UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

WATCHDOG
Um sistema de gerência de configuração em redes TCP/IP

por

Eduardo Carvalho de Souza Britto

Profa. Liane Tarouco
Orientadora

Porto Alegre,agosto de 1996

Agradecimentos

À professora Liane M. Rockenbach Tarouco pela orientação clara e objetiva que culminou neste trabalho.

Aos funcionários do ~~CPD~~ Grupo de Rede da UFRGS, especialmente ao Marcelo, Cleber e Gustavo pelo apoio oferecido nos momentos em que nada funcionava.

Aos amigos Pedro Pivotto e Vinícius Amaral pela força que me deram na Opus Dei nos momentos de maior aperto.

A minha tia Heloísa Barrili pela correção clara e precisa do trabalho final.

Aos meus pais pelos valores de vida que me passaram e me fizeram chegar até aqui.

A minha vó e irmã pelo carinho.

Ao Gervásio e ao Branco pelo apoio e incentivo.

A minha namorada Eunice pelo amor, incentivo e compreensão dados todos os dias de minha vida e que me dão forças para avançar mais e mais.

SUMÁRIO

Lista de Figuras

Lista de Tabelas

Resumo

A velocidade crescente com que aumentam o número de computadores interconectados entre locais distantes torna evidente a necessidade de uma gerência neste ambiente. Particularmente para redes com algumas centenas de máquinas de uma mesma empresa ou entidade, a falta de uma ferramenta gerência pode representar a perda total de controle sobre o que está ou não está conectado na rede num determinado momento e com que configuração.

O objetivo deste trabalho é criar um gerente de configuração que detecte as sub-redes e nodos que estão conectados a uma determinada rede definida pelo administrador e armazene a sua configuração. As informações a respeito de cada subrede serão coletados da MIB de cada nodo através do protocolo de gerenciamento SNMP. Serão coletadas dados sobre a configuração atual do nodo e de suas interfaces, dados esses presentes principalmente no grupo system, interfaces e ip da MIB. Além disso, dados a respeito da utilização dos nodos também serão coletados, criando um histórico em relação a cada dispositivo encontrado.

A fim de possiblitar um acesso fácil para o usuário, a interface do sistema é apresentada pelo WWW, permitindo assim que se consulte informações sobre essa rede de qualquer equipamento ou lugar. Pode-se encontrar um nodo através de uma consulta à rede a qual ele faz parte ou por uma pesquisa de nodos através das suas características. Para isso, os dados gerenciados serão armazenados num banco de dados que contém todas as informações coletadas sobre a rede.

Um sistema denominado WatchDog irá percorrer a rede e atualizar os dados armazenados, gerando um relatório com todas as modificações encontradas na rede em relação a última vez em que o sistema foi executado.

Abstract

The increasing velocity of the number of interconnected computers in a wide area makes evident the need of the enviroment management. In particular, in nets with a lot of machines in a same enterprise or entity, the lack of a management tool may represent the total lost of control of what is connected or not and of configuration.

The purpose of this work is to create a configuration manager that detects the subnets and nodes that are connected to an estabilished net, defined by the administrator and that stores its configuration. This information is about each subnet will be collected from MIB from every node through the SNMP management protocol. Data will be collected about the present node configuration and its interfaces, essentialy present in the system group, interfaces and IP of MIB. Besides information about the use of these nodes will be also collected creating a report about each device found.

In order to provide an easy access to the user, the interface system is presented by the web, thus permitting that the user consult information of this net from any hardware or place. It is possible to find a node through a consult to a private net to which it is attached by a research of nodes through theirs characteristics. For this, the managed information will be stored in a database, which contain all the information collected about this net.

A system called "WatchDog"will visit the net and to update the information stored, thus making a report with all the modifications found in the net in relation to the last time this system has been executed.

1. Introdução

A rapidez com que ocorre hoje o crescimento constante da tecnologia tem causado um efeito interessante : o obsolescência rápida dos equipamentos considerados modernos a pouco tempo atrás e a sua conseqüente queda de preço. Esse fenômeno torna a compra de computadores e produtos eletrônicos cada vez mais facilitada e abrangente. Dentro das empresas, esse efeito se reflete num crescimento na aquisição de computadores, que buscam a automatização de seus serviços e a informatização das suas atividades.

Esse crescimento dentro das empresas gerou num primeiro momento uma fragmentação departamental das informações, fazendo com que cada departamento possuísse acesso somente a aquelas utilizadas nas suas atividades, sem compartilhar essas informações com outros departamentos ou com os próprios computadores espalhados nas suas dependências. Como conseqüência direta desta situação, começaram a existir dados de um mesmo cadastro que eram necessários em mais de um departamento, e portanto se repetiam por diversas máquinas. Além disso, a falta de compartilhamento de dados gerava também a impossibilidade de se visualizar alguma determinada informações que, ao passar do escopo departamental, exigi-se uma busca em vários departamentos da empresa.

Para resolver a problemática da troca de dados entre os computadores, surgiram as redes locais . Através delas os computadores podem trocar informações de forma eficiente, rápida e organizada, ao mesmo tempo em que elas permitem eliminar uma grande quantidade de informações redundantes, uma vez que um determinado repositório de dados pode ser compartilhado por vários usuários.

Com o barateamento do hardware e a sua conseqüente popularização, o parque de máquinas conectados nesta rede subiu vertiginosamente, aumentando assim o acesso às informações e aplicativos da empresa. Esse crescimento de hardware gerou um aumento direto na criação de softwares, que se tornaram cada vez mais especializados às atividades de cada empresa e seus departamentos.

A medida em que os sistemas foram se especializando, foi aumentando a sua importância dentro da empresa, chegando ao ponto de atualmente todas as informações e documentos serem guardados no computador, tornando o funcionamento do sistema um fator crítico para as atividades da empresa.

Paralelamente ao crescimento das redes, aumentaram também o número de componentes passíveis de serem conectados numa rede (hosts, roteadores, pontes, impressoras, ...), bem como o número de empresas fabricantes destes componentes, de modo que as redes passaram a ser formadas por dispositivos diferentes, com funcionalidades, protocolos de comunicação e fabricantes distintos. Toda essa diferença gerou uma grande dificuldade em compatibilizar e unir os diversos equipamentos dentro da mesma rede.

Além disso, o crescimento das redes locais dentro das organizações gerou uma necessidade de troca de informações entre redes distintas, tais como redes de departamentos diferentes dentro da empresa, ou redes de filiais localizadas em lugares separados. Logo as empresas passaram a se conectar a nível organizacional, passando rapidamente para uma necessidade de comunicação regional e em seguida nacional, culminando na ligação dos computadores do mundo inteiro através da internet. Toda essa interligação fez com que o tráfego das redes aumentasse consideravelmente, decrementando proporcionalmente a performance da comunicação entre as redes. Surgiu daí a necessidade de se otimizar o uso dos recursos da rede bem como a comunicação entre eles.

Dentro deste cenário surgiu o conceito de **gerenciamento da rede**. Segundo [FAN 93], gerenciamento de rede é o processo de controlar uma rede complexa de forma a maximizar a sua eficiência e utilidade. Já segundo a ISO, o gerenciamento de redes provê mecanismos para a monitorização, controle e coordenação de recursos em um ambiente OSI e define padrões de protocolo OSI para a troca de informações entre estes recursos.

Dependendo das capacidades do sistema que administra a rede, o processo de gerenciamento normalmente possui as fases de :

a) Coleta de dados,

b) Processamento dos dados coletados e seu armazenamento, e

c) Apresentação destes dados para análise.

Esse processo pode funcionar de dois modos distintos. No primeiro, o acompanhamento da rede é feito de maneira on-line, permitindo ações imediatas que garantam o funcionamento do sistema. No segundo, a análise é off-line, servindo para verificar como foi o desempenho da rede durante um certo período bem como as causas possíveis para os problemas que ocorreram dentro deste período.

De modo a englobar toda a sua funcionalidade, a gerência de rede foi dividida pela ISO em cinco categorias :

a) Gerenciamento de Falha,

b) Gerenciamento de Segurança,

c) Gerenciamento de Performace e

d) Gerenciamento de Contabilização.

e) Gerenciamento de Configuração,

O gerenciamento de falhas consiste no processo de localizar os problemas que ocorreram na rede e corrigi-los. A resolução deste problema se dá através da sua descoberta, isolamento e finalmente resolução. Ela provê um conjunto de ferramentas que mostram ao administrador da rede o estado corrente da rede, apontando onde provavelmente o problema ocorreu e gerando relatórios que viabilizam a resolução rápida do problema.

O gerenciamento de segurança consiste no uso de técnicas para monitorar e controlar mecanismos de rede. Esses mecanismos podem ser tanto a nível de controle de acesso aos computadores e sistemas que rodam sobre eles, como a nível de controle de informações de carater sigiloso dentro da empresa e que só podem ser acessados por alguns usuários. Ele consiste na identificação da informação sigilosa a ser protegida, da busca dos seus pontos de acesso e da manutenção da segurança nestes pontos. Essa proteção é efetivada através do limite de acesso aos hosts e dispositivos da rede pelos usuários e pela notificação ao gerente da rede sobre alguma eventual quebra de segurança.

O gerenciamento de performance ou gerenciamento de desempenho consiste na tentativa de assegurar que a ligação entre os computadores de uma rede ou entre as redes permaneça acessível e desocupada para o seu uso eficiente. Ele assegura essa ligação através da monitorização aos dispositivos de rede e de suas ligações associadas, a fim de determinar a sua taxa de utilização e erro, bem como para ajudar a rede a prover um nível consistente de serviços aos usuários, garantindo que a capacidade dos dispositivos e dos links não seja estravazada de forma a não prejudicar a performance da rede. Os dados obtidos podem além de ajudar a isolar imediatamente os componentes que estão sendo utilizados com uma carga excessiva, também responder aos problemas potenciais tais como o porquê da demora no retorno da resposta de um servidor de banco de dados. Esse controle consiste em coletar os dados de utilização e ligações dos dispositivos, analisar os dados relevantes, verificar os pontos de maior

utilização e realizar simulações a fim de determinar como a rede pode ser alterada de forma a maximizar a performance.

O gerenciamento de contabilização consiste no processo de coleta de informações estatísticas sobre a rede. Ele envolve a mensuração do uso da rede pelos usuários de modo a estabelecer métricas, checagem das cotas e determinação dos custos e gastos de cada usuário. A utilização destas informações ajuda ao administrador a tomar decisões sobre a alocação dos recursos da rede, tornando-se indispensável na avaliação de recursos críticos tais como o espaço em disco e o poder de processamento utilizado.

O gerenciamento de configuração é processo de busca e configuração de dispositivos de vital importância, denominados críticos, para o funcionamento da rede. Ele dá ao administrador o controle sobre toda a configuração de cada equipamento conectado a rede, oferecendo uma fonte rápida de consulta e acesso a dados vitais de configuração destes dispositivos. Além disso, ele permite que se faça consultas sobre características da rede, tais como quantos dispositivos do tipo X existem presentes na rede hoje ou em quantas máquinas está instalado o software Y.

Dentre os tipos de gerenciamento descritos acima, o de configuração será o objeto de estudo e implementação neste trabalho.

