá ser retida, se houver certeza da eficiência da mesma. Caso contrário poderá passar novamente pelas etapas de validação e reparação (AAMODT; PLAZA, 1994).

4.7 RETENÇÃO DE NOVOS CASOS

Depois de realizado o processo de revisão, a solução do caso selecionado pode então ser utilizada para resolver o caso de entrada. Um sistema de RBC somente se torna eficiente quando estiver preparado para aprender a partir das experiências passadas e da correta indexação dos problemas (KOLODNER, 1993).

A retenção de casos significa incorporar à base, informações úteis relativas à resolução de um novo problema. Este processo corresponde à aprendizagem de um sistema de RBC, que é solicitado pelas tarefas de avaliação e adaptação de soluções (COSTA; SIMÕES, 2004).

O aprendizado de um sistema deve ocorrer de forma ordenada para não tornar a base de casos algo difícil de ser manipulado. A inclusão de novos casos e associação de índices deve ocorrer de forma que o sistema possa raciocinar sobre eles. Melhorias no cálculo do grau de similaridade assim como nas regras de adaptação, também podem ajudar na melhora da performance do sistema (PERES, 1999).

A aprendizagem em sistemas de RBC pode ser empregada à base de casos. Estas bases podem ser estendidas por processos incrementais de aprendizagem se a tarefa e o projeto do sistema permitirem. A partir de um pequeno conjunto de casos, a base pode crescer com novos elementos (LEAKE, 1996).

O aumento da base de casos acontece como expressão da aprendizagem com a experiência, assim a parte do caso destinada ao resultado do emprego de determinada solução ou interpretação serve a este propósito. Mantém-se no caso o registro de seu desempenho ao ser utilizado. Assim, tanto sucessos como fracassos são informados incrementando o conhecimento e as lições embutidas no caso. O registro do resultado de reutilização pode prevenir o usuário em relação às possíveis conseqüências de seu

uso. Este procedimento é valioso porque, para compensar a inclusão de informações no caso, o sistema evita a reutilização de sugestões menos favoráveis, resultando no incremento da qualidade da recuperação (LEE, 1998).

Resumidamente, o processo de RBC pode ser representado pela figura 4:

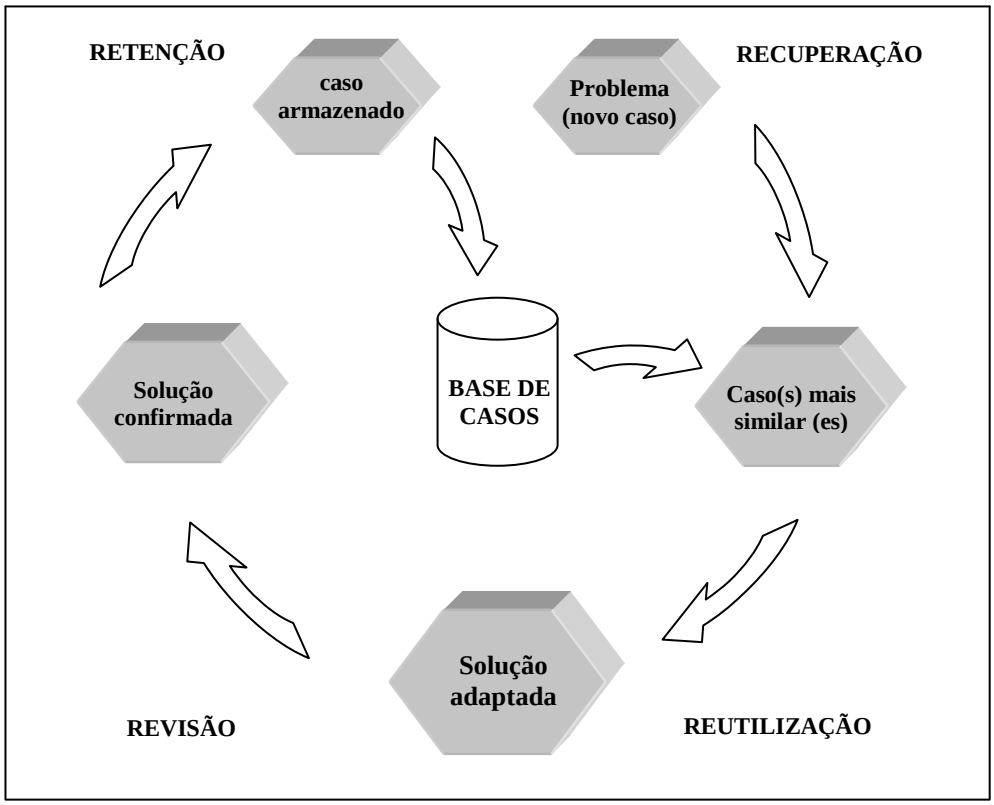

Figura 4. Ciclo de RBC
Fonte: VON WANGENHEIM, C;VON WANGENHEIM, A. (2003)

5 TRABALHOS CORRELATOS

Durante os estudos objetivando o desenvolvimento da pesquisa, foram estudados alguns trabalhos semelhantes, mas com outro enfoque. A seguir, estão descritos alguns trabalhos pesquisados.

5.1 SAGRES: UM SISTEMA BASEADO EM CONHECIMENTO PARA APOIO À GERÊNCIA DE FALHAS EM REDES DE COMPUTADORES

É uma Dissertação de Mestrado realizada na Universidade Federal do Ceará (UFC), no ano de 1998, pelo acadêmico (HOLANDA FILHO, 1998).

Consiste em um sistema baseado em conhecimento para apoio à gerência de falhas em redes de computadores, utilizando os conceitos de agentes SNMP e Sistemas Baseados em Conhecimento.

O SAGRES utiliza agentes e gerentes SNMP para automatizar a obtenção das informações referentes aos objetos gerenciados numa rede TCP/IP. Já o tratamento dessas informações é feito pela *shell* Nética, que aborda os conceitos de Sistemas Especialistas Probabilísticos (SEP).

5.2 RACIOCÍNIO BASEADO EM CASOS APLICADO AO GERENCIAMENTO DE FALHAS EM REDES DE COMPUTADORES

Consiste em uma Dissertação de Mestrado realizada na Universidade Federal do Rio Grande do Sul, no ano de 1999, pela acadêmica (MELCHIORS, 1999).

Neste trabalho foi desenvolvido um sistema denominado DUMBO, que

compreende uma aplicação baseada em conhecimento para apoio à gerencia de falhas em redes de computadores, utilizando também os conceitos de agentes SNMP, RBC e *Trouble Ticket System* (TTS), também conhecido como sistemas de registros de problemas.

O DUMBO utiliza agentes e gerentes SNMP para automatizar a obtenção das informações referentes aos objetos gerenciados numa rede TCP/IP. Já o tratamento dessas informações é feito por RBC em conjunto com um sistema de TTS.

5.3 SISTEMA DE GERENCIAMENTO DE REDES BASEADO EM CONHECIMENTO

É um trabalho de Pós-Graduação realizado na Universidade Federal de Lavras, no ano de 2004, pelo acadêmico (SANTOS, 2004).

Este trabalho foi desenvolvido baseado no SAGRES, porém foi desenvolvido um sistema de RBC baseado no método de indexação dos casos conhecido do Vizinho mais Próximo, para substituir a *shell* Nética utilizada no SAGRES. Porém, este trabalho é apenas uma modelagem.

Na Tabela 1 é realizado um breve comparativo entre os trabalhos correlatos, onde é informado alguns paradigmas utilizados em suas implementações.

Tabela 1. Comparativo dos trabalhos correlatos

Autor	**Ambiente**	**Linguagem**	**BD**	**Protocolo**	**IA**
(HOLANDA FILHO, 1998)	Local	Pascal	Arquivos	SNMP	SEP
(MELCHIORS, 1999)	Web	C / PHP	Postgres	SNMP	RBC
(SANTOS, 2004)	Apenas Modelagem do sistema Sagres			SNMP	RBC

Comparando os trabalhos correlatos a presente pesquisa, pode-se observar que há algumas diferenças, já que este protótipo foi desenvolvido em linguagem *Java*, o que possibilita ser executado em diferentes plataforma, além de utilizar um banco de dados eficiente e seguro para armazenar e realizar os cálculos de similaridade dos casos.

6 SISTEMA DE APOIO A GERÊNCIA DE FALHAS BASEADO EM CASOS - ANTIFAIL

Neste capítulo, será apresentada a parte prática da pesquisa, demonstrando a utilização dos métodos e o conhecimento a partir da teoria para o desenvolvimento de uma ferramenta capaz de auxiliar no gerenciamento de falhas em uma rede de computadores.

Dentre as ferramentas para que se pudesse prover a interface *desktop* na estação gerente foi utilizada a linguagem *Java*[7], por meio da plataforma de desenvolvimento *Netbeans* 5.5[8]. A linguagem *Java* foi utilizada devido ao fato de que seus aplicativos podem ser executados em diferentes plataformas operacionais, desde que estas possuam uma máquina virtual *java* instalada com uma versão adequada para interpretar o protótipo aqui desenvolvido. Outro fator determinante na escolha desta linguagem foi o fato de possuir distribuição livre. O *Netbeans* 5.5 foi escolhido pelos mesmos motivos da linguagem *Java* e também por permitir a manipulação de código *Java.*

O banco de dados escolhido foi o *OracleXE*[9] porque possui distribuição livre e permite desenvolver funções e procedimentos internos, o que torna a aplicação mais eficaz quanto ao seu desempenho. Para realizar a comunicação entre a aplicação e o banco de dados foi utilizado um *Java Database Connectivity* (JDBC)[10], porque permite trabalhar em conjunto com o banco de dados *OracleXE* para registrar as informações coletadas pelos agentes SNMP, além de possuir distribuição livre.

---

[7] Obitida no site: www.sun.com
[8] Obtido no site: www.sun.com
[9] Obtido no site: www.oracle.com
[10] Obitdo no site: www.sun.com

O desenvolvimento prático contém várias etapas, visando assim uma melhor organização para facilitar a conclusão desta pesquisa. Foi dividido da seguinte forma:

a) definição do problema a ser resolvido: a demora e a imprecisão por parte dos profissionais responsáveis pelo gerenciamento de redes de computadores no modelo TCP/IP no momento de solucionar problemas relacionados a ocorrência de falhas;
b) escolha das técnicas adotadas: para o gerenciamento da rede foi utilizado o protocolo SNMP, porque o mesmo foi criado para facilitar o gerenciamento de dispositivos de redes. Ele também possibilita em determinados casos, identificar com precisão a causa e o local onde ocorreu ou irá ocorrer uma falha. Neste momento pode-se resolver o problema de imprecisão. Para solucionar a demora na resolução das falhas ocorridas, utilizou-se a técnica de RBC, pois tem a capacidade de recuperar casos similares acontecidos no passado, para que possa servir de base na resolução de um atual;
c) levantamento das variáveis MIB utilizadas: foram escolhidas somente as variáveis que armazenam informações referentes as falhas ocorridas nos equipamentos, para que desta forma o sistema possa gerenciar somente a área funcional de falhas;
d) aquisição das ferramentas para o desenvolvimento do protótipo: foram utilizadas somente ferramentas com distribuição livre, e que possam funcionar em vários ambientes operacionais, onde o padrão de comunicação da rede seja o TCP/IP, e que possua também o protocolo SNMP com seus agentes devidamente funcionando;

e) a captura dos dados contidos na MIB: para capturar as informações específicas da MIB acerca do gerenciamento de falhas, é necessário que os dispositivos gerenciados possuam o protocolo SNMP com seus agentes. Com estes requisitos devidamente funcionando, é aplicada a técnica de *polling*, que consiste em uma solicitação de determinada informação feita pelo gerente ao agente, e esta por sua vez retorna ao gerente;

f) a aplicação do RBC: no desenvolvimento da aplicação de RBC, foram utilizados alguns métodos para o cálculo das medidas de similaridade local e global. Na similaridade local foram utilizados os métodos de contagem de palavras e função escada. Já na global, foi utilizado o método do *nearest neighbour* ponderado.

A seguir será descrito o modo como foi realizado a etapa de desenvolvimento do protótipo.

6.1 DIAGRAMAS DO SISTEMA

O sistema pode ser representado pelos diagramas a seguir, que ilustram o funcionamento da aplicação e do banco de dados respectivamente. Os diagramas foram gerados com a ferramenta *trial Pacestar Diagrammer UML*[11].

A Figura 5 representa o diagrama de atividade do padrão UML que representa funcionamento do protótipo desenvolvido, contemplando as áreas de gerenciamento de redes e inteligência artificial. Como pode-se observar, o sistema tem a capacidade de efetuar o gerenciamento dos equipamentos e aplicar a técnica de RBC

[11] Obtida no site: www.pacestar.com

simultaneamente, já que para isso foi desenvolvida utilizando *threds*[12].

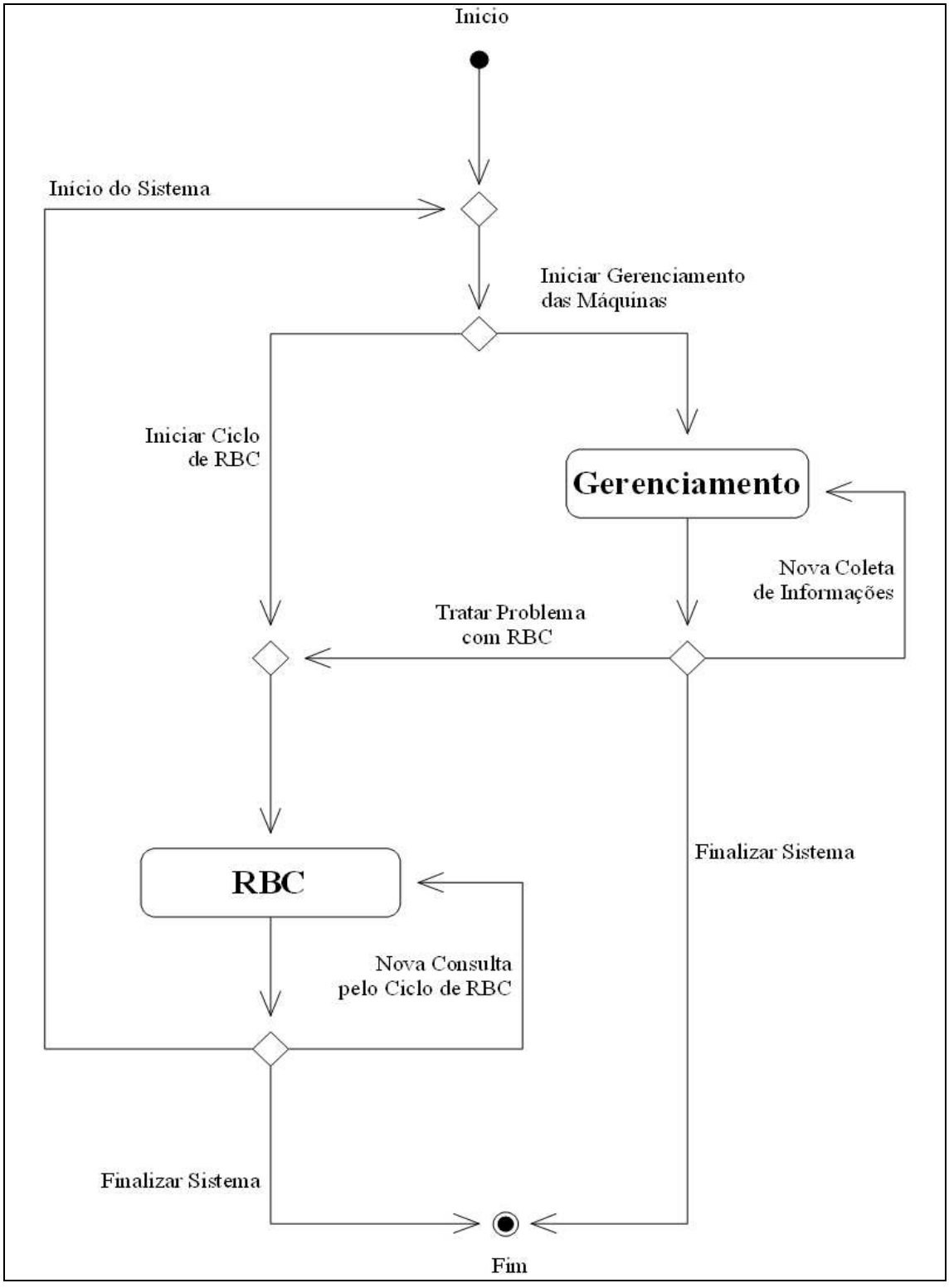

Figura 5. Diagrama de Atividade do ANTIFAIL

[12] São processos computacionais independentes

A Figura 6 representa o diagrama de entidade-relacionamento do protótipo, onde representa as tabelas criadas no banco de dados *OracleXE*, contendo também todas as relações de dependência, como as chaves primárias e estrangeiras. Com a criação desta estrutura relacional, pode-se então manipular os casos por meio do RBC, formando assim a base de conhecimento do sistema.

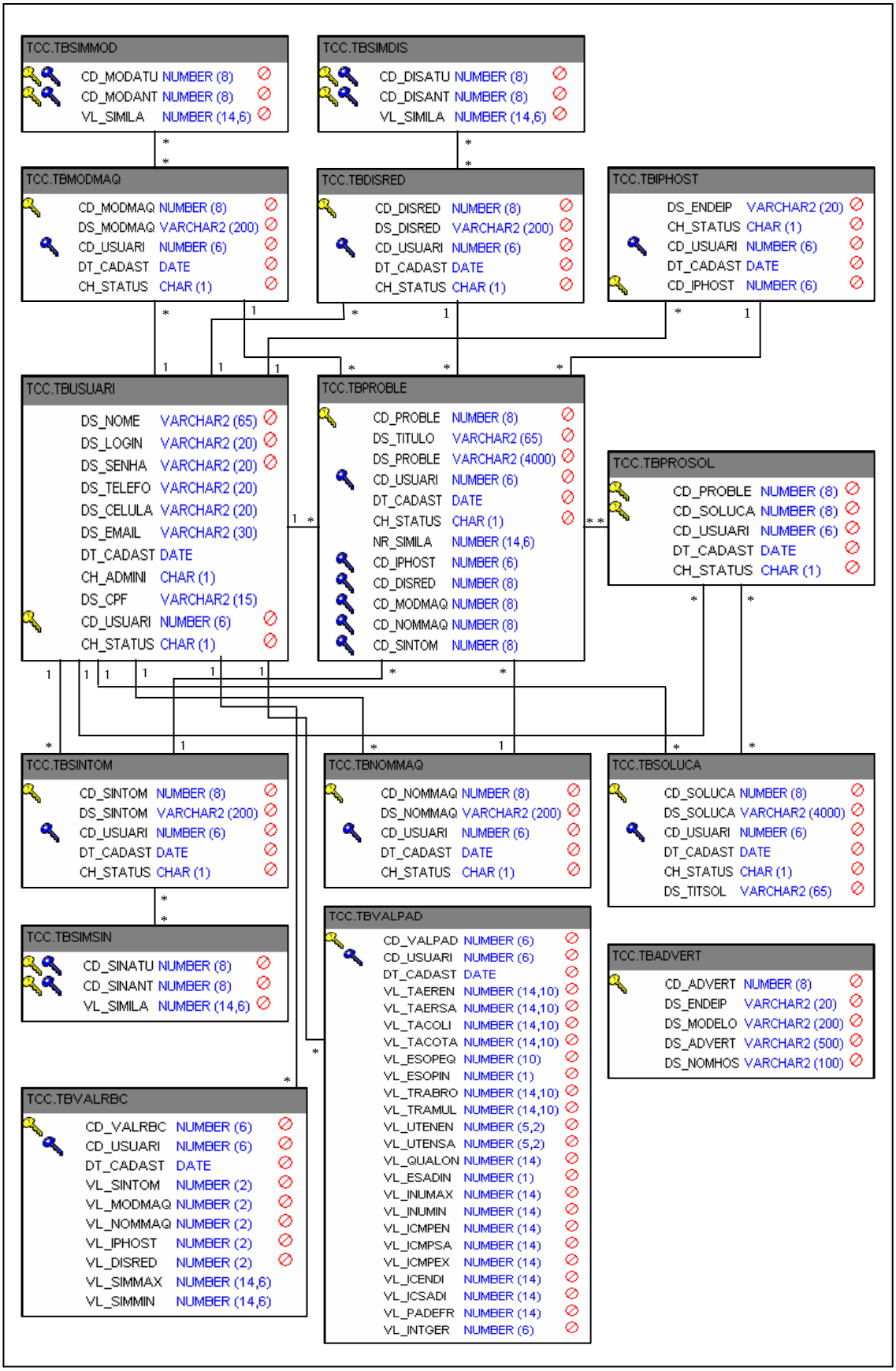

Figura 6. Diagrama de Entidade-Relacionamento do ANTIFAIL

A partir do momento em que a estrutura do banco dados foi criada, pode-se então começar a desenvolver a aplicação, realizando primeiramente um levantamento acerca das variáveis MIB utilizadas.

6.2 CAPTURANDO INFORMAÇÕES DA MIB VIA SNMP

A partir de um levantamento feito acerca das variáveis da MIB II[13] contidas nos equipamentos que implementam o protocolo SNMP, pôde-se então, por meio da estratégia de raciocínio implementada na aplicação Gerente efetuar os respectivos diagnósticos:

a) maiores que o valor padrão especificado no cadastro de parâmetros para gerência de redes do sistema: após o intervalo entre uma coleta e outra do valor de uma variável, resulta numa diferença. Se o valor resultante desta diferença for maior que o cadastrado nos parâmetros para gerência de redes no sistema, é emitido um alarme correspondente ao problema.

Tabela 2. Diagnósticos com valor maior que o padrão

taxa de erros de entrada
taxa de erros de saída
taxa de colisões
tráfego de *broadcast*
tráfego de *multcast*
utilização de enlace de entrada
utilização de enlace de saída
ocorrência de quadros muito longos
ocorrência de inundações por tempo
ocorrência de inundações por discarte
ocorrência de mensagens ICMP de redirecionamento de entrada
ocorrência de mensagens ICMP de redirecionamento de saída
ocorrência de mensagens ICMP de tempo excedido
tráfego de entrada de mensagens ICMP de destino inalcançável
tráfego de saída de mensagens ICMP de destino inalcançável
quantidade de pacotes que estão sendo descartados por falta de rotas

[13] Base de informações para gerenciamento de um equipamento (MAURO;SCHMIDT, 2001)

b) menor que o valor padrão especificado no cadastro de parâmetros para gerência de redes: após o intervalo entre uma coleta e outra do valor de uma variável, resulta numa diferença. Se o valor resultante desta diferença for menor que o cadastrado nos parâmetros para gerência de redes no sistema, é emitido um alarme correspondente ao problema.

Tabela 3. Diagnósticos com valor menor que o padrão

equipamento reiniciando com freqüência

c) diferentes do valor padrão especificado: após uma única coleta, se o valor da variável for diferente do valor definido na implementação, é emitido um alarme correspondente ao problema.

Tabela 4. Diagnósticos com valores diferentes aos definidos na implementação

ocorrência de incremento da taxa de colisões tardias
estado operacional da interface de rede não disponível
estado administrativo da interface de rede não disponível

A etapa de gerenciamento pode ser representada pela Figura 6, onde pode-se observar somente o funcionamento do processo de gerência dos equipamentos. Primeiramente a estação gerente busca no banco de dados os endereços IPs dos equipamentos que são monitorados e os valores dos parâmetros para gerência de redes. Depois ele captura os valores das MIB e compara com os parametrizados. Caso o valor de alguma variável não esteja condizente com o parametrizado, é emitido um alarme.