O gerenciamento de configuração, como explicado anteriormente, é o processo de obtenção de dados sobre a rede e o uso destes dados para gerenciar a configuração de todos os dispositivos da rede. Por exemplo, suponha que se deseje colocar um novo computador na rede XYZ da empresa. Antes de colocá-lo, é interessante verificar quantos computadores esta rede já possui e que computadores são esses, a fim de verificar se a inclusão de um novo computador daquele modelo é necessário naquela rede. Através do gerenciamento de configuração é possível não somente descobrir estes dados, como também obter o número de um endereço de rede que este computador pode receber e, se necessário, mudar o seu endereço de rede a partir de qualquer outra estação conectada.

Além disso, o gerenciamento de configuração provê um inventário dos dispositivos conectados a rede. Através deste inventário é possível obter-se informações sobre os componentes de toda a rede, tais como : a quantidade de impressoras que existem atualmente na rede e o local onde elas se encontram, quais são os computadores que estão conectados a mais de uma rede, quais são gateways e quais são hosts, etc.

O gerenciamento de configuração consiste nos seguintes passos :

a) Obtenção de informações sobre o ambiente atual da rede,

b) Uso destes dados para modificar a configuração dos dispositivos da rede, e

c) Armazenamento dos dados, mantendo um inventário de todos os componentes da rede e produzindo um conjunto de relatórios sobre eles.

O primeiro passo pode ser feito tanto de maneira manual como automática. Da maneira manual, o administrador deve se logar em cada máquina, obter as informações e registrá-las no gerente. Esse tipo de prática pode ser dificultada em redes que possuam uma quantidade significativa de dispositivos, além de ter que ser repetida a cada mudança de configuração de um dispositivo da rede. Novos equipamento conectados a rede e que não foram comunicados ao gerente também ficarão sem ser registrados.

A segunda maneira é de forma automática, onde os processos que são executados regularmente obtém os dados e registram esses dados no inventário do gerente. Outra opção é a de auto-descoberta, onde o gerente varre a rede e vai buscando todos os equipamentos conectados a ela. O grande problema dos procedimentos

automáticos dizem respeito a quantidade de tráfego gerada pelos mesmos na rede, reduzindo a performance dos sistemas.

Obtidos e armazenados os dados sobre a rede, esses dados podem ser modificados através do gerente. Para garantir que não haja inconsistência entre os dados armazenados e os da configuração atual do dispositivo devido a condição de concorrência no acesso aos dados, estes são modificados no inventário antes de ser enviada a modificação para o dispositivo, permitindo a monitorização desta modificação a fim de verificar se ela ocorreu com sucesso ou não.

O armazenamento de informações normalmente coloca os dados do dispositivo num computador central que permite tanto um acesso rápido aos gerentes da rede como a seus administradores. Este pode ser feito tanto com arquivos do tipo ASCII como com sistemas gerenciadores de banco de dados (SGBD). O ASCII é fácil de ser lido por vários programas e é armazenado de maneira simples, mas possui desvantagens significativas tais como a quantidade de espaço em disco que essa maneira simples gasta e a dificuldade de se realizar pesquisas sobre arquivos ASCII.

Já o SGBD guardam os dados eficientemente, possibilitando que um bom volume deles fiquem armazenados num só computador. Os dados são armazenados num formato próprio, permitindo uma pesquisa rápida sobre eles. Em compensação, ele envolve um conjunto de procedimentos complexos de administração, que utiliza uma linguagem própria (esta desvantagem está se tornando cada vez menor através da padronização do SQL como linguagem de consulta a bancos de dados), e normalmente limita o seu funcionamento somente a plataforma onde ele se encontra instalado, não permitindo a migração para outras plataformas.

A fim de resolver os problemas de detecção dos dispositivos presentes na rede e o controle sobre a sua configuração, este trabalho propõe uma ferramenta de gerência de configuração para redes do tipo TCP/IP, WatchDog, que permitirá a detecção dos dispositivos conectados à rede e gerenciará um banco de dados com todas as suas principais informações em termos de configuração. O WatchDog utilizará o banco de dados Postgres para gerenciar os seus dados e usará o pacote de SNMP da CMU(Carnegie-Mellon University), sendo dividido em dois processos com funcionalidades distintas : um processo que servirá como interface para consulta dos dados armazenados e outro para detecção dos dispositivos presentes na rede.

A Interface será implementada em WWW, permitindo deste modo o seu acesso de qualquer máquina que esteja conectada à internet, de qualquer lugar do mundo. Essa interface permitirá não somente a consulta e navegação sobre os dados que se encontram armazenados, como também a elaboração de pesquisas mais complexas sobre a rede, de modo a encontrar hosts e/ou sub-redes que atendam a determinada condição especificada pelo usuário.

O capítulo 2 apresenta conceitos sobre o gerenciamento de redes, desenvolvendo o padrão de gerenciamento na internet e fazendo uma breve descrição sobre o gerenciamento OSI.

O capítulo 3 trata das ferramentas e ambientes utilizados para desenvolver o gerente de configuração, mostrando as suas características, vantagens e desvantagens.

O capítulo 4 inicia um estudo sobre a implementação do gerente, através da escolha dos objetos que serão gerenciados bem como a sua modelagem lógica e física no banco de dados.

O capítulo 5 explicita como foi feita a implementação do WatchDog, descrevendo cada processo detalhadamente e explicando como se chegou ao algoritmo de detecção de dispositivos.

O capítulo 6 por último faz uma conclusão a respeito do trabalho e propõem futuras extensões.

2. Gerenciamento de Redes na Internet

O gerenciamento está presente na maioria da grandes redes WAN desde o seu surgimento. Implementados através de protocolos, eles costumavam rodar inicialmente em redes homogêneas e funcionavam em camadas de baixo nível, normalmente a nível de enlace, tornando a sua administração razoavelmente fácil.

Diferente destas WANs, o gerenciamento na internet é uma tarefa bem mais complexa, uma vez que ela consiste de múltiplas redes heterogêneas conectadas por gateways IP. Um nodo gerente nestas condições irá controlar gateways heterogêneos que podem ou não possuir o mesmo protocolo a nível de enlace. Além disso, os gateways gerenciados podem se encontrar em qualquer parte do mundo, tornando difícil a comunicação entre os nodos, a menos que se utilize uma conexão a nível de transporte.

Devido a todas essas razões, os protocolos de gerenciamento na internet são implementados a nível de aplicação e utilizam o TCP/IP para transportar as suas mensagens. Uma vez que esse protocolo foi projetado sem levar em consideração a plataforma de hardware, ele pode ser utilizado por qualquer tipo de gateway presente na internet. Isso dá uniformidade também entre as plataformas : uma vez que todos os gateways entendem e respondem ao protocolo seguindo o mesmo padrão, todos os gateways utilizam o mesmo conjunto de comandos.

O modelo de gerenciamento da rede é composto por :

- Nodos gerenciados, cada um possuindo um agente. Estes nodos podem ser tantos hosts comuns como gateways ou dispositivos de interligação como hubs, pontes ou multiplexadores.
- Estações gerenciadoras, que são hosts que estão rodando o protocolo de gerenciamento e aplicações de gerenciamento, e
- Protocolo de gerenciamento, que define como são realizadas as interações entre as estações gerenciadoras e gerenciadas. Na internet, os protocolos utilizam um paradigma de controle remoto dos nodos, onde cada nodo é visto pelo gerente como um conjunto de variáveis. A monitoração destes nodos é feita através da leitura destas variáveis e o controle sobre eles é exercido através das modificações dos valores destas variáveis. Desta forma, através de apenas dois comandos o gerente consegue gerenciar todos os dispositivos heterogêneos da rede, uma vez que todos seguem o mesmo padrão.

Além das duas operações acima(leitura e modificação), existe ainda um mecanismo de interrupção nos nodos gerenciados conhecido como trap, que envia notificações ao gerente quando ocorrem eventos extraordinários. Estas notificações servem apenas de alerta ao gerente, carregando apenas as informações indispensáveis para o mesmo, que fica responsável em consultar o nodo a fim de obter dados mais completos, conforme necessário. Esse mecanismo faz com que a notificação de eventos seja enviada na mesma hora da sua ocorrência sem que seja gerada uma carga excessiva na rede.

Do mesmo modo que as operações, os dados que são armazenados por cada dispositivo também devem seguir um padrão, de modo que se possa operar em ambientes heterogêneos. Mais do que ser acessível em qualquer plataforma de hardware, esses dados devem ser independentes do protocolo de gerenciamento utilizado, para que qualquer fabricante ou usuário de software possam utilizar os mesmos dados sem ter que se atrelar a um determinado protocolo.

Desta forma surgiu a MIB (Management Information Base), um padrão que especifica quais ítens de dados devem ser armazenados pelo host ou gateways e quais

operações são permitidas para cada um. A MIB divide os seus objetos em oito categorias, a saber :

a) system, que fornece informações gerais sobre o host ou gateway,

b) interfaces, com informações sobre as interfaces do dispositivo,

c) address translation, com mapeamento de endereços IP para endereços físicos,

d) ip, com informações sobre o software de protocolo da internet,

e) icmp, com informações do software de ICMP(Internet Control Message Protocol)

f) tcp, com informações sobre o protocolo TCP(Transmission Control Protocol)

g) udp, com informações sobre o protocolo UDP(user datagram protocol)

h) egp, com informações sobre EGP(Exterior Gateway Protocol).

Ela está definida através de uma especificação chamada de SMI (Structure of Management Informantion), estrutura de gerenciamento da informação, onde estão estabelecidas as regras utilizadas para definir e identificar as suas variáveis. De forma a tornar a MIB simples, a SMI criou restrições no tipo das variáveis permitidas na MIB, especificando regras para nomear estas variáveis e criando tipos para defini-las. Assim, foram criados tipos de variáveis que representam tipos de dados comums, tais como o ipAddress para representar endereços e Counter para representar contadores.

Todas as variáveis são definidas e referenciadas utilizando o ASN.1 (Abstract Syntax Notation) criado pela ISO. O ASN.1 é uma linguagem formal que possui duas características : utiliza a notação usada em documentos que os homens lêem, e possui uma representação compacta da mesma informação que é utilizada pelos protocolos. Além disso, ela é uma notação extremamente precisa, tirando qualquer ambigüidade entre a representação e o seu significado.

Os nomes utilizados pela MIB fazem parte de uma identificação de objetos administrada pela ISO. A identificação dos objetos é feita hierarquicamente, ou seja, existe uma relação de composição entre os objetos, onde a referência ao nome de um objeto é feita através de um nome único global composto por todos os objetos hierarquicamente mais elevados. Cada objeto é numerado dentro do seu nível hierárquico, podendo assim ser referenciado tanto pelo seu nome como pelos seus números. Desta forma, a MIB pode ser referenciada como iso.org.dod.internet.mgmt.mib ou como 1.3.6.1.2.1 (Figura 2.1). Do mesmo modo, um objeto da MIB pode ser referenciado por qualquer uma das duas maneiras.

Quando os protocolos de gerenciamento de rede utilizam variáveis da MIB em suas mensagens, cada nome ganha um sufixo na hora de ser referenciado. No caso de variáveis simples, esse sufixo é formado pela simples adição do 0. Assim, a representação do objeto ipForwarding é feita pelo número 1.3.6.1.2.1.4.1.0, ou resumidamente como *mib.ip.ipForwarding.0* . Já no caso de variáveis complexas, tais como variáveis que fazem parte de um array, por exemplo, o sufixo é o responsável por diferenciar cada elemento do array. Assim, no caso da tabela de endereços IP do dispositivo, o sufixo padrão consiste no endereço IP da entrada na tabela. Dessa forma, se for necessário representar o objeto ipAdEntNetMask cuja entrada corresponda ao número ip 143.54.1.20, ele seria referenciado por :

iso.org.dod.internet.mgmt.mib.ip.ipAddrTabel.ipAddrEntry.ipAdEntNetMask.143.54.1.20,

ou através do seu número

1.3.6.1.2.1.4.20.1.3.143.54.1.20.

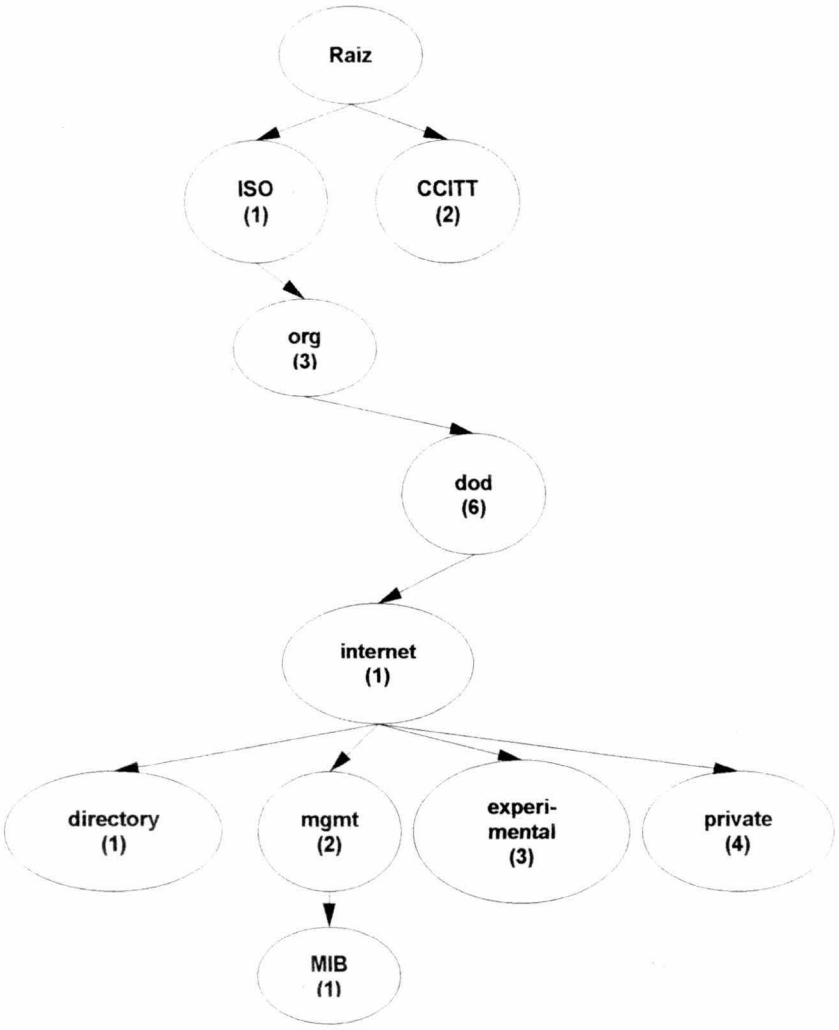

Figura 2.1 : Identificação da MIB pela ISO

Por ter um ambiente altamente heterogêneo, muitas vezes dispositivos interligados a internet não suportam o seu padrão de gerenciamento e se tornam por isso incomunicáveis. São exemplos típicos destes dispositivos repetidores e pontes, que não têm capacidade para implementar este tipo de protocolo, assim como hosts ou gateways que estão conectados a rede mas possuem outro padrão de protocolo. Visando realizar o gerenciamento sobre estes "dispositivos estrangeiros", existem agentes proxy que servem como intermediadores entre os nodos e os gerenciadores. Esses agentes tornam-se responsáveis por realizar a conversão entre os protocolos. Além disso, eles possuem também a função de armazenar as informações mais solicitadas, de modo a reduzir o tráfego da rede.

Deste modo, um gerente que deseje se comunicar com um dispositivo estrangeiro irá submeter a sua operação ao agente proxy, que converterá o seu pedido para o protocolo utilizado pelo nodo gerenciado e receberá a sua resposta, convertendo-a novamente para o nodo gerente.

Dentro do gerenciamento internet, os padrões que mais se destacam são o SNMP e o CMOT. O SNMP foi um protocolo estratégico criado a curto prazo que evoluiu rapidamente, tornando-se o padrão de facto utilizado pelo mercado. Porém o SNMP possui um conjunto limitado de operações voltados para os aspectos físicos da rede, tais como pontes, roteadores e hubs. A criação de extensões ao SNMP para

solucionar este problema começou a gerar uma falta de integração entre os sistemas. A fim de solucionar as limitações e conseqüentemente a falta de um padrão, a arquitetura de gerenciamento do padrão OSI foi mapeada sobre o TCP, gerando assim o protocolo CMOT.

A seguir será analisado o SNMP e estudado alguns conceitos empregados no gerenciamento OSI de redes.

2.1 SNMP

O SNMP é um protocolo de gerenciamento de redes TCP/IP que obteve desde a sua criação uma grande popularidade, tornando-se o padrão de fato para redes TCP/IP. Suas principais características são a sua simplicidade e a sua baixa exigência de recursos para que ele execute eficientemente.

Ele é composto por quatro operações :

a) **get**, que é utilizado para recuperar uma informação específica de gerenciamento,

b) **get-next**, que é utilizado para percorrer transversalmente as base de informação de gerenciamento,

c) **set**, que é utilizado para atribuir valores para as variáveis de gerenciamento, e

d) **trap**, que é utilizado para relatar a ocorrência de eventos extraordinários.

A comunicação entre os dispositivos é feita através do conceito de comunidade, que define o relacionamento que existe entre um agente SNMP e um ou mais gerentes SNMP. Assim, quando dispositivos interligados na rede desejam trocar dados entre si, eles o fazem enviando mensagens que contém o nome da comunidade destino, que possui ainda algumas informações adicionais que validam o remetente como sendo membro daquela entidade, e os dados a serem enviados, que são compostos por operações e as suas operações associadas.

O mecanismo de autenticação utilizado pelo SNMP é extremamente trivial. Ele parte do princípio que se a comunidade utilizada é uma das comunidades conhecidas pelo dispositivo que está recebendo a mensagem, então o dispositivo remetente é considerado como sendo membro daquela comunidade.

O SNMP é implementado através de um protocolo assíncrono de pedido/resposta, ou seja, uma entidade SNMP que realiza um pedido não precisa ficar aguardando a chegada da sua resposta, podendo se ocupar de outras tarefas. Existem quatro interações primitivas do protocolo(Figura 2.2) :

1. O gerente obtém informações de gerenciamento do seu agente(a),

2. O gerente realiza uma consulta transversal à próxima variável do agente(b),

3. O gerente manipula algum dos atributos do agente(c), e

4. O agente envia um evento extraordinário (trap) para o gerente(d).

As mensagens trocadas pelas entidades SNMP são definidas em ASN.1 . Ao contrário do que ocorre com os protocolos TCP/IP, as suas mensagens não possuem um conjunto fixo de campos, consistindo basicamente de três partes : o número da versão do protocolo, um identificador do agente SNMP e uma área de dados. A área de dados é dividida em PDUs (Protocol Data Units). Cada PDU consiste ou num pedido feito pelo cliente ou numa resposta enviada pelo servidor. Existem 5 tipos diferentes de PDUs : GetRequest-PDU, GetNextRequest-PDU, GetResponse-PDU, SetRequest-PDU e Trap-PDU. A figura 2.3 mostra a definição de uma mensagem SNMP e de um PDU descritos em ASN.1 .

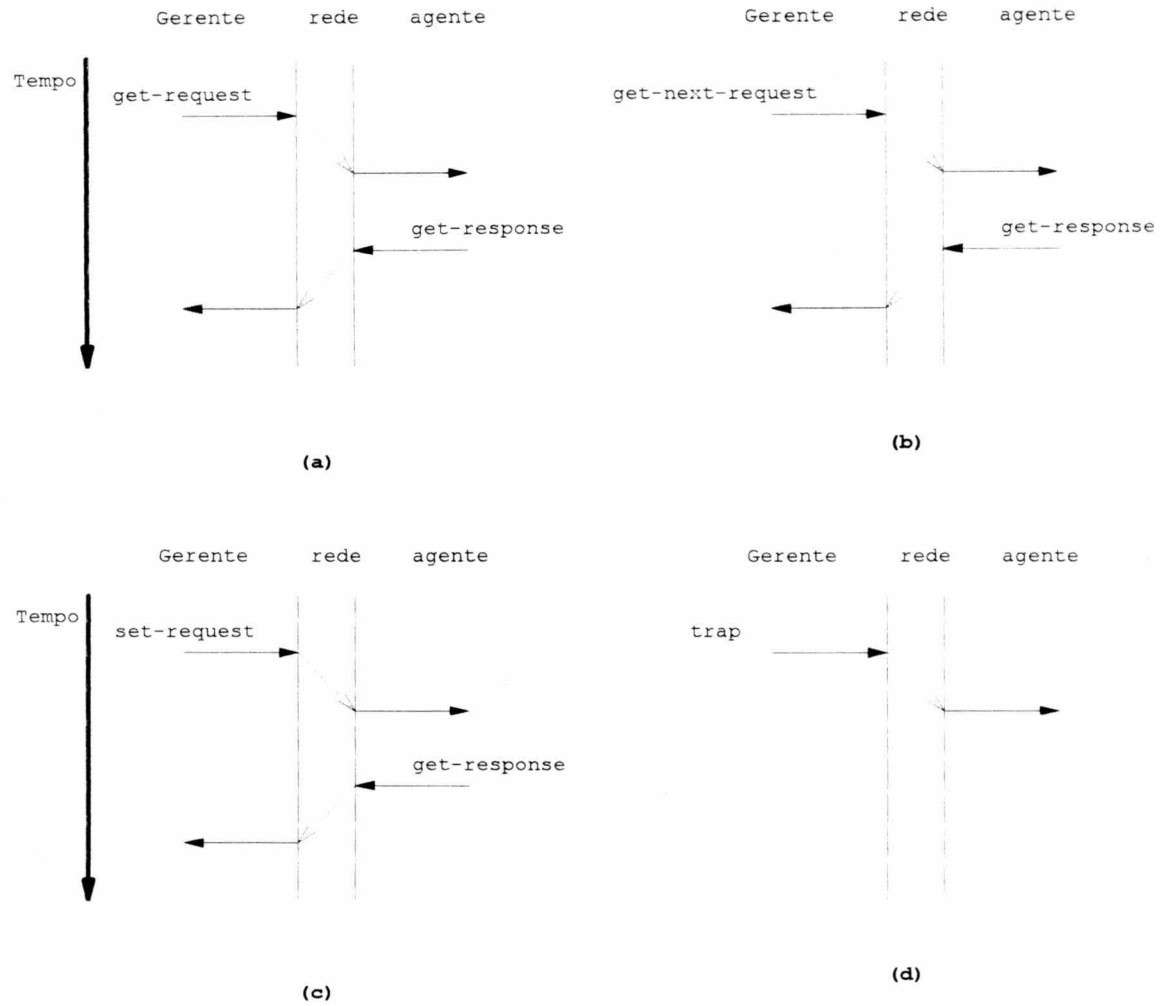

Figura 2.2 : Interações primitivas no protocolo SNMP.

```
SNMP-Message ::=
  SEQUENCE {
      version INTEGER {
          version-1 (0)
      },
      community
          OCTET STRING,
      data
          ANY
  }
```

(a)

```
SNMP-PDUs ::=
  CHOICE {
      get-request
          GetRequest-PDU,
      get-next-request-PDU
          GetNextRequest-PDU,
      get-response
          GetResponse-PDU,
      set-request
          SetRequest-PDU,
      trap
          Trap-PDU
  }
```

(b)

Figura 2.3 : O formato de uma mensagem SNMP e a definição de uma SNMP-PDU em ASN.1 .