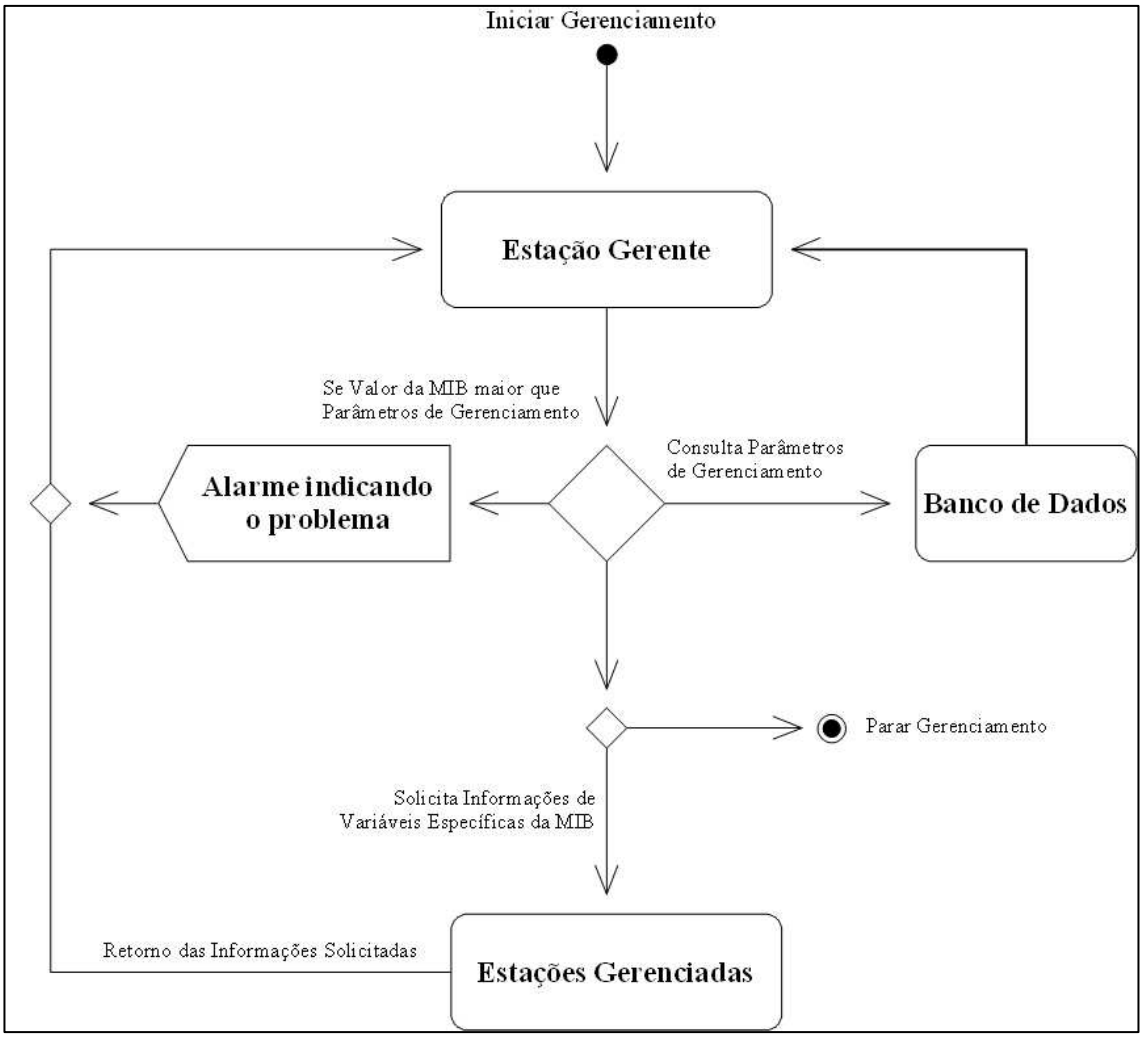

Figura 7. Diagrama de Atividade do Processo de Gerenciamento

A seguir serão descritos os critérios e as variáveis da MIB II que foram utilizados para diagnosticar os problemas citados anteriormente (LOPES; SAUVÉ; NICOLLETTI, 2003).

6.2.1 Variáveis com Valores Capturados em Momentos Distintos

A captura dos valores das variáveis a seguir devem ser efetuadas em momentos distintos. A primeira coleta é realizada ao iniciar o processo de gerenciamento e as demais serão feitas após o intervalo de tempo especificado no

cadastro de parâmetros para gerência de redes. Por último, compara-se o valor resultante das funções com o armazenado no cadastro de Parâmetros de Rede do sistema, conforme a ilustração 8:

Figura 8. Cadastro de Parâmetros para Gerência de Redes

Caso o valor retornado pela função seja maior que o cadastrado nos parâmetros, é disparado um alarme informando o problema, como pode ser visualizado pela Figura 9:

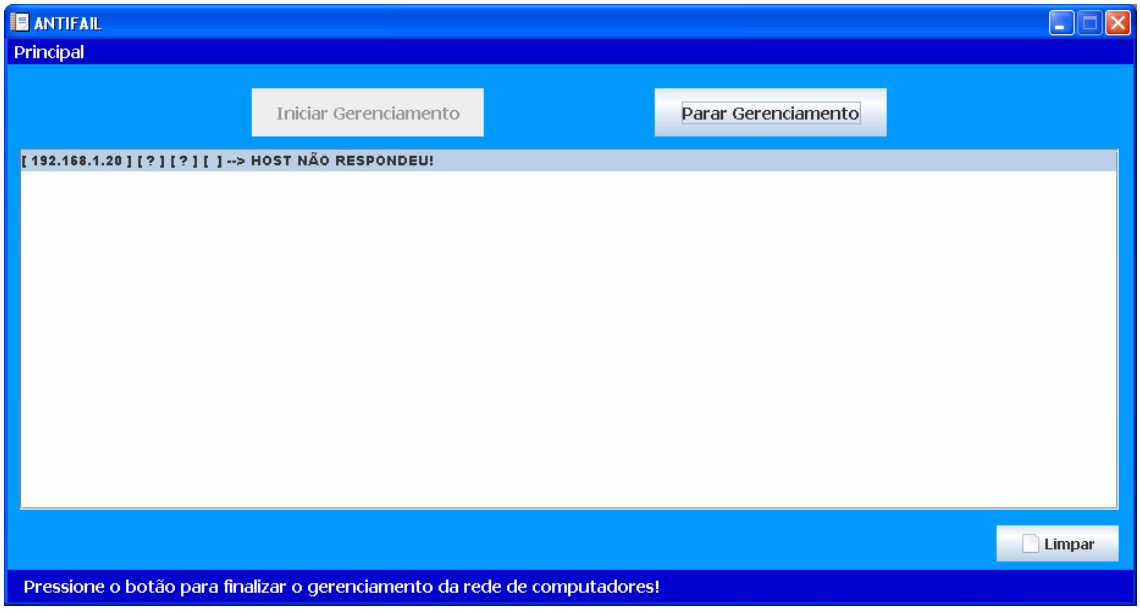

Figura 9. Alarmes Disparados

Nos próximos itens são descritos os procedimentos e as variáveis MIB utilizadas para realizar cada diagnóstico.

6.2.1.1 Taxa de Erros de Entrada

A taxa de erros de entrada tem por objetivo informar o percentual de quadros que chegaram com erros em uma interface, e conseqüentemente não puderam ser entregues a protocolos de camadas superiores.

Antes de se efetuar o cálculo da taxa de erros de entrada, é necessário capturar os valores das seguintes variáveis da MIB II, que serão utilizadas neste cálculo:

a) *IfInErros*: quantidade de quadros que chegaram com erros a uma interface. É definida pela OID 1.3.6.1.2.1.2.2.1.14.1;

b) *IfInBroadcastPkts*: quantidade de quadros com endereços-destino *broadcast* que chegaram na interface e foram entregues a protocolos da camada superior. É definida pela OID 1.3.6.1.2.1.31.1.1.1.3;

c) *IfInMultcastPkts*: quantidade de quadros com endereços-destino *multcast* que chegaram na interface e foram entregues a protocolos da camada superior. É definida pela OID 1.3.6.1.2.1.31.1.1.1.2;

d) *IfInUcastPkts*: quantidade de quadros com endereços-destino *unicast* que chegaram na interface e foram entregues a protocolos da camada superior. É definida pela OID 1.3.6.1.2.1.2.2.1.11.1;

A comparação é realizada pela seguinte função (LOPES; SAUVÉ; NICOLLETTI, 2003):

TE = (5)

((ifinerrosB – ifinerrosA) / (ifinucastpktsB - ifinucastpktsA + ifinmulticastpktsB - ifinmulticastpktsA + ifinbroadcastpktsB - ifinbroadcastpktsA + ifinerrosB - ifinerrosA)) * 100.

Onde:

a) TE: percentual da taxa de erros de entrada;
b) ifinerrosA: valor da coleta anterior da variável *ifinerros*;
c) ifinerrosB: valor da última coleta da variável *ifinerros*;
d) ifinucastpktsA: valor da coleta anterior da variável *ifinucastpkts*;
e) ifinucastpktsB: valor da última coleta da variável *ifinucastpkts*;
f) ifinmulticastpktsA: valor da coleta anterior da variável *ifinmulticastpkts*;
g) ifinmulticastpktsB: valor da última coleta da variável *ifinmulticastpkts*;
h) ifinbroadcastpktsA: valor da coleta anterior da variável *ifinbroadcastpkts*;
i) ifinbroadcastpktsB: valor da última coleta da variável *ifinbroadcastpkts*;

6.2.1.2 Taxa de Erros de Saída

O objetivo da taxa de erros de saída é informar o percentual de quadros que não foram transmitidos devido à ocorrência de erros.

Antes de se efetuar o cálculo da taxa de erros de entrada, é necessário capturar os valores das seguintes variáveis da MIB II, que serão utilizadas neste cálculo:

a) *IfOutErros*: quantidade de quadros que não foram transmitidos porque apresentaram erros. É definida pela OID 1.3.6.1.2.1.2.2.1.20.1;
b) *IfOutBroadcastPkts*: quantidade de quadros com endereços-destino

broadcast que foram requisitados por protocolos de camada superior. É definida pela OID 1.3.6.1.2.1.31.1.1.1.5;

c) *IfOutMultcastPkts*: quantidade de quadros com endereços-destino *multcast* que foram requisitados por protocolos de camada superior. É definida pela OID 1.3.6.1.2.1.31.1.1.1.4;

d) *IfOutUcastPkts*: quantidade de quadros com endereços-destino *unicast* que foram requisitados por protocolos de camada superior. É definida pela OID 1.3.6.1.2.1.2.2.1.17.1;

A comparação é realizada pela seguinte função (LOPES; SAUVÉ; NICOLLETTI, 2003):

TE = (6)

((ifouterrosB – ifouterrosA) / (ifoutucastpktsB - ifoutucastpktsA + ifoutmulticastpktsB - ifoutmulticastpktsA + ifoutbroadcastpktsB - ifoutbroadcastpktsA)) * 100.

Onde:

a) TE: percentual da taxa de erros de saída;

b) ifouterrosA: valor da coleta anterior da variável *ifinerros*;

c) ifouterrosB: valor da última coleta da variável *ifinerros*;

d) ifoutucastpktsA: valor da coleta anterior da variável *ifoutucastpkts*;

e) ifoutucastpktsB: valor da última coleta da variável *ifoutucastpkts*;

f) ifoutmulticastpktsA: valor da coleta anterior da variável *ifoutmulticastpkts*;

g) ifoutmulticastpktsB: valor da última coleta da variável *ifoutmulticastpkts*;

h) ifoutbroadcastpktsA: valor da coleta anterior da variável *ifoutbroadcastpkts*;

i) ifoutbroadcastpktsB: valor da última coleta da variável *ifoutbroadcastpkts*;

6.2.1.3 Taxa de Colisões

A taxa de colisões tem por objetivo informar o percentual de colisões ocorridas nas interfaces *Ethernet*.

Antes de se efetuar o cálculo da taxa de erros de entrada, é necessário capturar os valores das seguintes variáveis da MIB II, que serão utilizadas neste cálculo:

a) *dot3StatsSingleCollisionFrames*: quantidade de quadros que colidiram somente uma vez antes de serem transmitidos. É definida pela OID 1.3.6.1.2.1.10.7.2.1.4;
b) *dot3StatsMultipleCollisionFrames*: quantidade de quadros que colidiram várias antes de serem transmitidos. É definida pela OID 1.3.6.1.2.1.10.7.2.1.5;
c) *IfOutBroadcastPkts*: quantidade de quadros com endereços-destino *broadcast* que foram requisitados por protocolos de camada superior. É definida pela OID 1.3.6.1.2.1.31.1.1.1.5;
d) *IfOutMultcastPkts*: quantidade de quadros com endereços-destino *multcast* que foram requisitados por protocolos de camada superior. É definida pela OID 1.3.6.1.2.1.31.1.1.1.4;
e) *IfOutUcastPkts*: quantidade de quadros com endereços-destino *unicast* que foram requisitados por protocolos de camada superior. É definida pela OID 1.3.6.1.2.1.2.2.1.17.1;

A comparação é realizada pela seguinte função (LOPES; SAUVÉ;

NICOLLETTI, 2003):

TE = (7)

((dot3StatsSingleCollisionFramesB - dot3StatsSingleCollisionFramesA + dot3StatsMultipleCollisionFramesB - dot3StatsMultipleCollisionFramesA) / (ifoutucastpktsB - ifoutucastpktsA + ifoutmulticastpktsB - ifoutmulticastpktsA + ifoutbroadcastpktsB - ifoutbroadcastpktsA)) * 100.