2.2 Considerações sobre o Gerenciamento OSI

O gerenciamento OSI classifica as suas áreas de atuação em gerenciamento de Falhas, gerenciamento de configuração, gerenciamento de contabilização, gerenciamento de desempenho e gerenciamento de segurança, todas já conceituadas na primeira parte deste trabalho.

A arquitetura de gerenciamento possui três elementos básicos :

- objetos gerenciados : recursos do sistema que estão sujeitos a serem gerenciados. São definidos em cima das operações que podem ser executadas sobre eles, dos seus atributos, das notificações que podem ser emitidas e de suas relações com outros objetos gerenciados.
- agente : elemento que se encontra entre os objetos e o gerente, ele executa operações de gerência sobre os objetos e transmite notificações emitidas pelo objeto ao gerente.
- gerente : sistema responsável por controlar os objetos gerenciados e obter informações suas, através dos agentes.

Ele é estruturado em três tipos de gerenciamentos :

a) Gerenciamento de sistemas : gerencia todo o sistema como um todo, desde o sistema local até o meio físico, os protocolos e os dispositivos associados. Exige funções de apoio nos sete níveis da camada OSI.

b) Gerenciamento de Camada : diz respeito a gerência de objetos que estão relacionados com a mesma camada, utilizando funções internas de apoio e protocolos especiais. Não prestam serviço para as camadas superiores, tornando-se independentes do protocolo das outras camadas.

c) Operação de Camada : Gerencia uma única instância de comunicação numa determinada camada, utilizando o protocolo normal de cada camada para realizar as trocas de informações entre elas.

Cada camada do modelo OSI possui uma interface específica através das Entidades de Gerenciamento de Camadas (LME - Layer Management Entities), concentrando em cada uma as funcionalidades de sua camada. A entidade de aplicação de gerenciamento de sistema(SMAE - System Mangement Application Entity) disponibiliza a interface entre as camadas e entre o gerente, bem como a interface entre as camadas de mesmo nível dos nós que estão se comunicando. Essa interface entre nós é feita através do Protocolo de Informação de Gerenciamento Comum (CMIP - Common Management Infomation Protocol). Dentro desse contexto, as operações utilizadas para realizar o gerenciamento em todas as camadas é feito via MIB.

A entidade de aplicação de gerenciamento de sistema(SMAE) utiliza uma série de serviços para realizar seu serviço de interface, entre eles têm-se :

- SMASE (System Management Application Service Element) - Elemento de serviço de aplicação de gerenciamento, que define a forma como será transferida a informação entre os SMAEs.
- ACSE (Association Control Service Element) - Elemento de serviço de controle de associação, responsável por estabelecer a conexão e a liberação das associações entre diferentes SMAEs.
- CMISE (Common Management Information Service Element) - presta serviço ao SMASE, especificando serviços e procedimentos usados para transferência de dados entre as operações de gerenciamento. Utiliza serviços do ACSE e do ROSE.
- ROSE (Remote Operation Services Element) - responsável por fornecer serviços que disponibilizem operações em um sistema remoto.

3. Ferramentas Utilizadas

No capítulo a seguir será dada uma breve descrição a respeito das ferramentas e meios que foram utilizados para implementar o WatchDog, juntamente com as suas principais características, potencialidades, funcionalidades e formas de utilização.

3.1 Sistema Gerenciador de Banco de Dados PostGres

O Postgres é um SGBD relacional que extendeu a sua DDL para incorporar algumas construções básicas da orientação a objetos, de forma a aumentar o poder da sua linguagem e da manipulação dos dados armazenados.

Ele começou a ser esboçado em 1986, e durante o ano de 1987 teve definido o seu modelo de dados inicial, as suas regras, arquiteturas e gerência de armazenamento. Em 1989 foi lançada a sua primeira versão, e desde então surgiram diversas novas versões, criando novas características e aumentando a sua portabilidade. Hoje ele se encontra na versão 4.2, estando disponíveis em diversas plataformas e sistemas operacionais tais como a DECstation 3000(OSF/1 1.3, 2.0), Sun4(Sun OS 4.1.3, Solaris 2.3), HP-9000(HP-UX 9.0) e IBM RS/6000(AIX 3.2.5). Ele é um software de domínio público e em 1994 já estava presente em mais de 600 sites pelo mundo inteiro.

Uma das grandes vantagens do Postgres é a sua interface amigável com a linguagem C. Através de uma rápida leitura a um de seus manuais, é possível construir facilmente aplicações em C que se comunicam com o Postgres através do uso de algumas funções da sua biblioteca LIBPQ. Esse mecanismos permite o envio de requisições de consulta ao gerenciador de banco de dados, a obtenção do seu resultado e a atualização dos dados armazenados.

Tudo isso é possível devido ao fato do Postgres usar um modelo cliente/servidor com um processo por usuário. Uma sessão do Postgres consiste em três processos cooperativos Unix :

a) Um processo supervisor, chamado postmaster,

b) A aplicação do cliente que deseja realizar consultas/atualizações no banco de dados, e

c) O servidor de banco de dados, que é o próprio Postgres.

O processo postmaster gerencia um conjunto de banco de dados que se encontra numa máquina da rede. Ele funciona como intermediador entre os clientes do Postgres e o servidor de banco de dados. Para tanto, esse processo fica sempre executando, escutando a porta da máquina onde ele está rodando à espera de pedidos de conexão. O estabelecimento de uma conexão entre o cliente e o servidor se dá da seguinte maneira(Figura 3.1) :

1. As aplicações clientes, ao desejarem acessar algum banco de dados Postgres, fazem chamadas a funções da biblioteca LIBPQ.

2. Estas funções enviam a requisição do cliente para o processo postmaster.

3. Este, por sua vez, cria um processo backend que fará o papel de servidor de banco de dados somente para o cliente que fez a requisição.

4. O postmaster conecta o cliente ao servidor criado. A partir daí, o cliente e o servidor se comunicam sem a intervenção do postmaster.

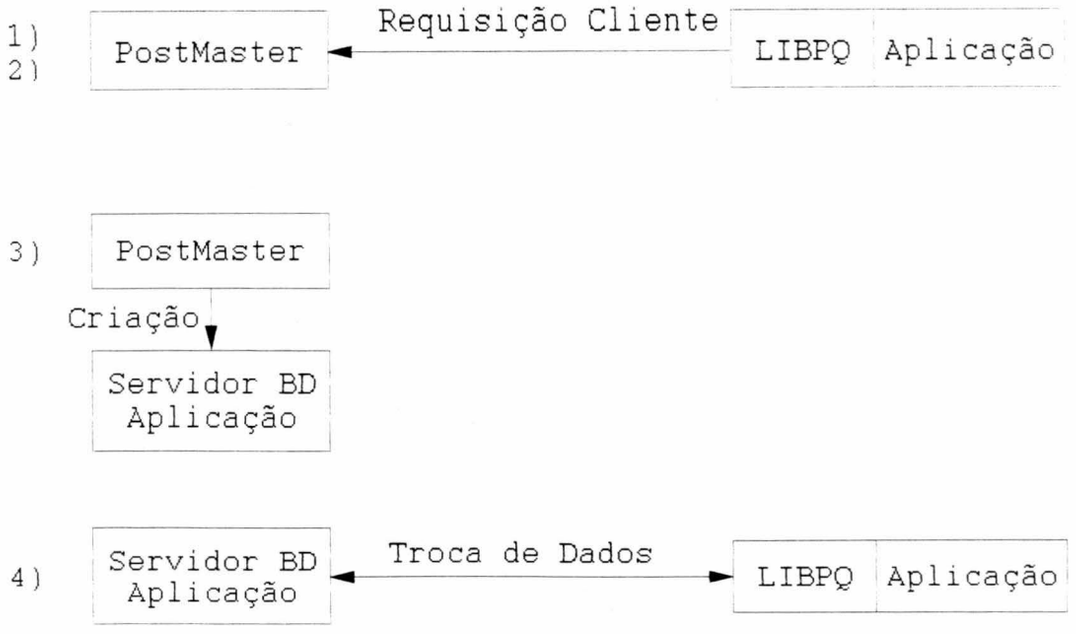

Figura 3.1 : Estabelecimento de conexão entre um processo cliente e o Postgres.

Uma das grandes vantagens desta arquitetura é que, enquanto o processo postmaster e o processo servidor rodam sempre na mesma máquina, o cliente pode estar sendo executado em qualquer outra máquina que tenha acesso ao nodo que está rodando o postmaster.

Devido a orientação a objetos do Postgres, uma entidade dentro do sistema é chamada de classe. A classe é passível de sofrer todas as operações que uma entidade admite no modelo relacional, através de uma linguagem chamada PostQuel, que lembra em muitos aspectos o SQL. A sua sintaxe é praticamente a mesma, só modificando o nome dos comandos, tais como o retrieve (select), append (insert), delete e replace (update).

Além destas características padrões do modelo relacional, ele possui algumas características extras tais como :

- O conceito de herança da orientação a objetos, herdando porém somente os dados das classes, uma vez que elas não possuem métodos.
- Atributos de classe do tipo Array, que podem ter várias dimensões e podem assumir um tamanho variável,
- Atributos do tipo Conjunto, onde o atributo é formado por um número variável de elementos de outra classe, e
- A definição de funções dentro da linguagem, com características de passagem de parâmetros e retorno de diversos tipos.

O usuário dentro do postgres é identificado pelo seu username no sistema operacional. Assim, o acesso ao postgres só é permitido para aqueles que possuem o nome do seu usuário Unix cadastrado no Postgres. Isso faz com que não seja necessário realizar nenhum pedido de senha para se conectar ao Postgres. Este mecanismo parte do princípio de que, uma vez que o Unix permitiu que se entrasse numa conta de usuário, a pessoa que está utilizando aquele usuário possui permissão para alterar os seus dados. Dentro de suas classes existe o conceito de direito de uso sobre as mesmas. Um usuário só pode acessar as classes que foram criadas por ele ou que tiveram os seus direitos de acesso liberados por seus criadores. Uma classe pode ter direito de leitura, modificação e definição de regras.

3.2 SNMP/CMU

O pacote SNMP da CMU visa o desenvolvimento de aplicações que necessitam realizar operações SNMP. Seu código foi escrito visando a sua portabilidade para a maioria das plataformas, sendo suportado atualmente na maioria dos ambientes que possuem interface para os sockets de Berkley, tais como os PCs rodando ACIS versão 3, SUN 3/60 utilizando SUNOS 3.5, DEC Microvax utilizando Ultrix 3.3 e DECStation 3100's utilizando Ultrix 3.0 .