Onde:

a) TE: percentual da taxa de colisões;
b) dot3StatsSingleCollisionFramesA: valor da coleta anterior da variável *dot3StatsSingleCollisionFrames*;
c) dot3StatsSingleCollisionFramesB: valor da última coleta da variável *dot3StatsSingleCollisionFrames*;
d) dot3StatsMultipleCollisionFramesA: valor da coleta anterior da variável *dot3StatsMultipleCollisionFrames*;
e) dot3StatsMultipleCollisionFramesB: valor da última coleta da variável *dot3StatsMultipleCollisionFrames*;
f) ifoutucastpktsA: valor da coleta anterior da variável *ifoutucastpkts*;
g) ifoutucastpktsB: valor da última coleta da variável *ifoutucastpkts*;
h) ifoutmulticastpktsA: valor da coleta anterior da variável *ifoutmulticastpkts*;
i) ifoutmulticastpktsB: valor da última coleta da variável *ifoutmulticastpkts*;
j) ifoutbroadcastpktsA: valor da coleta anterior da variável *ifoutbroadcastpkts*;
k) ifoutbroadcastpktsB: valor da última coleta da variável *ifoutbroadcastpkts*;

6.2.1.4 Tráfego de *Broadcast*

A verificação de quadros *broadcast* enviados e recebidos tem por objetivo informar a taxa de transmissão desses quadros em relação a um intervalo de tempo. Caso está taxa esteja elevada, há indícios de que algum problema está ocorrendo na rede, podendo diminuir a performance da mesma.

Antes de se efetuar o cálculo da taxa de tráfego *broadcast*, é necessário capturar os valores das seguintes variáveis da MIB II, que serão utilizadas neste cálculo:

a) *IfInBroadcastPkts*: quantidade de quadros que colidiram somente uma vez antes de serem transmitidos. É definida pela OID 1.3.6.1.2.1.31.1.1.1.3;
b) *IfOutBroadcastPkts*: quantidade de quadros que colidiram várias antes de serem transmitidos. É definida pela OID 1.3.6.1.2.1.31.1.1.1.5;

A comparação é realizada pela seguinte função (LOPES; SAUVÉ; NICOLLETTI, 2003):

TE = (8)

((IfInBroadcastPktsB - IfInBroadcastPktsA) +(IfOutBroadcastPktsB – IfOutBroadcastPktsA)) / T.

Onde:

a) TE: quantidade de quadros *broadcast* transmitidos por segundo;
b) IfInBroadcastPktsA: valor da coleta anterior da variável *IfInBroadcastPkts*;
c) IfInBroadcastPktsB: valor da última coleta da variável *IfInBroadcastPkts*;
d) IfOutBroadcastPktsA: valor da coleta anterior da variável

IfOutBroadcastPkts;

e) IfOutBroadcastPktsB: valor da última coleta da variável *IfOutBroadcastPkts*;

f) T: intervalo de tempo das duas coletas em segundos.

6.2.1.5 Tráfego de *Multcast*

A verificação de quadros *multcast* enviados e recebidos tem por objetivo informar a taxa de transmissão desses quadros em relação a um intervalo de tempo. Caso está taxa esteja elevada, há indícios de que algum problema está ocorrendo na rede, podendo diminuir a performance da mesma.

Antes de se efetuar o cálculo da taxa de tráfego *multcast*, é necessário capturar os valores das seguintes variáveis da MIB II, que serão utilizadas neste cálculo:

a) *IfInMultcastPkts*: quantidade de quadros que colidiram somente uma vez antes de serem transmitidos. É definida pela OID 1.3.6.1.2.1.31.1.1.1.2;

b) *IfOutMultcastPkts*: quantidade de quadros que colidiram várias antes de serem transmitidos. É definida pela OID 1.3.6.1.2.1.31.1.1.1.4;

A comparação é realizada pela seguinte função (LOPES; SAUVÉ; NICOLLETTI, 2003):

TE = (9)

((IfInMultcastPktsB - IfInMultcastPktsA) +(IfOutMultcastPktsB – IfOutMultcastPktsA)) / T.

Onde:

a) TE: quantidade de quadros *multcast* transmitidos por segundo;

b) IfInMultcastPktsA: valor da coleta anterior da variável *IfInMultcastPkts*;

c) IfInMultcastPktsB: valor da última coleta da variável *IfInMultcastPkts*;

d) IfOutMultcastPktsA: valor da coleta anterior da variável *IfOutMultcastPkts*;

e) IfOutMultcastPktssB: valor da última coleta da variável *IfOutMultcastPkts*;

f) T: intervalo de tempo das duas coletas em segundos.

6.2.1.6 Utilização de Enlace de Entrada

A utilização de enlace de entrada tem por objetivo, informar o percentual de utilização da mesma, e consequentemente avaliar o congestionamento da rede.

Antes de se efetuar o cálculo da taxa de utilização, é necessário capturar os valores das seguintes variáveis da MIB II, que serão utilizadas neste cálculo:

a) *ifInOctets*: quantidade de *bytes* recebidas por uma interface. É definida pela OID 1.3.6.1.2.1.2.2.1.10.1;

b) *ifSpeed*: velocidade em que a interface está operando. É definida pela OID 1.3.6.1.2.1.2.2.1.5.1;

A comparação é realizada pela seguinte função (LOPES; SAUVÉ; NICOLLETTI, 2003):

TE = ((ifInOctetsB – ifInOctetsA) * 8 * 100) / (300 * ifSpeed). (10)

Onde:

a) TE: percentual da taxa de utilização;

b) ifInOctetsA: valor da coleta anterior da variável *ifInOctets*;

c) ifInOctetsB: valor da última coleta da variável *ifInOctets*;

d) ifSpeed**:** valor da coleta da variável *ifSpeed***.**

6.2.1.7 Utilização de Enlace de Saída

A utilização de enlace de saída tem por objetivo, informar o percentual de utilização da mesma, e consequentemente avaliar o congestionamento da rede.

Antes de se efetuar o cálculo da taxa de utilização, é necessário capturar os valores das seguintes variáveis da MIB II, que serão utilizadas neste cálculo:

a) *ifOutOctets*: quantidade de *bytes* recebidas por uma interface. É definida pela OID 1.3.6.1.2.1.2.2.1.16.1;

b) *ifSpeed*: velocidade em que a interface está operando. É definida pela OID 1.3.6.1.2.1.2.2.1.5.1;

A comparação é realizada pela seguinte função (LOPES; SAUVÉ; NICOLLETTI, 2003):

TE = ((ifOutOctetsB – ifOutOctetsA) * 8 * 100) / (300 * ifSpeed). (11)

Onde:

a) TE: percentual da taxa de utilização;

b) ifOutOctetsA: valor da coleta anterior da variável *ifOutOctets*;

c) ifOutOctetsB: valor da última coleta da variável *ifOutOctets*;

d) ifSpeed*:* valor da coleta da variável *ifSpeed.*

6.2.1.8 Ocorrência de Quadros Muito Longos

Os quadros transmitidos via *Ethernet* podem ter no máximo 1518 bytes. Porém, alguns quadros podem apresentar mais que 1518 bytes, e são conhecidos como quadros muito longos. A ocorrência deste significa que alguma interface com defeito está emitindo-o.

Antes de se efetuar o cálculo para obtenção da quantidade deste tipo de quadro, é necessário capturar o valor de *dot3StatsFrameTooLongs* da MIB II, representada pela OID 1.3.6.1.2.1.10.7.2.1.13, que será utilizada neste cálculo.

A comparação é realizada pela seguinte função (LOPES; SAUVÉ; NICOLLETTI, 2003):

TE = dot3StatsFrameTooLongsB – dot3StatsFrameTooLongsA. (12)

Onde:

a) TE: quantidade de quadros;
b) dot3StatsFrameTooLongsA: valor da coleta anterior da variável *dot3StatsFrameTooLongs*;
c) dot3StatsFrameTooLongsB: valor da última coleta da variável *dot3StatsFrameTooLongs*;

6.2.1.9 Ocorrência de Mensagens ICMP de Redirecionamento de Entrada

Mensagens ICMP de Redirecionamento de Entrada são enviadas pelo roteador aos hospedeiros para informar ao mesmo que ele deveria utilizar outra rota para se comunicar com o endereço de destino. Quando há uma ocorrência elevada deste tipo de mensagem, indica que as tabelas de roteamento dos hospedeiros estão incompletas ou incorretas.

Antes de se efetuar o cálculo para obtenção da quantidade deste tipo de mensagem, é necessário capturar o valor de *icmpInRedirects* da MIB II, representada pela OID 1.3.6.1.2.1.5.7.0, que será utilizada neste cálculo.

A comparação é realizada pela seguinte função (LOPES; SAUVÉ; NICOLLETTI, 2003):

TE = icmpInRedirectB – icmpInRedirectA. (13)

Onde:

a) TE: quantidade de mensagens ICMP;

b) icmpInRedirectA: valor da coleta anterior da variável *icmpInRedirect*;

c) icmpInRedirectB: valor da última coleta da variável *icmpInRedirect*;

6.2.1.10 Ocorrência de Mensagens ICMP de Redirecionamento de Saída

Mensagens ICMP de Redirecionamento de Saída são emitidas somente pelo roteador. Quando há uma ocorrência elevada deste tipo de mensagem, indica que as tabelas de roteamento do roteador estão incompletas ou incorretas.

Antes de se efetuar o cálculo para obtenção da quantidade deste tipo de mensagem, é necessário capturar o valor de *icmpOutRedirects* da MIB II, representada pela OID 1.3.6.1.2.1.5.20.0, que será utilizada neste cálculo.

A comparação é realizada pela seguinte função (LOPES; SAUVÉ; NICOLLETTI, 2003):

TE = icmpOutRedirectB – icmpOutRedirectA. (14)

Onde:

a) TE: quantidade de mensagens ICMP;

b) icmpOutRedirectA: valor da coleta anterior da variável *icmpOutRedirect*;

c) icmpOutRedirectB: valor da última coleta da variável *icmpOutRedirect*;

6.2.1.11 Ocorrência de Mensagens ICMP de Tempo Excedido

Mensagens ICMP de tempo excedido são emitidas pelo roteador ao

hospedeiro. Quando há uma ocorrência elevada deste tipo de mensagem, indica que há algum problema na rede, exigindo uma investigação mais detalhada.

Antes de se efetuar o cálculo para obtenção da quantidade deste tipo de mensagem, é necessário capturar o valor de *icmpOutTimeExcds* da MIB II, representada pela OID 1.3.6.1.2.1.5.17.0, que será utilizada neste cálculo.

A comparação é realizada pela seguinte função (LOPES; SAUVÉ; NICOLLETTI, 2003):

TE = icmpOutTimeExcdsB – icmpOutTimeExcdsA. (15)

Onde:

a) TE: quantidade de mensagens ICMP;
b) icmpOutTimeExcdsA: valor da coleta anterior da variável *icmpOutTimeExcds*;
c) icmpOutTimeExcdsB: valor da última coleta da variável *icmpOutTimeExcds*;

6.2.1.12 Tráfego de Entrada de Mensagens ICMP de Destino Inalcançável

Mensagens de Entrada ICMP de Destino Inalcançável são recebidas pelo equipamento quando o mesmo tenta enviar algo para um destino inalcançável.

Antes de se efetuar o cálculo para obtenção da quantidade deste tipo de mensagem, é necessário capturar o valor de *icmpInDestUnreachs* da MIB II, representada pela OID 1.3.6.1.2.1.5.3.0, que será utilizada neste cálculo.

A comparação é realizada pela seguinte função (LOPES; SAUVÉ; NICOLLETTI, 2003):

TE = icmpInDestUnreachsB – icmpInDestUnreachsA. (16)

Onde:

a) TE: quantidade de mensagens ICMP;
b) icmpInDestUnreachsA: valor da coleta anterior da variável *icmpInDestUnreachs*;
c) icmpInDestUnreachsB: valor da última coleta da variável *icmpInDestUnreachs*;

6.2.1.13 Tráfego de Saída de Mensagens ICMP de Destino Inalcançável

Mensagens de Saída ICMP de Destino Inalcançável são emitidas pelo equipamento quando o mesmo tenta enviar algo para um destino inalcançável.