A versão 2.0 utilizada inclui uma biblioteca para desenvolvimento, um conjunto de aplicações clientes e uma documentação de suporte.

As funções básicas disponibilizadas dão suporte a criação de conexões em SNMP, envio e recebimento de mensagens e desconexão. Para isso, a biblioteca utiliza um mecanismo baseado no preenchimento de estruturas de dados, onde elas são preenchidas e submetidas a funções, que agirão dependendo da forma como foram preenchidos os campos destas estruturas de dados.

Para implementar essas funções, o pacote utiliza uma série de funções separadas por arquivo fonte, tais como funções para leitura do padrão ASN.1, para leitura de objetos da mib, etc... Os objetos da MIB são definidos num arquivo mib.txt, que deve estar sempre presente no diretório em que roda a aplicação. A utilização deste arquivo torna a inclusão de novos objetos na MIB conhecida pelo pacote uma tarefa simples, bastando para isso colocar uma nova definição em ASN.1 neste arquivo.

Além disso, um conjunto de aplicações prontas são disponibilizadas pelo sistema. Essas aplicações foram desenvolvidas visando implementar um conjunto de funções comuns que podem ser reaproveitadas pela maioria dos usuários da biblioteca. Entre essas funções têm-se as quatro operações SNMP (get, get-next, set e trap), aplicações que realizam comandos semelhantes ao do Unix, como o netstat e outros.

Essas aplicações facilitam bastante o trabalho para o programador, uma vez que funções básicas como o get e o get-next, que implementam a base para uma consulta SNMP, são sempre necessárias e já se encontram prontas para o uso, bastando apenas que elas sejam encaixadas no aplicativo desenvolvido.

A documentação disponibilizada pelo CMU é que fica realmente muito a dever. A documentação sobre as funções da biblioteca é fraquíssima. Ela explica um subconjunto de funções e o significado dos campos nas estruturas de dados utilizadas, porém essa explicação é extremamente superficial e não prepara ninguém para utilizar a biblioteca. O pacote consegue se salvar realmente por causa das aplicações que por já estarem prontas e serem de grande utilidade, facilitam a utilização do pacote sem que seja necessário um conhecimento aprofundado sobre como funcionam os programas.

3.3 WWW

O WWW (World Wide Web) é um instrumento utilizado na internet para consulta a documentos hipermídia. Através de um conjunto de recursos, que vão desde o texto até visualização de figuras, filmes e sons, o WWW segue uma linha não seqüencial de consultas à documentos, tornando mais fácil a busca de informações.

Ele foi desenvolvido para ser um centralizador de conhecimento, onde qualquer pessoa situada em qualquer parte do mundo pode publicar as suas idéias e seus conhecimentos a respeito de um determinado assunto que pode vir a interessar outras pessoas.

Criado em 1989 por Tim Berners-Lee, do CERN, ele se disseminou vertiginosamente na internet, tornando-se o ambiente mais popular da internet e

passando a ser consultado pela mais variada gama de usuários, desde pesquisadores e empresários até donas de casa.

Baseado numa arquitetura cliente-servidor, o WWW disponibiliza os seus documentos através de um computador denominado servidor HTTP. O HTTP (HyperText Transfer Protocol) é o protocolo de transferência de textos utilizado, cujo ponto forte é a velocidade da transmissão de dados.

Para os clientes acessarem o servidor, é necessário o uso de um software denominado browser. Através deste programa, o cliente conectado a internet busca um documento através do endereço requisitado pelo usuário. O browser realiza a comunicação com o servidor, buscando o documento e apresentando-o na tela.

Os documentos trazidos pelo browser são escritos em HTML (HyperText Markup Language), um padrão que permite o armazenamento e intercâmbio de informações em sistemas abertos. O HTML define a apresentação e estrutura dos documentos através de marcas definidas na linguagem e conhecidas como tags. Esses tags formatam o documento e definem como será feita a sua apresentação. Essa apresentação é feita em formato gráfico, com fontes de tamanhos variados, apresentação de imagens e de links que permitem que se faça referência a outros documentos.

O HTML é um dos principais responsáveis pela velocidade e beleza do WWW. O fato da montagem da página gráfica ser feita no browser do cliente ao invés de no servidor faz com que o tamanho dos documentos seja relativamente pequeno, gerando uma baixa carga de comunicação na rede.

Os documentos e recursos da internet são todos encontrados através da sua URL (Uniform Resource Locator) . A URL é um padrão de identificação única de documentos dentro da internet. Ela se encontra estruturada em três partes distintas : O protocolo que será utilizado (http, ftp, news, ...), o servidor onde o arquivo se encontra e a localização deste arquivo dentro da máquina. Desta forma, na URL

http://tucano.inf.ufrgs.br/homepage.html

é utilizado o protocolo de comunicação http, o servidor tucano.inf.ufrgs.br e o arquivo homepage.html, formando assim o endereço da página de apresentação do instituto de informática da UFRGS.

Um dos grandes problemas que surge com a explosão do WWW é da atualização dos dados. Considerando que os documentos são guardados em arquivos, a atualização dos dados contidos nestes arquivos requer que sejam criados novos arquivos. Além disso, para se fornecer informações a respeito de dados de uma empresa ou organização, os arquivos ofereceriam uma quantidade restrita de informações se comparados ao tamanho do banco de dados da entidade. Outro grave problema é que a utilização de arquivos gera uma interface somente de saída com o usuário, não permitindo que ele agregue novas informações.

Visando solucionar este tipo de problema, foram criados meios de conexão do WWW com bancos de dados. Dentro desta conexão, existe um agente chamado gateway, que faz a conexão entre o banco de dados e o WWW, criando um mecanismo conhecido como cgi (Common Gateway Interface).

Através de programas cgi pode-se criar páginas em HTML conforme for o contexto pedido pelo usuário. Dentro deste contexto, o usuário não interage mais com o WWW somente através da leitura aos seus dados e da seleção de seus link, mas também através de informações passadas com o uso de formulários. Os formulários são páginas em HTML que possuem a maioria dos controles fornecidos em sistemas operacionais gráficos, tais como campos de edição, listas, combos, botões de rádio e de check, etc...

Através destes formulários o usuário pode não só requisitar consultas ao WWW, como também fornecer informações para serem processadas pela empresa.

Desta maneira, é possível buscar informações contidas em bancos de dados, tanto genérica ou especializada, conforme for necessário. Estas consultas permitem que os dados disponibilizados pelo WWW estejam tão atualizados como o banco de dados utilizado por ela para manter os seus documentos.

3.4 Ping

O ping (packet internet groper) é um aplicativo que permite verificar o estado da rede e de algum nodo específico presente na mesma. Para isso, o nodo requisitaste envia um pacote de pedido de eco ICMP (eco request) para o endereço IP desejado e aguarda por uma mensagem de retorno (echo reply).

Tendo em vista que a implementação de ICMP é obrigatória para todos os dispositivos que suportam o protocolo da internet, o envio ou não de uma mensagem de retorno permite verificar se o nodo destino se encontra alcançável e ativo. A falta de uma mensagem de retorno pode significar um dos seguintes problemas :

- Um dos gateways intermediários entre os nodos não se encontra operando ou não está roteando os pacotes corretamente,
- O nodo destino não está rodando,
- O nodo destino se encontra rodando mas não está com o software de ICMP e IP funcionando, ou
- Existe algum problema nas informações de roteamento dos nodos intermediários.

4. Definição do Trabalho

Conforme foi mencionado na introdução do trabalho, uma das grandes necessidades em ambientes de redes que possuem um tamanho considerável é a sua gerência de configuração. O conceito de gerência de configuração aqui citado não está restrito somente a capacidade de guardar as informações a respeito dos nodos que estão conectados à rede, mas também de descobrir que nodos se encontram ligados a ela.

Um dos grandes problemas enfrentados em organizações cujas redes já se expandiram razoavelmente é o controle de inclusão e exclusão de nodos na mesma. Imaginem o caso da UFRGS do exemplo. A rede de computadores da UFRGS se encontra espalhada por diversos institutos da universidade, entre eles o de Informática, Física, Biologia, Matemática, CPD, CESUP, etc... Cada um destes institutos possui no mínimo uma sub-rede para interligar as suas máquinas. Algumas destas unidades, como o Instituto de Informática, possui várias sub-redes espalhadas nas suas dependências. Todas essas sub-redes são consideradas elementos que compõem a rede da UFRGS como um todo. A inclusão ou exclusão de nodos nessas sub-redes é também a inclusão ou exclusão de nodos na rede da UFRGS.

Considerando a quantidade de institutos localizados em dependências distintas e muitas vezes distantes entre si, bem como o número razoável de sub-redes que cada uma destas unidades pode ter, o controle de todos os hosts conectados à rede da UFRGS torna-se uma tarefa difícil de controlar. Para que esse controle seja efetivo, é necessário que exista uma forte disciplina a fim de informar sempre à gerência da rede sobre quaisquer modificações efetuadas, de forma que ela possa estar sempre atualizada. Infelizmente, a adoção e controle desta disciplina por parte de todos os institutos que fazem parte da UFRGS é uma tarefa difícil e árdua, tornando-se necessário que esse controle seja feito de maneira automática.

Dentro deste contexto surgiu a idéia do WatchDog. O WatchDog é um sistema que irá manter atualizada a base de dados do nosso gerente de configuração. Mais do que apenas verificar se os hosts conhecidos continuam com a mesma configuração, o WathDog irá realizar um estudo sobre as mudanças que ocorreram na rede como um todo. Esse estudo compreende em verificar se novas sub-redes foram criadas, se novos nodos foram conectados às redes já conhecidas e se algum nodo se encontra a um tempo razoavelmente grande sem ser encontrado, informando ao usuário da possibilidade deste nodo ter sido desconectado da rede.

Além disso, o WatchDog irá registrar os nodos que estavam ativos no momento em que ele foi disparado, permitindo um controle histórico sobre a utilização dos nodos e mostrando os nodos que se encontravam ativos num determinado instante . Esse controle, chamado de controle de contabilização, irá permitir o acompanhamento da utilização da rede (nodos que ficam maior tempo ligado ou desligado) e que se possa suspeitar sobre a exclusão de algum nodo (caso este nodo não se encontre conectado a um tempo considerável).

O sistema de gerência de configuração de rede WatchDog se encontra estruturado pelos seguintes elementos :

a) Um mecanismo de banco de dados que armazena as informações gerenciadas pelo sistema,

b) Um processo que faz a detecção e coleta de informações da rede,

c) Uma interface que permite que sejam feitas consultas sobre as informações gerenciadas.

O restante deste trabalho irá realizar um estudo aprofundado sobre os problemas e necessidades para se implementar um sistema desse tipo. Serão analisados quais

objetos de cada dispositivo serão armazenados pelo sistema, seguido de uma definição de diagrama de entidade-relacionamento para armazenar esses objetos.

No capítulo seguinte serão analisadas as funcionalidades que uma interface para esse sistema deve possuir e as várias opções de algoritmos para se realizar a busca de informações sobre a rede, seguido da escolha de um deles.

4.1 Escolha dos Objetos Gerenciados

A escolha dos objetos que serão armazenados possuem um reflexo direto com a funcionalidade que o sistema irá trazer para o usuário. O WatchDog irá armazenar objetos que tragam informações a respeito da configuração do dispositivo. Esses objetos serão todos retirados da MIB de cada nodo utilizando-se consultas SNMP. A escolha do SNMP como protocolo de gerenciamento se deve a uma série de razões, estando entre elas as seguintes considerações :

- A rede da UFRGS é predominantemente TCP/IP, possuindo várias máquinas com agentes SNMP instalados.
- O SNMP já foi aceito a nível mundial como protocolo de gerenciamento de redes de fato dentro do ambiente da internet,
- A simplicidade das operações oferecidas pelo SNMP.