Antes de se efetuar o cálculo para obtenção da quantidade deste tipo de mensagem, é necessário capturar o valor de *icmpOutDestUnreachs* da MIB II, representada pela OID 1.3.6.1.2.1.5.16.0, que será utilizada neste cálculo.

A comparação é realizada pela seguinte função (LOPES; SAUVÉ; NICOLLETTI, 2003):

TE = icmpOutDestUnreachsB – icmpOutDestUnreachsA. (17)

Onde:

a) TE: quantidade de mensagens ICMP;
b) icmpOutDestUnreachsA: valor da coleta anterior da variável *icmpOutDestUnreachs*;
c) icmpOutDestUnreachsB: valor da última coleta da variável *icmpOutDestUnreachs*.

6.2.1.14 Quantidade de Pacotes que Estão Sendo Descartados por Falta de Rotas

Quando um roteador recebe um pacote cujo destino não é especificado, o mesmo é descartado por falta de rota.

Antes de se efetuar o cálculo para obtenção da quantidade deste tipo de ocorrência, é necessário capturar o valor de *ipOutNoRoutes* da MIB II, representada pela OID 1.3.6.1.2.1.4.12.0, que será utilizada neste cálculo.

A comparação é realizada pela seguinte função (LOPES; SAUVÉ; NICOLLETTI, 2003):

TE = ipOutNoRoutesB – ipOutNoRoutesA. (18)

Onde:

a) TE: quantidade de pacotes;

b) ipOutNoRoutesA: valor da coleta anterior da variável *ipOutNoRoutes*;

c) ipOutNoRoutesB: valor da última coleta da variável *ipOutNoRoutes*.

6.2.1.15 Ocorrência de Incremento da Taxa de Colisões Tardias

Colisões tardias ocorrem quando uma colisão é detectada sendo que já foram transmitidos mais de 512 bits de pelo menos um quadro envolvido. Quando este tipo de evento ocorre, é indício forte de que há algum problema na rede.

O cálculo para verificação da ocorrência deste tipo de colisão necessita o valor da variável *dot3StatsLateCollisions* da MIB II, identificado pela OID 1.3.6.1.2.1.10.7.2.1.8, que será utilizada neste cálculo.

A comparação é realizada pela seguinte função (LOPES; SAUVÉ; NICOLLETTI, 2003):

TE = dot3StatsLateCollisionsB - dot3StatsLateCollisionsA. (19)

Onde:

a) TE: valor resultante da diferença entre as duas coletas;

b) dot3StatsLateCollisionsA: valor da coleta anterior da variável *dot3StatsLateCollisions*;

c) dot3StatsLateCollisionsB: valor da última coleta da variável *dot3StatsLateCollisions*;

6.2.1.16 Equipamento Reiniciando com Freqüência

É importante verificar se os equipamentos estão reiniciando com freqüência, pois existem alguns que dificilmente se consegue perceber tal fato. Com o auxílio de uma estação de gerência SNMP, pode-se monitorar este tipo de ocorrência.

O cálculo para verificação da freqüência de reinicialização de equipamentos, necessita da obtenção do valor da variável *sysUpTime* da MIB II, identificada pela OID 1.3.6.1.2.1.1.3.0, que será utilizada neste cálculo. Esta variável armazena a quantidade de tempo que o equipamento está operando desde a ultima vez que foi reiniciado.

A comparação é realizada pela seguinte função (LOPES; SAUVÉ; NICOLLETTI, 2003):

TE = sysUpTimeB - sysUpTimeA**.** (20)

a) TE: valor resultante da diferença entre as duas coletas;

b) sysUpTimeA: valor da coleta anterior da variável *sysUpTime*;

c) sysUpTimeB: valor da última coleta da variável *sysUpTime*;

Após realizar todos os procedimentos referentes as variáveis cujo valor é necessário ser capturado em momentos distintos, inicia-se as rotinas para as variáveis

que necessitam somente uma coleta.

6.2.2 Variáveis com Valores Capturados em um Único Momento

A captura dos valores das variáveis a seguir são efetuadas em um único momento, ou seja, apenas uma coleta já é o suficiente para o sistema realizar o diagnóstico. A comparação é realizada a cada captura, onde o valor resultante das funções deverá ser condizente com o parametrizado.

6.2.2.1 Estado Administrativo da Interface de Rede não Disponível

Verificar o estado administrativo das interfaces de rede é importante, pois se elas não estiverem operacionais, pode indicar que a mesma está desabilitada ou com problemas.

A verificação do estado administrativo das interfaces de rede, necessita da obtenção do valor da variável *ifAdminStatus* da MIB II, identificada pela OID 1.3.6.1.2.1.2.2.1.7.1, que é utilizada neste cálculo. Esta variável armazena os seguintes valores (LOPES; SAUVÉ; NICOLLETTI, 2003):

a) 1: ativa;

b) 2: inativa;

c) 3: em modo de teste;

Se o valor de *ifAdminStatus* for diferente de um, é disparado um alarme informando que o estado administrativo da interface de rede não está ativo.

6.2.2.2 Ocorrência de Inundações por Tempo

A ocorrência de inundações consiste em quadros enviados para todas as portas de um comutador. Isso ocorre porque no momento da transmissão do quadro não foi encontrado o endereço físico de destino na tabela de endereços do comutador.

A verificação de inundações necessita da obtenção do valor da variável *dot1dTpAgingTime* da MIB II, identificada pela OID 1.3.6.1.2.1.17.4.2, que é utilizada neste cálculo. Esta variável indica o tempo em que os endereços contidos na tabela do comutador estarão disponíveis (LOPES; SAUVÉ; NICOLLETTI, 2003).

Compara-se o valor *dot1dTpAgingTime* com o armazenado no campo Quantidade de Inundações por Tempo do cadastro de Parâmetros de Rede do sistema. Caso o valor retornado pela função seja menor que o cadastrado nos parâmetros, é disparado um alarme informando que a quantidade de inundações por tempo está menor que o valor padrão especificado.

6.2.2.3 Ocorrência de Inundações por Discarte

A ocorrência de inundações por discarte consiste em quadros enviados para todas as portas de um comutador. Isso ocorre porque no momento da transmissão do quadro não foi encontrado o endereço físico de destino na tabela de endereços do comutador. Isso porque, não há mais espaço para novos endereços nesta tabela.

A verificação de inundações necessita da obtenção do valor da variável *dot1dTpLearnedEntryDiscards* da MIB II, identificada pela OID 1.3.6.1.2.1.17.4.1, que é utilizada neste cálculo. Esta variável indica a quantidade de vezes que uma entrada na tabela de endereços foi descartada por falta de espaço (LOPES; SAUVÉ;

NICOLLETTI, 2003).

Compara-se o valor *dot1dTpLearnedEntryDiscards* com o armazenado no campo Quantidade de Inundações por Descarte do cadastro de Parâmetros de Rede do sistema. Caso o valor retornado pela função seja maior que o cadastrado nos parâmetros, é disparado um alarme informando que a quantidade de inundações por discarte está maior que o valor padrão especificado.

6.2.2.4 Estado Operacional da Interface de Rede não Disponível

Verificar o estado operacional das interfaces de rede é importante, pois se elas não estiverem operacionais, indica que os equipamentos estão desligados ou apresentando alguma falha no que se refere a sua operação.

A verificação do estado operacional das interfaces de rede dos equipamentos, necessita da obtenção do valor da variável *ifOperStatus* da MIB II, identificada pela OID 1.3.6.1.2.1.2.2.1.8.1, que é utilizada nesta comparação. Esta variável armazena os seguintes valores (LOPES; SAUVÉ; NICOLLETTI, 2003):

a) *up* (1): operacional;
b) *down* (2): não operacional;
c) *testing* (3): em modo de teste;
d) *unknown* (4): em estado indeterminado;
e) *dormant* (5): sem condições de transmitir quadros;
f) *notPresent* (6): faltando algum componente da interface;
g) *lowerLayerDown* (7): camadas inferiores com interfaces não operacional.

Se o valor de *ifOperStatus* for diferente de um, é disparado um alarme

informando que o estado operacional da interface de rede do equipamento não está operacional.

Após a realização de todos os procedimentos para efetuar a identificação de alguma anomalia na rede, inicia-se a aplicação do raciocínio baseado em casos para dar auxílio na tomada de decisões por parte dos profissionais responsáveis pelo gerenciamento da rede.

6.3 APLICANDO TÉCNICAS DE RBC PARA ENCONTRAR O CASO MAIS SIMILAR

A partir do levantamento realizado acerca das técnicas de RBC, pôde-se efetuar o desenvolvimento conforme apresentado a seguir, seguindo o ciclo de RBC.

O processo de RBC pode ser representado pelo diagrama de atividades ilustrado na Figura 10:

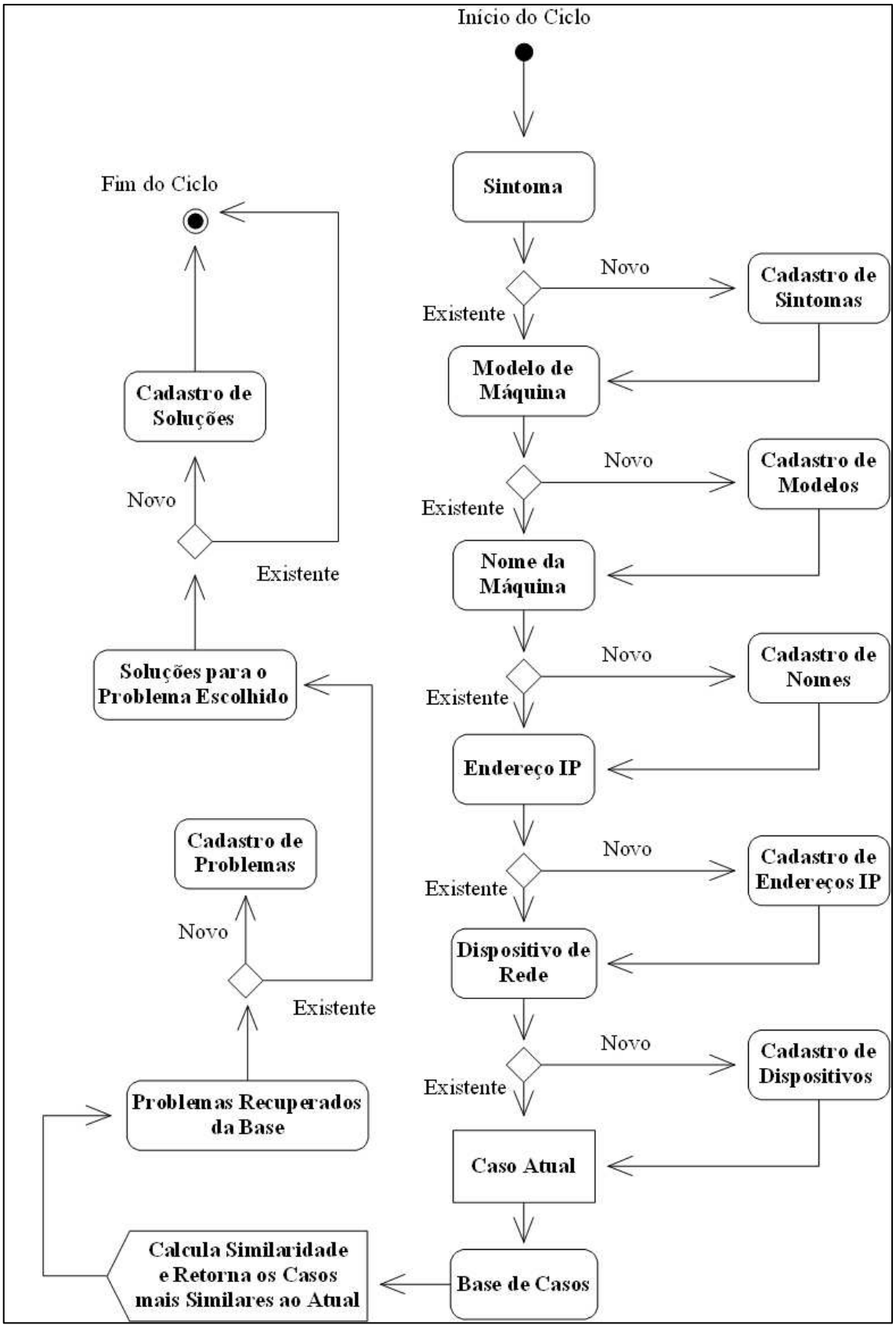

Figura 10. Diagrama de Atividade do Processo de RBC

6.3.1 Indexando os Casos

Os casos foram indexados por meio dos seguintes índices:

a) sintoma;

b) modelo da máquina;

c) nome da máquina na rede;

d) endereço IP;

e) dispositivo de rede.

Por meio destes índices será calculado o valor da similaridade entre o caso atual e os armazenados na base.

6.3.2 Calculando a Similaridade dos Casos

A princípio, calcula-se a similaridade local, que consiste em avaliar a semelhança entre os índices.

Os índices sintoma, modelo da máquina e dispositivo de rede utilizam a técnica de Contagem de Palavras, que tem por objetivo verificar o número de palavras idênticas entre dois casos. Este método pode ser representado pela função (VON WANGENHEIM; VON WANGENHEIM, 2003):

$$S = B / A. \quad (21)$$

Onde:

a) A é o número de palavras do caso que possui mais palavras;

b) B é o número de palavras idênticas entre os dois casos;

c) S é o valor da similaridade local.

Desta forma, se todas as palavras de ambos os casos forem idênticas, o valor

de S será 1.0. Caso todas sejam diferentes, S será 0.0, mas se existir algumas palavras iguais, S irá assumir o valor intermediário entre os limiares 1.0 e 0.0. As palavras com menos de três letras não são consideradas, pois representam pouca relevância para o caso, como pode ser observado no exemplo da Tabela 5:

Tabela 5. Contagem de Palavras

Índice	**Valor de entrada**	**Valor na Base**	**Similaridade**
Sintoma	Rede lenta	Rede sem conectividade	S =1/3 = 0,333333
Sintoma	Rede sem conectividade	Rede sem conectividade	S = 3/3 = 1
Sintoma	Cabo rompido	Rede sem conectividade	S = 0/3 = 0

Os índices Endereço IP e Nome da Máquina utilizam a técnica da Função Escada, que fundamenta-se no princípio de que o valor de um atributo é totalmente igual ou diferente ao mesmo de um outro caso. Esta técnica utiliza os limiares 1.0 para representar a igualdade total ou 0.0 para a desigualdade. Sendo assim, não existe valores intermediários, o que caracteriza uma função binária. Observe os exemplos a seguir:

Tabela 6. Função Escada

Índice	**Valor de entrada**	**Valor na Base**	**Similaridade**
Nome da Máquina	Comp Lab2	Comp Lab2	S = 1
Nome da Máquina	Comp Lab2	Comp Lab3	S = 0
Nome da Máquina	Comp Lab2	Comp2_01	S = 0

Feito os cálculos da similaridade local entre os índices, calcula-se a similaridade global, utilizando a técnica do *Nearest Neighbour* Ponderado, onde cada atributo pode ter uma importância diferente em relação aos demais para o cálculo da similaridade. Representa-se matematicamente pelo seguinte função (VON WANGENHEIM; VON WANGENHEIM, 2003):

$$S_1 = ((X_1 * V_1) + (Y_1 * V_2) + ... + (Z_1 * V_3)) / (V_1 + V_2 + V_3). \quad (22)$$

$$S_2 = ((X_2 * V_1) + (Y_2 * V_2) + ... + (Z_2 * V_3)) / (V_1 + V_2 + V_3).$$

Onde:

S é o valor da similaridade.

X, Y e Z são os valores de atributos diferentes.

V é o grau de importância para cada atributo.

Observe os exemplos a seguir:

Caso atual:

Sintoma: Rede lenta;

Modelo de Máquina: PentiumIV 3.0GHZ, 512MB de RAM

Nome da Máquina: Comp2_01

Endereço IP: 10.0.7.125

Dispositivo de rede: VIA Rhine II Fast Ethernet

Após montado o caso atual, compara-se o mesmo com todos os casos armazenados na base.

Caso armazenado na base:

Sintoma: Rede sem conectividade;

Modelo de Máquina: PentiumIII 750MHZ, 512MB de RAM

Nome da Máquina: Comp2_10

Endereço IP: 10.0.5.123

Dispositivo de rede: VIA Rhine II Fast Ethernet

No cálculo da similaridade local, deve-se aplicar o método da contagem de palavras para os índices: sintoma, modelo de máquina e dispositivo de rede. Já os índices nome da máquina e endereço IP aplica-se o método da função escada.

Calculando o valor da similaridade local:

- Índice: Sintoma;

- Valor de entrada: Rede Lenta;

- Valor armazenado na base: Rede sem conectividade;

Efetuando o cálculo por meio da fórmula 21:

Similaridade = 1 / 3 = 0,333333

- Índice: Modelo de máquina;

- Valor de entrada: PentiumIV 3.0GHZ, 512MB de RAM

- Valor armazenado na base: PentiumIII 750MHZ, 512MB de RAM

Efetuando o cálculo por meio da fórmula 21:

Similaridade = 2 / 4 = 0,5

- Índice: Nome da máquina;

- Valor de entrada: Comp2_01

- Valor armazenado na base: Comp2_10

Similaridade = 0

- Índice: Endereço IP;

- Valor de entrada: 10.0.7.125

- Valor armazenado na base: 10.0.5.123

Similaridade = 0

- Índice: Dispositivo de rede;

- Valor de entrada: VIA Rhine II Fast Ethernet

- Valor armazenado na base: VIA Rhine II Fast Ethernet

Efetuando o cálculo por meio da fórmula 21:

Similaridade = 4 / 4 = 1

O cálculo da similaridade global é realizado utilizando-se os valores da similaridade local e o peso que cada índice possui, sendo que estes são cadastrados nos Parâmetros de RBC do sistema e definidos pelo especialista de gerência de redes. A tela de cadastro pode ser visualizada pela Figura 11:

Figura 11. Cadastro dos parâmetros de RBC

Neste exemplo é usado os seguintes valores para cada índice:

- Sintoma = 10;

- Modelo da máquina = 1;

- Nome da maquina = 1;

- Endereço IP = 1;

- Dispositivo de rede = 1;

Isso significa que o atributo Sintoma é dez vezes mais importante que os demais. Sendo assim, o calculo da similaridade global conforme a técnica de *Nearest Neighbour* Ponderado (fórmula 22) fica da seguinte forma:

*Similaridade = ((0,333333 * 10) + (0,5 * 1) + (0 * 1) + (0 * 1) + (1 * 1)) / (10 + 1 + 1 + 1 + 1) = 0,345237*

Com este resultado, pode-se concluir que o grau de similaridade entre os dois casos é de *0,345237*, numa escala que vai de 0 à 1.

O sistema desenvolvido compara o caso atual com todos os demais armazenados na base, e recupera os que possuem o grau de similaridade entre os valores mínimo e máximo cadastrados nos Parâmetros de RBC. Caso os resultados obtidos não satisfaçam a situação atual, o sistema permite o usuário adaptar o caso desejado e posteriormente armazená-lo na base de conhecimento.

6.4 FUNCIONALIDADES DO SISTEMA

O sistema apresenta-se de forma simples e objetiva, com o intuito de auxiliar no gerenciamento de redes de computadores no que diz respeito à monitoração dos equipamentos e ao armazenamento das experiências profissionais aplicadas sobre os mesmos.

Inicialmente o uma tela de acesso, onde informa-se o nome de usuário e senha, com o objetivo de restringir a utilização do sistema por pessoas não autorizadas. A incorporação de novos usuários ao sistema é realizada por meio de uma tela de cadastro de usuários, que possibilita simultaneamente a inserção, exclusão ou apenas a consulta do usuário.

Após realizar o cadastro dos usuários que terão acesso ao sistema, é necessário cadastrar os seguintes requisitos do sistema:

a) endereço IP das máquinas que serão gerenciadas;

b) parâmetros de gerenciamento de redes;

c) parâmetros de RBC.

Todos os procedimentos acima são encontrados na sessão de cadastros, onde todos estão dispostos em telas separadas e devidamente identificadas. Com as informações até aqui cadastradas, pode-se dar início ao gerenciamento acionando o

botão Iniciar Gerenciamento localizado na tela principal, mas é necessário que os agentes SNMP das máquinas gerenciadas estejam ativados e funcionando corretamente, caso contrário, o sistema pode emitir um alarme informando que a máquina não responde às solicitações do gerente. O cancelamento da atividade de gerenciamento é realizada por meio do pressionamento do botão Parar Gerenciamento, que localiza-se ao lado do botão Iniciar Gerenciamento.

O tratamento das informações de gerenciamento acerca das falhas ocorridas na rede é realizado por meio do ciclo de RBC, localizado no menu principal. Este ciclo inicia-se com a escolha dos atributos de entrada do problema que se deseja resolver, que estão dispostos em seqüência e são exibidos automaticamente, para manter a consistência do ciclo de RBC e assim oferecer os benefícios que esta técnica pode oferecer. As telas são apresentadas na seguinte seqüência:

a) sintoma;

b) modelo da máquina;

c) nome da máquina;

d) endereço IP;

e) dispositivo de rede.

Após inserir os valores dos atributos de entrada, é exibida a tela contendo os resultados obtidos, se houver, pois estes dependem do grau de conhecimento que a base possui. Selecionando o resultado desejado nesta tela, o sistema exibe posteriormente a solução ou soluções para o problema desejado.

O sistema possui para todas as etapas de RBC a possibilidade de adaptação, pois se um atributo, problema ou solução não for satisfatório para o problema atual, pode-se inserir novos valores, finalizando o ciclo.

6.5 RESULTADOS OBTIDOS

Nesta etapa do trabalho são relatados os resultados obtidos, sendo que os mesmos foram realizados por meio de um estudo de caso baseado na base de conhecimento inicial do *AntiFail*. Este estudo é representado por meio de figuras do ciclo de RBC do sistema, cujo problema atual que deseja-se resolver consiste no baixo desempenho da rede, que a princípio não sabe-se qual a causa deste problema. A seguir são apresentadas as etapas do ciclo de RBC para o caso a ser resolvido:

a) O ciclo de RBC inicia-se com a escolha do atributo de entrada Sintoma, que recebe o valor *Rede Lenta*.

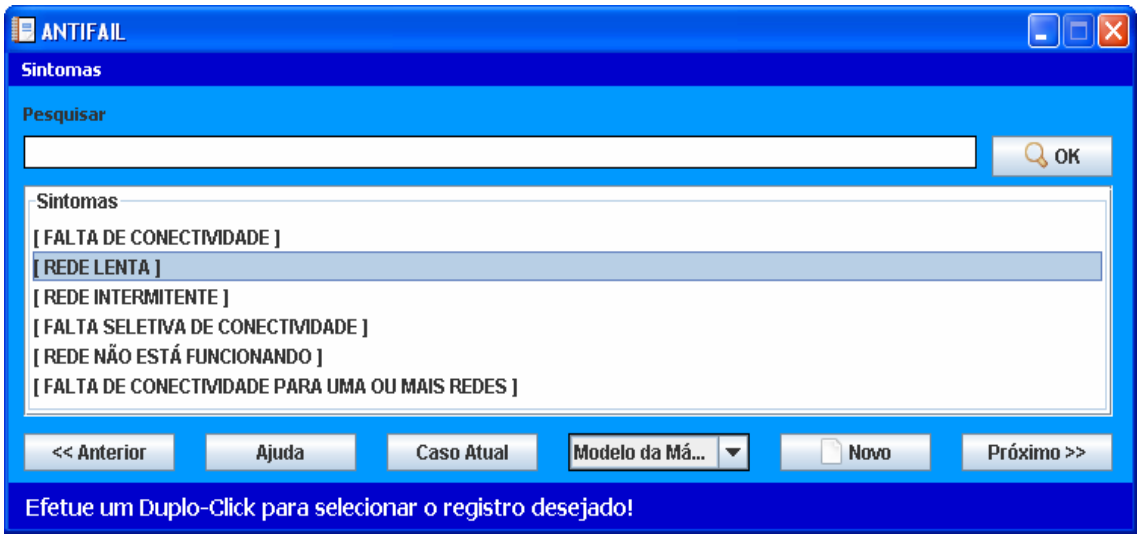

Figura 12. Sintoma

b) Em seguida, escolhe-se o valor para o atributo Modelo de Máquina, porém, como a principio não é possível determinar que o problema é específico de um equipamento, é atribuído o valor *null*.