Existem várias implementações do SNMP de domínio público, entre elas a da CMU, na qual será utilizada para desenvolver o sistema.

Uma vez escolhida a forma como a MIB será acessada, deve-se selecionar quais os objetos da MIB que são de interesse para o sistema. Dentro deste escopo, interessam ser armazenados :

a) Informações sobre a configuração atual do nodo, onde ele se encontra, etc.
b) Informações sobre as interfaces que ligam os nodos a rede,
c) Informações sobre o estado atual do nodo, se ele está operacional ou não, etc..

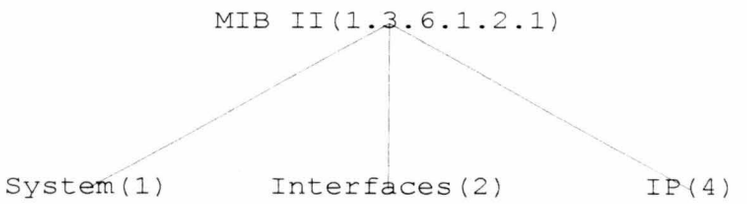

Figura 4.1 : Grupos da MIB-II utilizados na Gerência de Configuração

Dentro dos onze grupos presentes na atual MIB-II, são de interesse para o trabalho apenas os primeiros grupos : system(1), interface(2) e ip(4). O terceiro grupo, at(Address Translantion), era utilizado na MIB-I e suas informações foram redistribuídas em outros grupos na nova versão da MIB, estando presente na MIB-II apenas para manter a compatibilidade. Os grupos restantes, apesar de armazenarem algumas informações de configuração, não trazem nenhuma que seja de grande relevância ao escopo deste trabalho. A figura 4.1 mostra um mapa com os grupos que serão utilizados. A seguir será analisado cada grupo e verificado que informações devem ser retiradas de lá.

4.1.1 System

Neste grupo encontram-se informações sobre o nodo como um todo, sendo um dos principais grupos dentro da MIB para o tipo de dados que se deseja gerenciar. Os seguintes objetos foram escolhidos dentro deste grupo(Figura 4.2) :

- sysName - Possui um nome administrativo para o nodo gerenciado.
- sysDescr - Possui uma descrição textual do nodo. Normalmente este campo possui o nome do tipo de hardware e a sua identificação de versão, bem como o nome do sistema operacional e do software de rede.
- sysObjectID - Possui o Id com a identificação da autorização do fabricante, ajudando assim a localizar o fabricante do nodo.
- sysUpTime - Indica em centésimos de segundos a quanto tempo o nodo está em funcionamento. Apesar de não ser uma informação típica de configuração, é importante para o controle da contabilização do nodo, indicando assim não somente quais nodos estão ativos no momento em que o WatchDog foi disparado, mas também a quanto tempo.
- sysContact - Informa que pessoa deve ser contactada para a resolução de problemas, bem como a forma de encontra-la.
- sysLocation - Indica qual a localização física do nodo.
- sysServices - Indica qual o conjunto de serviços OSI que o nodo oferece. No caso do uso de protocolos padrão internet, deve-se considerar somente as camadas físicas(1), de enlace(2), de rede(3), de transporte(4) e de aplicação(7).

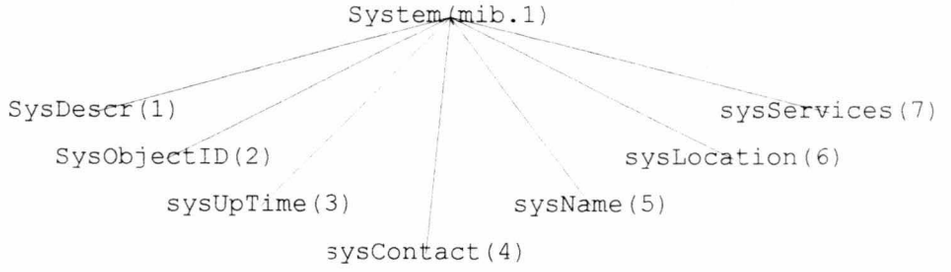

Figura 4.2 : Objetos Gerenciados do grupo system

4.1.2 Interface

Este grupo contém informações sobre todas as interfaces do dispositivo. Ele é composto por um objeto, ifNumber, que indica quantas interfaces o nodo possui, e por outro objeto, ifTable, que possui informações sobre todas as interfaces do dispositivo. O objeto ifTable na verdade é um array com um número de entradas equivalente a ifNumber, onde cada entrada de ifTable possui informações sobre uma interface específica. Essas entradas são do tipo ifEntry, e possuem um conjunto de objetos que descrevem as características da interface. A figura 4.3 mostra o grupo Interface detalhadamente, onde os seguintes objetos de ifEntry interessam ao trabalho :

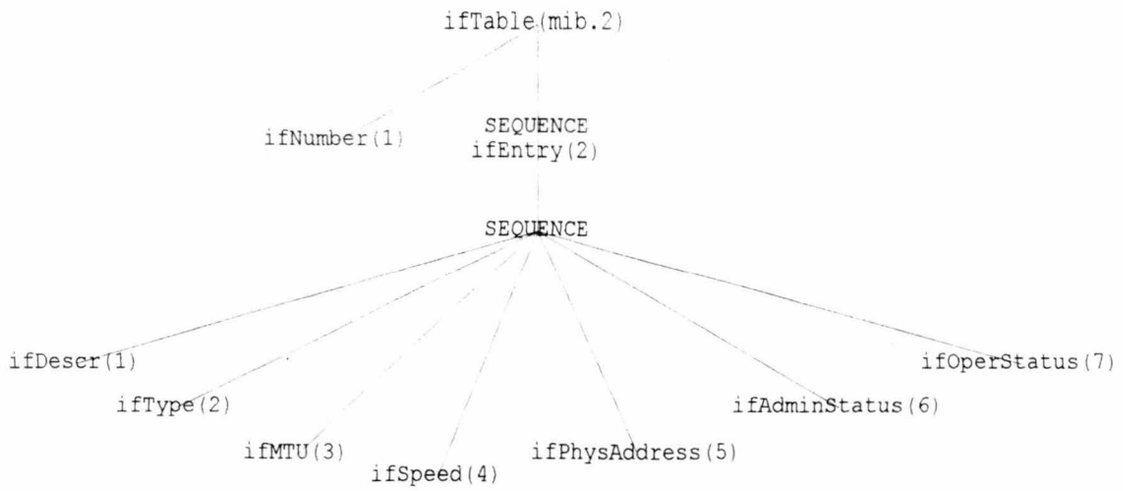

Figura 4.3 : Objetos gerenciados do grupo Interface

• ifIndex - Valor que varia de 1 até o número total de interfaces, é um valor único que serve como referência para relacionar uma determinada interface com objetos de outros grupos.

• ifDescr - Contém informações sobre a interface, podendo incluir o nome do fabricante, o nome do produto e o número da versão do hardware.

• ifType - Contém o tipo da interface (fddi, sdlc, frame-relay, ...), distinguida de acordo com o protocolo físico e de enlace. Ao contrário dos outros objetos, o valor de ifType é definido por uma enumeração e não por uma string.

• ifMTU - Contém o tamanho em octetos do maior datagrama que pode ser recebido ou enviado pela interface.

• ifSpeed - Contém a velocidade estimada da interface em bits por segundo(bps).

• ifPhysAddress - Contém o Endereço físico da Interface abaixo da camada de rede.

• ifAdminStatus - Contém o estado desejado para a interface. Pode assumir o estado Up, Down e Testing.

• ifOperStatus - Contém o estado operacional em que a interface realmente se encontra. Através do cruzamento do estado desejado para a interface(ifAdminStatus) com o valor em que ela está (ifOperStatus), se obtém o modo na qual ela está operando (tabela 4.1) : Operacional, Falha, Desligada, Teste ou Impróprio.

Tabela 4.1 : Modo de Operação da Interface

ifOperStatus\ifAdminStatus	Up(1)	Down(2)	Testing(3)
Up(1)	Operacional	Impróprio	Impróprio
Down(2)	Falha	Desligada	Impróprio
Testing(3)	Impróprio	Impróprio	Teste

4.1.3 IP

Este grupo contém as informações que são utilizadas pelo protocolo IP. Ele possui tanto informações de configuração do host como também as informações de endereçamento IP das interfaces obtidas no grupo interfaces. Os objetos utilizados são(figura 4.4) :

- ipForwarding - Indica se o nodo age como um gateway IP em relação aos pacotes recebidos e não endereçados a si ou como um host.
- ipAddrTable - Tabela com informações sobre os endereços IP do nodo. É um array onde cada linha, do tipo ipAddrEntry, possui informações sobre um desses endereços. Para cada entrada desta tabela, obtêm-se os seguintes objetos :
  - ipAdEntIfIndex - Identifica a qual interface os dados IP se aplicam, referenciando o número de uma das entradas da tabela de interfaces (ifTable). Toda entrada em ipAddrTable possui uma interface correspondente, e uma interface possui no máximo uma entrada em ipAddrTable.
  - ipAdEntAddr - O endereço IP da entrada da tabela.
  - ipAdEntNetMask - Máscara de rede que ao ser utilizada junto o endereço IP, informa a qual rede aquela interface está conectada.

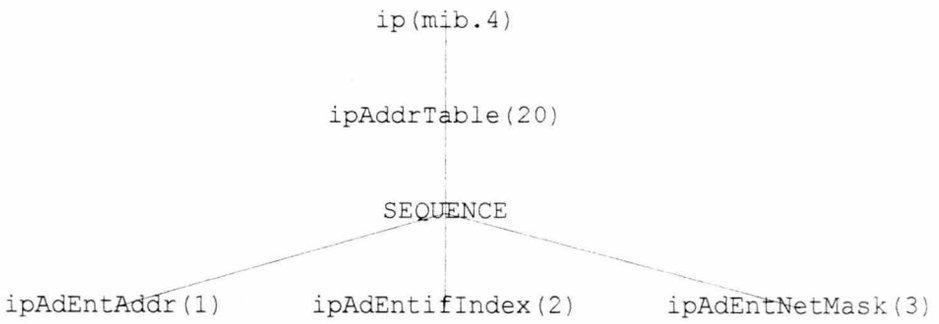

Figura 4.4 : Objetos gerenciados do grupo IP

4.2 Definição do Banco de Dados Lógico

Um dos requisitos básicos para se desenvolver um software de gerência de configuração é que os dados coletados pelo gerente a respeito dos nodos presentes na rede e sobre a estrutura da rede como um todo não se percam , de modo que a cada vez que se deseje consultar alguma informação sobre a rede, não seja necessário ativar o algoritmo de detecção para descobrir essa informação.