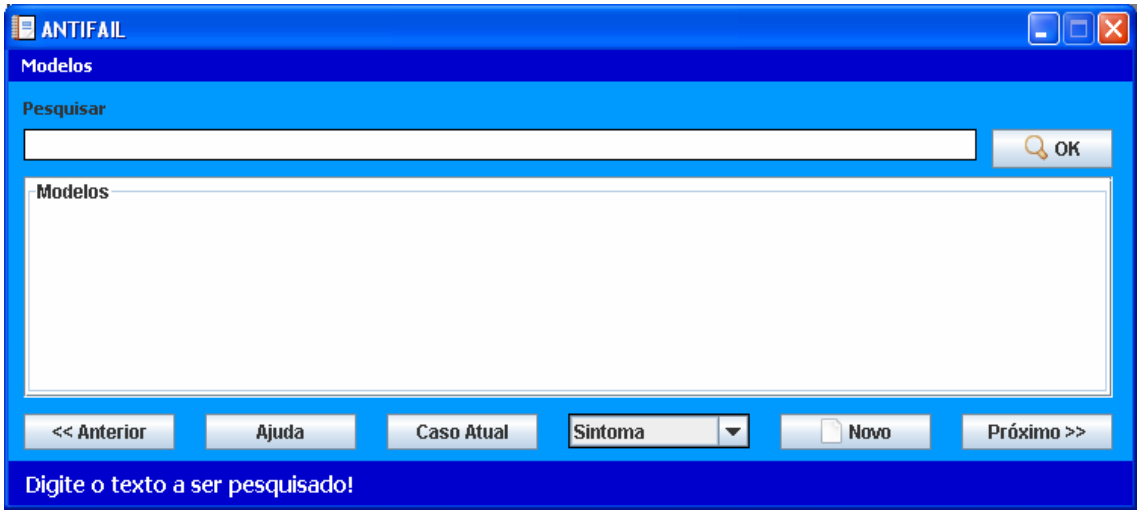

Figura 13. Modelo

c) Assim como ocorreu com o atributo de entrada Modelo de Máquina, utiliza-se da mesma lógica para atribuir valor ao Nome da Máquina na Rede, fazendo com que o mesmo receba *null.*

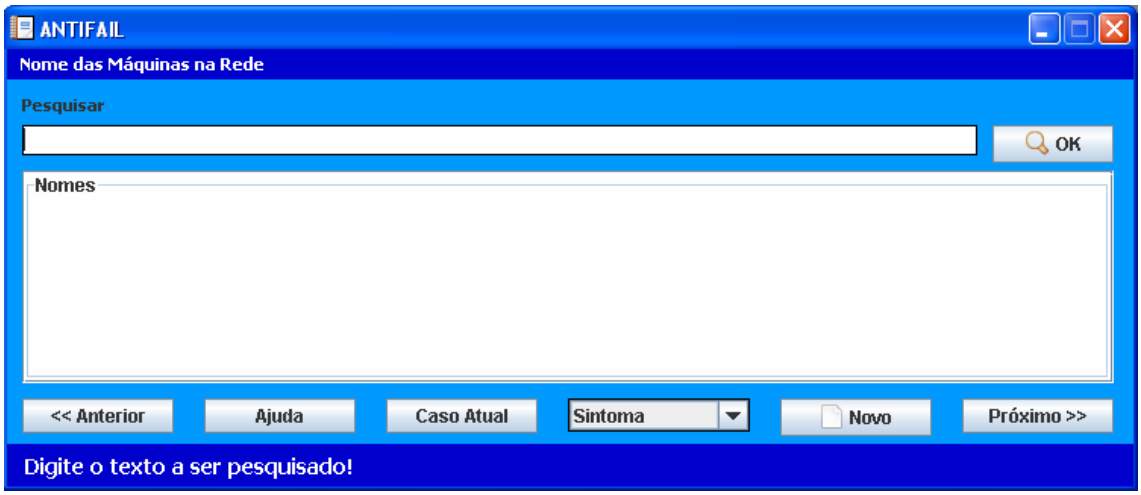

Figura 14. Nome

d) Utilizando-se do mesmo raciocínio do atributo anterior, o valor *null* é atribuído para o Endereço IP.

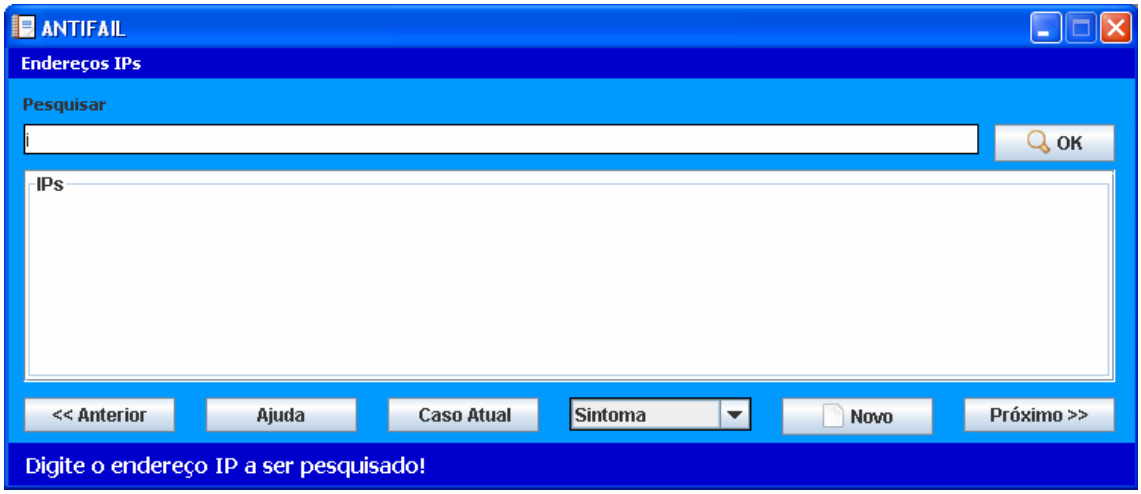

Figura 15. Endereço IP

e) Finalizando a atribuição dos valores de entrada do RBC, o atributo Dispositivo de Rede também recebe o valor *null*, pelos mesmos critério adotados anteriormente.

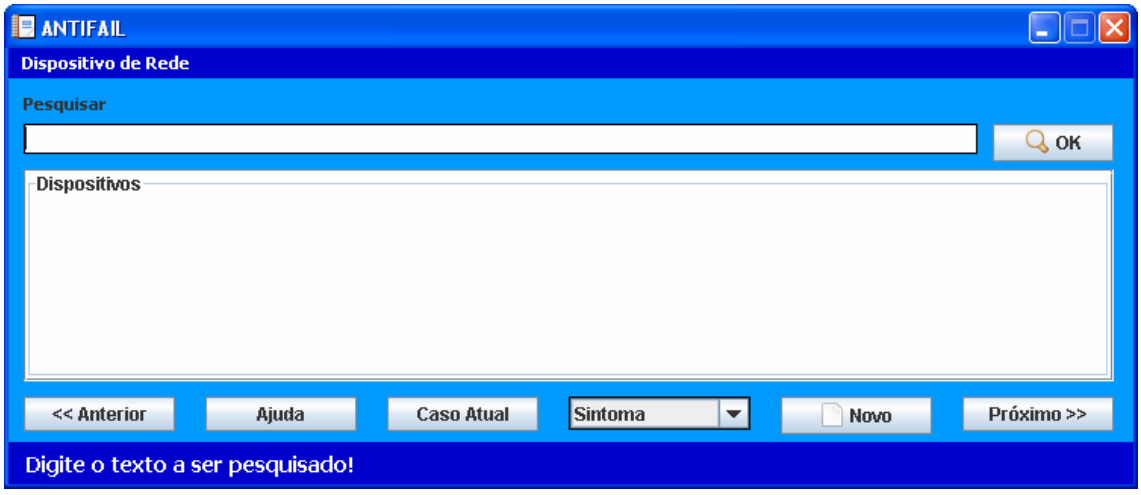

Figura 16. Dispositivo

A partir dos dados de entrada informados até agora, obteve-se os seguintes casos recuperados da base: *EQUIPAMENTO DE INTERCONEXÃO DEFEITUOSO, CONECTOR DEFEITUOSO e PLACA DE REDE DEFEITUOSA.*

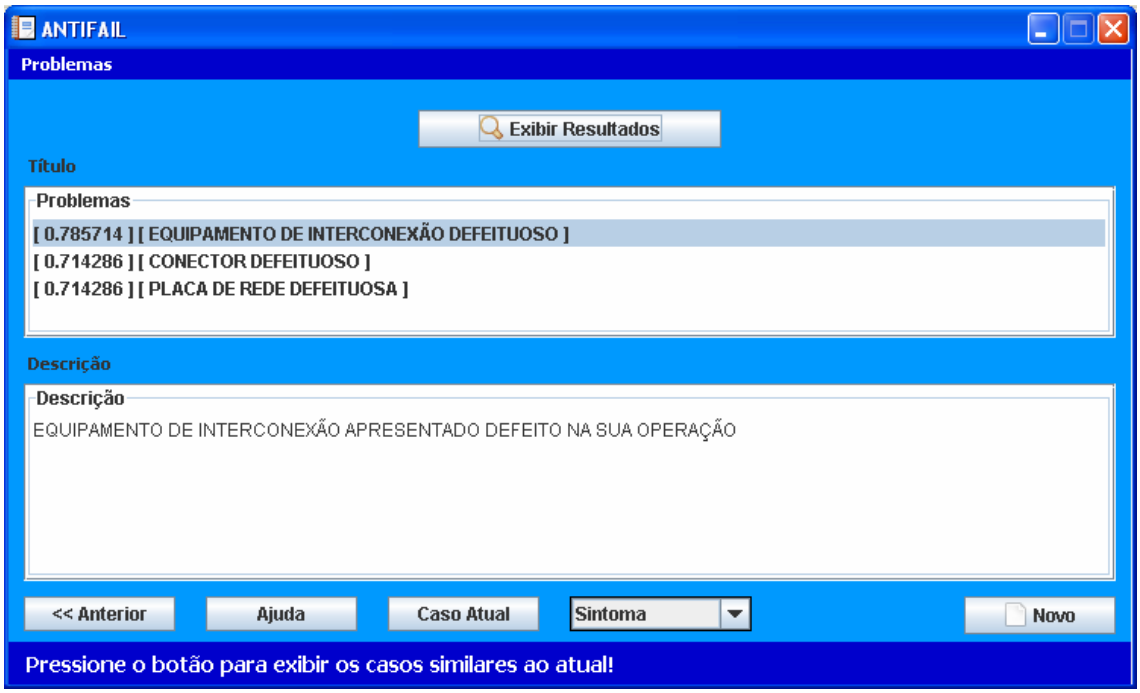

Figura 17. Problemas

Ao selecionar o problema *EQUIPAMENTO DE INTERCONEXÃO DEFEITUOSO*, o sistema retorna as seguintes soluções cadastradas para este problema: *EFETUAR A TROCA DO EQUIPAMENTO e REINICIAR O EQUIPAMENTO.*

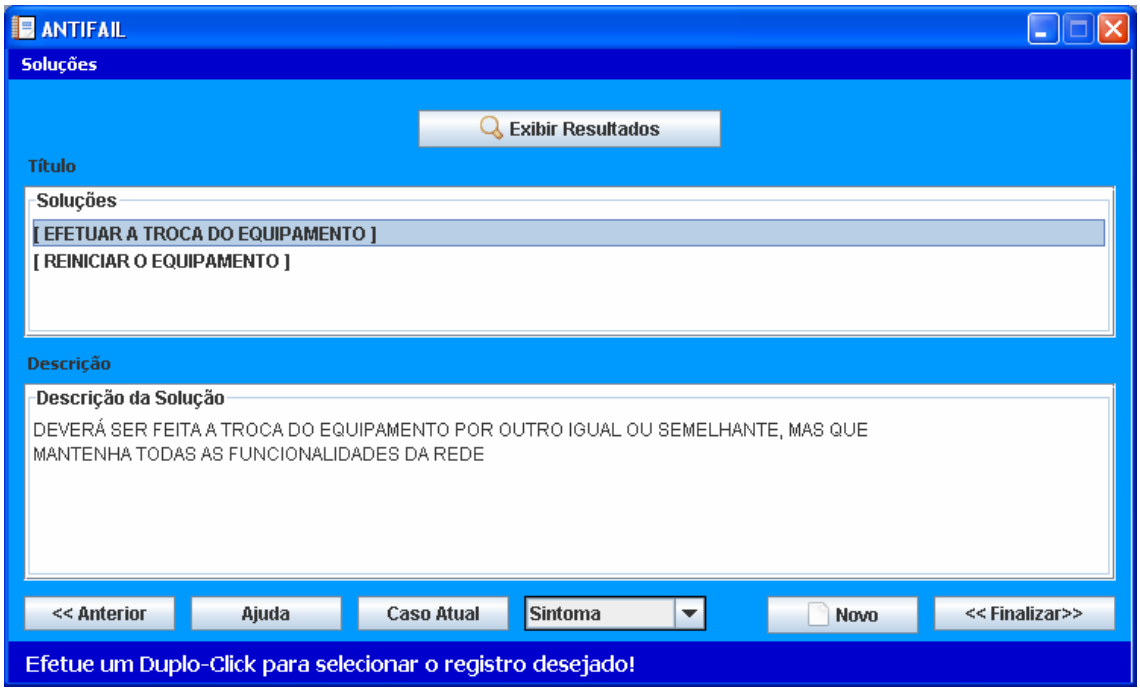

Figura 18. Solução

Observando o estudo de caso apresentado, pode-se afirmar que o sistema obteve resultados satisfatórios, pois conseguiu recuperar os casos mais similares ao

atual, prestando desta forma o auxilio a tomada de decisão, sendo que para isso, é importante ter uma boa base de conhecimento cadastrada.

CONCLUSÃO

À medida que as redes de computadores aumentam e se tornam cada vez mais importantes para as organizações, cresce a necessidade da utilização de ferramentas capazes de gerenciar com eficácia os equipamentos que a compõe.

Especialistas da área convivem quase que diariamente com problemas oriundos de diferentes origens problemáticas, sendo que os mesmos necessitam muitas vezes pesquisar sobre o assunto para encontrar a melhor solução. Sendo assim, a manifestação da inteligência do ser humano, pressupõe aquisição de conhecimento e armazenamento, voltando-se sempre para a adaptação às circunstâncias de seu meio. Essa inteligência que é construída pelo homem vem sendo aprimorada nas máquinas atuando como um instrumento de solução de problemas.

A inteligência de máquina é um tipo de raciocínio construído pelo homem, que estabelece instruções para que esta traga soluções rápidas e precisas, de modo que torne a sua vida cotidiana mais acessível.

Por meio da inteligência artificial integrada com o gerenciamento de redes de computadores, pôde-se capacitar o computador para que o mesmo apresente um comportamento inteligente, podendo atuar no auxílio à resolução de problemas com os especialistas da área.

O desenvolvimento desta pesquisa resultou em um protótipo para gerência de falhas denominado *Antifail*, que tem a capacidade de emitir alarmes informando a ocorrência de determinados eventos que foram definidos na sua implementação. Ele também permite a recuperação, adaptação e o armazenamento dos problemas ocorridos para serem utilizados futuramente, caso seja necessário. Para sua construção foram utilizados agentes SNMP para atender as solicitações do gerente SNMP, tendo como

base de informações uma *Management Information Base* (MIB).

No desenvolvimento do gerente SNMP, que equivale ao protótipo desenvolvido, foram definidas as variáveis MIB que são gerenciadas e a maneira que seus valores são tratados, possibilitando desta forma a criação de uma política de gerenciamento direcionada para a área de falhas. Além disso, o protótipo possui um sistema de raciocínio baseado em casos para tratar as informações relacionadas às falhas ocorridas na rede gerenciada, e desta forma auxiliar os profissionais responsáveis pelo gerenciamento na resolução de tais falhas.

O tratamento das informações referentes aos valores capturados das variáveis MIB, são comparados em relação a outros valores parametrizados no sistema. Estes por sua vez, são definidos pelo especialista da área de gerência de redes e podem ser alterados quando o mesmo achar conveniente. Já para o tratamento das informações no que se refere à recuperação dos problemas armazenados na base de casos, o sistema leva em consideração os valores parametrizados para a aplicação da técnica de RBC. Tais valores também podem ser definidos pelo especialista em gerência de redes, e também contar com a colaboração de um especialista da área de inteligência artificial.

Os resultados positivos obtidos com a aplicação deste protótipo num ambiente real foram:

a) emissão de alarmes informando anomalias nos objetos gerenciados;
b) resolução das anomalias encontradas antes que as mesmas pudessem ocasionar falhas graves;
c) recuperação de problemas similares armazenados na base de conhecimento em relação ao problema atual;
d) armazenamento de novos problemas na base de conhecimento;

e) auxílio na tomada de decisões no momento de identificar e resolver os problemas;

f) eficiência e eficácia na resolução dos problemas.

Em contrapartida, houve um resultado negativo que podê-se observar: dependendo do número de objetos gerenciados e do tempo entre uma coleta e outra por meio do *pooling*, gera um tráfego elevado de informações referentes ao gerenciamento, podendo causar congestionamento na rede. Devido a este fato, sugere-se que o monitoramento seja efetuado somente em objetos cuja importância para o funcionamento da rede seja maior que os demais, e o intervalo das coletas seja adequado ao número dos objetos gerenciados.

Durante o desenvolvimento, algumas dificuldades foram encontradas na aquisição do conhecimento, onde foi necessário compreender assuntos tanto da área gerenciamento de redes computadores quanto a de inteligência artificial, mais especificamente o raciocínio baseado em casos. Outra dificuldade encontrada foi a adaptação às linguagens e ferramentas para o desenvolvimento do sistema. Todas estas dificuldades foram superadas por meio da orientação de especialistas da área de gerência de redes e inteligência artificial, que com sabedoria, auxiliaram na definição e desenvolvimento das técnicas aqui utilizadas.

Para encerrar, o presente trabalho abre as portas para os trabalhos futuros, que podem ser aprofundados em áreas específicas como:

a) aplicação de novas técnicas de inteligência artificial;

b) utilização de novos objetos SNMP para realizar diferentes diagnósticos;

c) aplicação de gerenciamento em outras áreas funcionais;

d) utilização de novas tecnologias quanto a ferramentas de desenvolvimento; entre outros.

Além das possibilidades acima citadas, pode-se desenvolver outros trabalhos que venham a contribuir para o crescimento das áreas de gerenciamento de redes de computadores e inteligência artificial.

REFERÊNCIAS

AAMODT, A.; PLAZA, E. **Case-based reasoning:** foundational issues, methodological variations and system approaches**.** Artificial Intelligence Communications, Vol. 7, 1994.

BARRETO, Jorge Muniz. **Inteligência artificial no limiar do século XXI**. 3.ed Florianópolis: Duplic, 2001.

BITTENCOURT, Guilherme. **Inteligência artificial: ferramentas e teorias**. 3.ed. Florianópolis: UFSC, 2006.

BRAGA JÚNIOR, Mário de Sena. **Proposta de Modelo RBC para a Recuperação Inteligente de Jurisprudência na Justiça Federal.** Dissertação de Mestrado (Engenharia de Produção) Universidade Federal de Santa Catarina (UFSC), Florianópolis, 2001.

CASTELLS, Manuel. **A sociedade em rede**. 9.ed, rev. e ampl. São Paulo: Paz e Terra, 2006.

COSTA, Ernesto; SIMÕES, Anabela. **Inteligência artificial**: fundamentos e aplicações**.** Lisboa: FCA, 2004.

FARREL, Adrian. **A internet e seus protocolos**: **uma análise comparativa**. Rio de Janeiro: Elsevier, 2005.

FREITAS, Geraldo Magela Lopes de. **Uma Estratégia Para Implementação de Gerenciamento de Redes – Estudo de Caso do Tribunal de Contas da União.** Dissertação de Mestrado (Engenharia de Produção) Universidade Federal de Santa Catarina (UFSC), Florianópolis, 2001.

HOLANDA FILHO, Raimir. **SAGRES: Um Sistema Baseado em Conhecimento para Apoio à Gerência de Falhas em Redes de Computadores.** Dissertação de Mestrado (Ciência da Computação) Universidade Federal do Ceará (UFC), Fortaleza, 1998.

LEAKE, David. **Case-Based Reasoning:** Experiences, Lessons e Future Directions. **C**alifornia: AAAI Press/The MIT Press, 1996.

LEE, Rosina Weber. **Pesquisa Jurisprudencial Inteligente.** Tese de Doutorado (Engenharia de Produção) Universidade Federal de Santa Catarina (UFSC), Florianópolis, 1998.

LOPES, Raquel Vigolvino; SAUVÉ, Jacques Philippe; NICOLLETTI, Pedro Sérgio. **Melhores práticas para gerência de redes de computadores**. Rio de Janeiro: Campos, 2003.

KOLODNER, Janet. **Case-based reasoning.** Morgan Kauffman. 1993.

KUROSE, James F.; ROSS, Keith W. **Redes de computadores e a internet**: **uma abordagem top-down**. 3. ed São Paulo: Pearson Addison Wesley, 2006.

MAURO, Douglas R; SCHMIDT, Kevin J. **SNMP essencial**. São Paulo: Campus, 2001.

MELCHIORS, Cristina. **Raciocínio baseado em casos aplicado ao gerenciamento de falhas em redes de computadores.** Dissertação de Mestrado (Ciência da Computação) Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, 1999.

MICHAELIS: moderno dicionário da língua portuguesa. São Paulo: Melhoramentos, 1998.

PAL, Sankar K.; SHIU, Simon C. K. **Foundations** of **soft case-based reasoning**. New Jersey: Wiley-Interscience, 2004.

PERES, Sarajane Marques. **Raciocínio Baseado em Casos para Avaliação de Planos de Rotas.** Dissertação de Mestrado (Engenharia de Produção) Universidade Federal de Santa Catarina (UFSC), Florianópolis, 1999;

PETERSON, Larry L; DAVIE, Bruce S. **Redes de computadores** : **uma abordagem de sistemas**. Rio de Janeiro: Elsevier, 2004.

RAMOS, Alexandre Moraes. **Modelo para incorporar conhecimento baseado em experiências à arquitetura TMN.** Tese de Doutorado (Engenharia de Produção) Universidade Federal de Santa Catarina (UFSC), Florianópolis, 2000.

REZENDE, Solange Oliveira. **Sistemas inteligentes**: fundamentos e aplicações. Barueri, SP: Manole, 2005.

RIGANTI, Andréa. **Progetto e realizzazione di un sistema di monitoraggio per reti eterogenee basato su protocollo SNMP.** Dissertação de Mestrado (Engenharia de Telecomunicações) Universita di Pisa, Itália, 2005.

RUSSELL, Stuart J.; NORVIG, Peter. **Inteligência artificial**. Rio de Janeiro: Elsevier, 2004.

SANTOS, Fabio José Justo dos. **Sistema de gerenciamento de redes baseado em conhecimento.** Trabalho de Pós-Graduação (Administração em Redes LINUX) Universidade Federal de Lavras (UFLA), Lavras, 2004.

SILVA, João Ricardo Busi da**. A utilização de uma ferramenta da inteligência artificial aplicada a resolução de não conformidades do sistema de saída de emergência das edificações**. Dissertação de Mestrado (Engenharia de Produção) Universidade Federal de Santa Catarina (UFSC), Florianópolis, 2000.

SPECIALSKI, Elizabeth Sueli. **Modelo de informação baseado em relacionamentos entre objetos gerenciados para a gerência integrada de ambientes de telecomunicações.** Tese de Doutorado (Engenharia de Produção) Universidade Federal de Santa Catarina (UFSC), Florianópolis, 2000.

STALLINGS, William. **SNMP, SNMPv2, SNMPv3 and RMON 1 AND 2**. 3. ed. Reading: Addison-Wesley, 1999.

TANENBAUM, Andrew S. **Redes de computadores**. 4.ed Rio de Janeiro: Campus, 1997.

TANENBAUM, Andrew S. **Redes de computadores**. Rio de Janeiro: Campus, 2003.

VON WANGENHEIM, Christiane Gresse; VON WANGENHEIM, Aldo. **Raciocínio baseado em casos**. 1. ed. Barueri, SP: Manole, 2003.