Existe uma série de vantagens em armazenar os dados obtidos pela última ativação do algoritmo de detecção ao invés de reativar o algoritmo a cada nova consulta submetida ao sistema. Entre elas podem ser citadas as seguintes :

a) As redes de computadores são estruturas com razoável nível de estaticidade. As mudanças de configuração de um nodo ocorrem pouquíssimas vezes em relação ao seu tempo de permanência na rede. Ou seja, as informações sobre um determinado nodo são atualizadas eventualmente, criando assim um ambiente com pequeno índice de modificação. Em contrapartida, pedidos de informações sobre um determinado nodo podem ocorrer com freqüência, seja para atender a pedidos específicos de consulta daquele nodo, ou para verificar se o nodo se encaixa numa determinada condição de busca que foi submetida ao gerente visando filtrar um conjunto restrito de dispositivos.

b) Enquanto a consulta a um arquivo buscando informações de um determinado nodo pode ser feita numa fração pequena de tempo, consultas a agentes SNMP podem

levar um tempo com algumas ordens de grandeza maiores em relação aos arquivos. Esse tempo de busca vai depender :

• Do tipo de consulta necessária : Consultas a somente um nodo da rede levam um tempo bem menor de resposta se comparadas a consultas a todos os nodos de uma determinada sub-rede, ou a consulta sobre toda a estrutura da rede, sua divisão em sub-redes, etc...

• Da forma como se obtém acesso à sub-rede ou nodo que se deseja consultar : as respostas de nodos que estão conectados a mesma rede do nodo que está buscando informações terão um tempo bem menor se comparadas ao tempo de resposta de consultas a nodos que estão conectados a outras redes, onde existe a necessidade da resposta ser roteada por vários gateways e onde existe o risco de se enfrentar congestionamento entre os nodos e em que parte das rotas podem possuir uma velocidade menor de transmissão.

O grande problema do armazenamento dos dados é que o resultado das consultas não irão refletir as modificações feitas recentemente. Uma vez atualizado os dados em arquivo através do algoritmo de detecção, quaisquer mudanças que venham a ocorrer na rede após o término do algoritmo só se refletirão para o gerente quando o algoritmo for executado novamente. Ou seja, as informações armazenadas refletirão a realidade da rede somente no momento em que ela foi percorrida. Quaisquer inclusões ou exclusões de nodos, bem como modificações em sua configuração, não serão vistas até que a rede seja percorrida novamente.

Isso exige não só que o algoritmo de detecção seja disparado periodicamentecomo também que se estude a freqüência em que esse disparo deve ocorrer. Essa frequência vai depender da necessidade com que são exigidas informações atualizadas para a gerência da rede, e da disponibilidade de recursos computacionais para suportar o algoritmo de detecção, uma vez que em redes de porte médio a grande ele irá demorar um tempo razoável e consumir uma parcela significativa da CPU da máquina.

Porém, o problema da informação desatualizada assume proporções bem menores se comparada a demora que pode levar a consulta on-line dos dispositivos da rede em tempo de execução da consulta. O tempo de desatualização dos dados pode ser facilmente medido e adequado às necessidades da gerência da rede através de uma escolha adequada da freqüência em que o algoritmo deve ser disparado. Já a demora para se obter as informações on-line não possui uma solução de fácil implementação, uma vez que as suas barreiras são físicas, dependem do tráfego, velocidade e tamanho da rede.

Perante essa necessidade implícita de se armazenar as informações, deve-se escolher em que formato serão armazenados os dados : como arquivos ascii normais ou através de um sistema gerenciador de banco de dados.

Como já explanado no capítulo 1, arquivos ascii são mais fáceis de serem implementados, além de possuírem uma execução mais rápida. Porém, realizar consultas sobre a rede como um todo utilizando arquivos ascii se torna uma atividade complexa, uma vez que eles não disponibilizam o uso de índices para pesquisa nem tão pouco a utilização de comandos SQL para expressar o consulta que se quer realizar. Por isso o uso de bancos de dados tornou-se a alternativa mais adequada ao nosso ambiente.

Uma vez escolhido SGBDs para o armazenamento dos dados, torna-se óbvia a necessidade de estudar a forma como serão guardadas as informações dentro do mesmo. Como será utilizado um banco de dados relacional, deve-se definir a estrutura de entidade-relacionamento que irá armazenar os dados.

A definição das entidades vai de encontro a um estudo sobre como as informações estão estruturadas no mundo real. No caso do gerente, tem-se claramente

três níveis de elementos que se agregam através de relações de composição : Redes, Dispositivos e Interfaces.

As redes são compostas por um conjunto de máquinas que estão conectadas a ela. Cada máquina, por sua vez, possui um conjunto de interfaces que serve para conecta-las às diversas redes na qual ela faz parte. Pode-se assim chegar a diversas conclusões que ajudarão a definir o modelo E-R :

a) Existem inicialmente três entidades bem destacadas : Redes, Interfaces e Dispositivos.

b) Uma interface não existe sem um dispositivo, existindo assim um relacionamento de 1:n entre eles, uma vez que uma interface é de somente um dispositivo, enquanto que um dispositivo pode possuir várias interfaces. A interface, portanto, é considerada uma entidade fraca.

c) Apesar de uma rede ser composta por dispositivos, o que liga um dispositivo a uma rede é a sua interface. Ou seja, o relacionamento entre dispositivos e redes se dá indiretamente através de um relacionamento 1:n entre interfaces e redes (uma vez que uma interface pode estar ligada a no máximo uma rede, mas uma rede provavelmente possui mais de um dispositivo conectado a ela) .

d) Duas redes se interligam através de dispositivos que possuem interfaces para redes distintas. Esses dispositivos provavelmente serão gateways e servirão como meio de acesso para a comunicação entre os nodos de redes distintas.

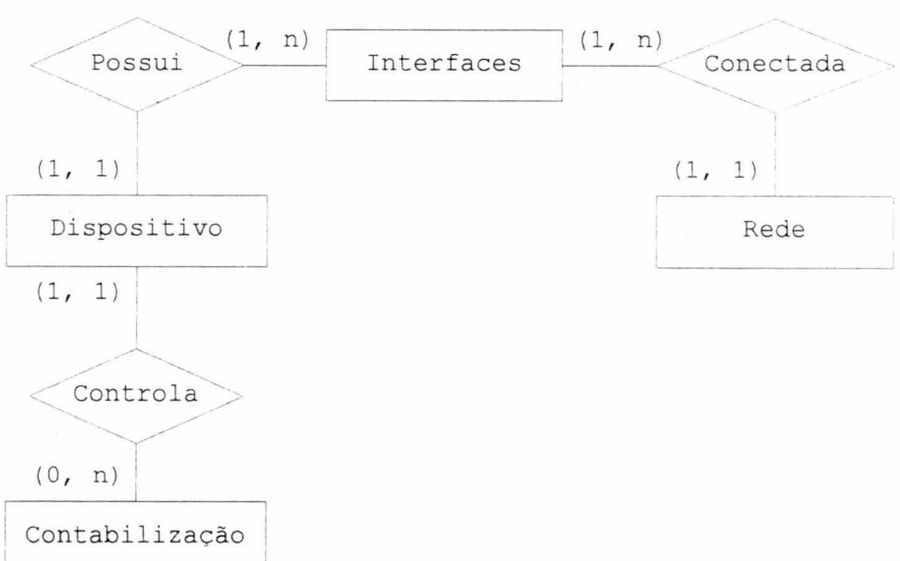

Figura 4.5 : Diagrama E-R do modelo proposto

Além disso, devem ser armazenadas informações sobre o histórico de utilização de cada nodo, guardando quais nodos estavam ativos quando o algoritmo de detecção foi disparado. Esse armazenamento, chamado de controle de contabilização, é representado através de uma entidade de contabilização que se relaciona com o dispositivo.

A figura 4.5 mostra o diagrama E-R proposto. Cabe ressaltar que a entidade de dispositivo está sendo utilizada para representar tanto nodos gateways como nodos hosts. Apesar da divisão desta entidade em duas trazer maior clareza para o modelo(Figura 4.6), ela traria sérios prejuízos na performance das consultas, uma vez que a busca de dispositivos que fazem parte de uma rede, ou que atendem a uma determinada condição de busca, no caso da utilização de duas entidades, traria a necessidade de se realizar a mesma consulta em duas entidades, realizando um merge

depois entre os resultados. Além disso, apesar das entidades possuírem funções completamente distintas dentro da rede, elas possuem estruturas de dados semelhantes, o que facilita bastante a sua representação numa só entidade.

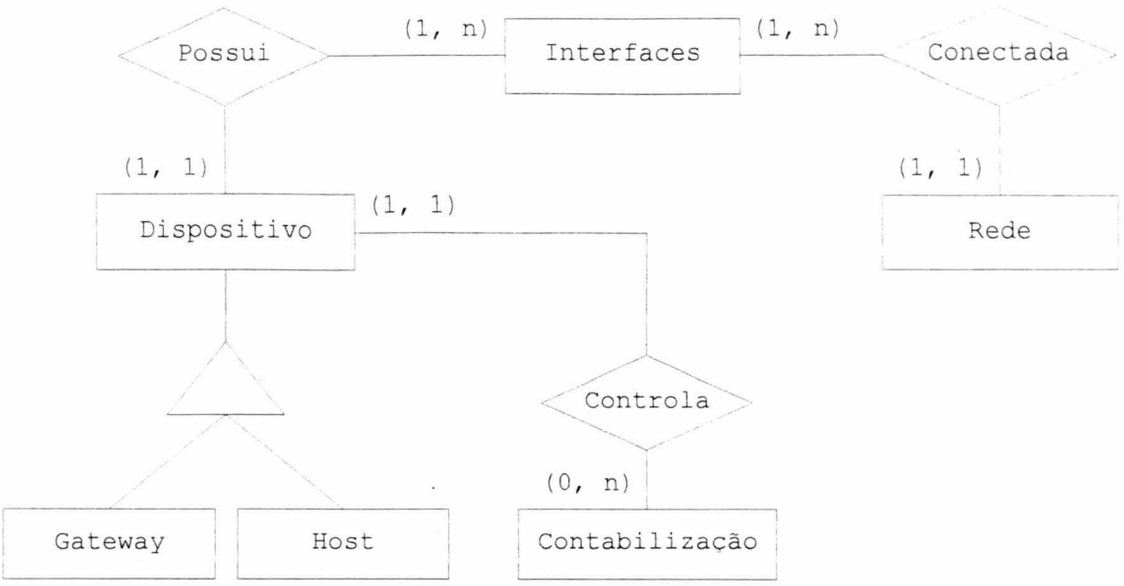

Figura 4.6 : Diagrama E-R com a divisão da entidade de Dispositivo

Resta agora definir os atributos que compõe cada entidade. Eles serão na sua maioria retirados da MIB e armazenados como atributos em uma das entidades. Como a própria MIB já é organizada de forma estruturada, é fácil mapear os diversos objetos de cada grupo diretamente para atributos de uma determinada entidade.

Por exemplo, o grupo system traz informações sobre o nodo como um todo, sendo mapeado para a entidade de dispositivo. Já o grupo de interface, como diz o nome, traz informações sobre as interfaces de um dispositivo, sendo mapeado para a entidade interface. Cada entrada ifEntry deste grupo irá gerar um registro no banco de dados com informações sobre uma das interfaces do dispositivo.

Finalmente, o grupo IP é o único que não possui uma correspondência direta. Os objetos ipAdEntIfIndex e ipAdEntAddr serão utilizados para armazenar na entidade interface a que interfaces correspondem os endereços IP de cada dispositivo. Apesar de todo endereço IP estar sempre ligada a uma interface, e uma interface ter no máximo um endereço IP, é possível existirem interfaces sem endereços IP correspondentes.

Através dos dois atributos acima utilizados em conjunto com ipAdNetMask, pode-se obter o endereço de rede do dispositivo. É importante ressaltar aqui que o cruzamento do endereço IP de um dispositivo com a sua máscara de rede é o único meio de se descobrir a existência de uma rede. As redes são descobertas pelo gerente através da identificação de dispositivos que estão conectados a ela. Ela não possui informação própria, e todos os dados cadastrados sobre ela são obtidos através de cálculos efetuados sobre os dados obtidos dos dispositivos. Não existe uma MIB que traga informações sobre uma rede da forma que existe com os nodos. Portanto essas informações devem ser todas deduzidas. Os atributos derivados desta entidade são os seguintes :

- NumHosts - Contabiliza quantos hosts a rede possui conectados. Apesar de se conseguir essa informação através de um procedimento de contagem no banco de dados, esse atributo é mantido para aumentar a performance.

• NumGateways - Contabiliza quantos gateways a rede possui conectados.

• Classe - Indica a classe a qual pertence a rede(A, B ou C). A classe da rede é obtida através da análise do seu número IP.

Existem também nas outras entidades informações adicionais que são obtidas pelo algoritmo de detecção ou pela análise dos dados coletados. Na entidade de Dispositivos existem os seguintes atributos adicionais :

• Fala_SNMP : Atributo Booleano que indica se o dispositivo possui um agente SNMP. Esse atributo é setado como Falso caso o dispositivo não responda a nenhuma consulta do algoritmo. Nesse caso, o dispositivo é registrado no banco de dados mas são armazenadas somente as suas informações básicas as quais não necessitam que se consulte, tais como a existência de pelo menos uma interface com o endereço IP utilizado para tentar acessá-lo, e a ligação deste endereço à rede que disparou o seu pedido de informações.

• Tipo : Enumeração que diferencia um Host de um Gateway. Essa diferenciação é obtida através do objeto ipForwarding.

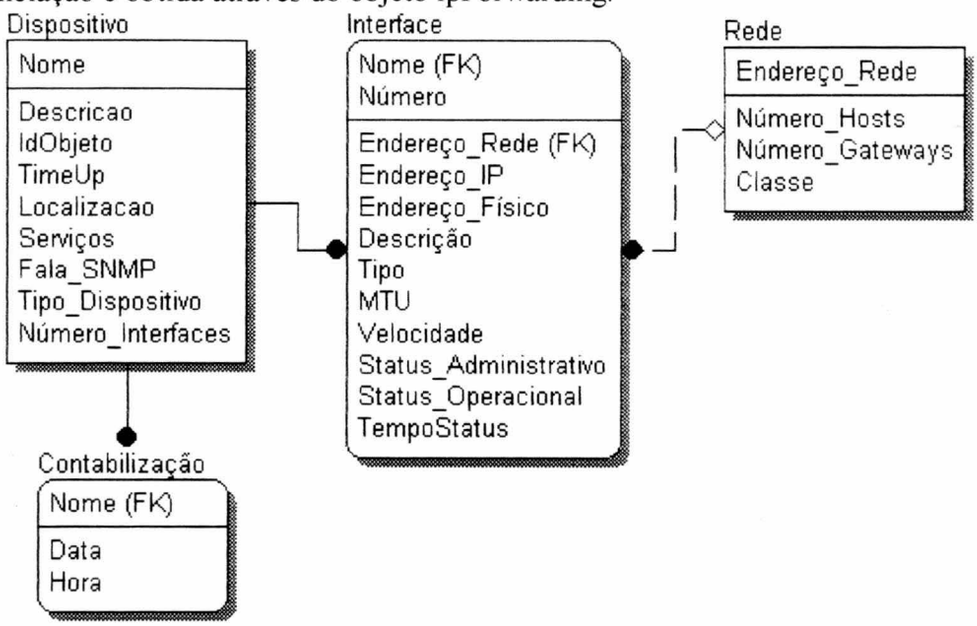

Figura 4.7 : Modelagem das entidades propostas

Finalmente, a entidade de contabilização têm os seus dados obtidos através do registro da data e hora em que cada host foi detectado durante a execução do algoritmo. Seguindo esta descrição e distribuindo os objetos coletados da MIB, as entidades são modeladas conforme mostra a figura 4.7.

4.3 Modelagem Física dos Dados

Uma vez definido o modelo lógico dos dados, deve-se definir como esse modelo será armazenado fisicamente. Para tanto, parte-se de algumas definições que irão transportar o modelo lógico para o modelo físico :

a) As entidades que representam objetos existentes no mundo real (Rede, Dispositivo e Interface) possuirão um código que será único e servirá como chave primária para a mesma. Esse código será utilizado internamente e será transparente para o usuário. A utilização de códigos substituindo chaves primárias digitadas pelo usuário possui uma série de vantagens, entre elas temos :

• Uma razoável economia de espaço, uma vez que o tamanho de um código normalmente é menor do que o tamanho de um campo que contém um nome

único da entidade (Por exemplo, é melhor dar um código de 5 caracteres a uma rede para representa-la internamente em um dos seus relacionamentos. Esse código possibilitará a existência de até cem mil redes distintas, número esse suficiente para a aplicação desejada, e bem menor do que o seu número IP de quinze caracteres).

- Evita que mudanças de valor na chave de primária acarretem na propagação dos seus efeitos por todas as entidades que se relacionam com ela. Por exemplo, se um nodo de nome Vega tivesse seu nome alterado para Vegas, e não fosse utilizado código, essa alteração teria que se refletir em todos os registros da entidade contabilização e Interface. Já se fosse utilizado o código 100 como chave primária, nada teria que ser alterado em outras entidades, uma vez que apesar do seu nome ter mudado, o seu código permanece o mesmo.

b) Os objetos da MIB cujos valores não necessitam de conversão para serem compreendidos serão guardados no mesmo formato em que eles se encontram.

c) Os objetos da MIB que necessitam de conversão, serão convertidos antes de serem armazenados no Banco de Dados, de forma a otimizar o tempo de resposta na hora da consulta.

A ordem de magnitude de cada código assume a seqüência mostrada na figura 4.8, fazendo portanto que uma rede possua um código de tamanho menor que um dispositivo, que é menor por sua vez do código de uma interface.

Figura 4.8 : Ordem de magnitude do número de registros em cada entidade

Outra modificação necessária a fim de melhorar a performance do sistema é a inclusão de um atributo de Tempo na entidade de contabilização, que armazenará quantos segundos se passaram desde o dia 01/01/70 até a data atual. A data de referência é utilizada devido a funcionalidade fornecida pela linguagem C. Através deste atributo fica bem mais fácil realizar a ordenação dos registros de contabilização, uma vez que no Postgres não existe nenhum campo do tipo data.

O Banco de dados fisico fica então definido conforme mostrado na figura 4.9.

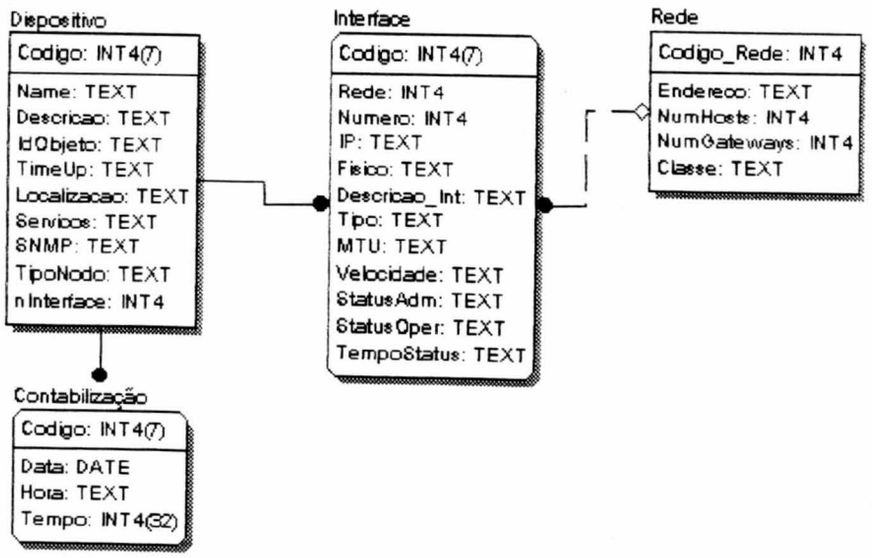

Figura 4.9 : Banco de Dados Físico

5. Implementação do sistema

5.1 Interface em WWW

A interface do sistema será a responsável por responder por todas as consultas que seus usuários desejarem submeter ao gerente de configuração. Essa interface será desenvolvida em HTML(HyperText Marked Language) de modo que a sua utilização seja feita via internet no WWW.

O acesso fácil e eficiente que a internet oferece ao usuário é um dos principais motivos para se escolher o WWW como plataforma de interface. Essa característica faz com que o sistema possa ser utilizado de um computador de qualquer parte do mundo, desde que o mesmo esteja conectado a internet. No caso específico da UFRGS, a maioria dos seus computadores que estão interligados em rede possuem acesso a internet. Desta maneira, usuários de qualquer departamento ou instituto poderão usufruir do sistema.

Além disso, a independência do tipo de hardware que a internet proporciona faz com que o gerente possa ser utilizado em qualquer plataforma disponível. Isso facilita ainda mais a sua utilização de qualquer parte da UFRGS, não importando se quem o utilizará possui um PC ou de uma estação de trabalho.

A função básica da interface é permitir que os dados que o gerente dispõe sejam consultados. Essa consulta poderá ser tanto sobre a rede como sobre a utilização da mesma(contabilização). Além disso, o sistema disponibiliza também uma interface gerencial para o administrador, onde ele pode configurar o seu ambiente, excluir informações do WatchDog e vê as alterações encontradas na última passada do WatchDog sobre a rede.

A consulta sobre dados da rede poderá ser feita de duas maneiras :

a) De forma simples e intuitiva, obedecendo a estrutura de composição e hierarquia do ambiente, conforme mostra a figura 5.1. As informações obtidas aqui não possuem nenhuma condição de filtro e mostram todos os dados disponíveis em cada nível de consulta.

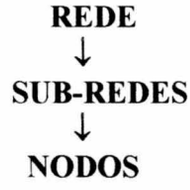

Figura 5.1 : Estrutura de consulta simples ao gerente.

b) De forma elaborada, buscando informações específicas de alguns elementos da rede que atendam a determinados requisitos de filtros. Por elementos aqui entendem-se os dois níveis que estão abaixo de rede e que formam a sua estrutura (sub-rede e nodos).

O primeiro tipo de consulta é fácil de ser imaginado. Ele obedece a hierarquia mostrada na figura 5.1 e é representado pelo mapeamento de uma página em HTML para cada um dos níveis da hierarquia.

Desta maneira, o primeiro nível representa uma visualização da rede como um todo, mostrando todas as sub-redes que a compõem. Apesar desta ser apenas uma visão geral, todas as informações sobre a sub-rede são mostradas, devido ao pequeno número

de atributos que a mesma possui. Assim, cada linha da listbox possuirá o endereço IP da sub-rede, o número de hosts e gateways que a mesma possui e a sua classe. Além disso, as linhas seguintes de cada sub-rede mostrarão quais gateways estão conectados a ela, seu nome, seu número IP e o seu estado atual (Figura 5.2).

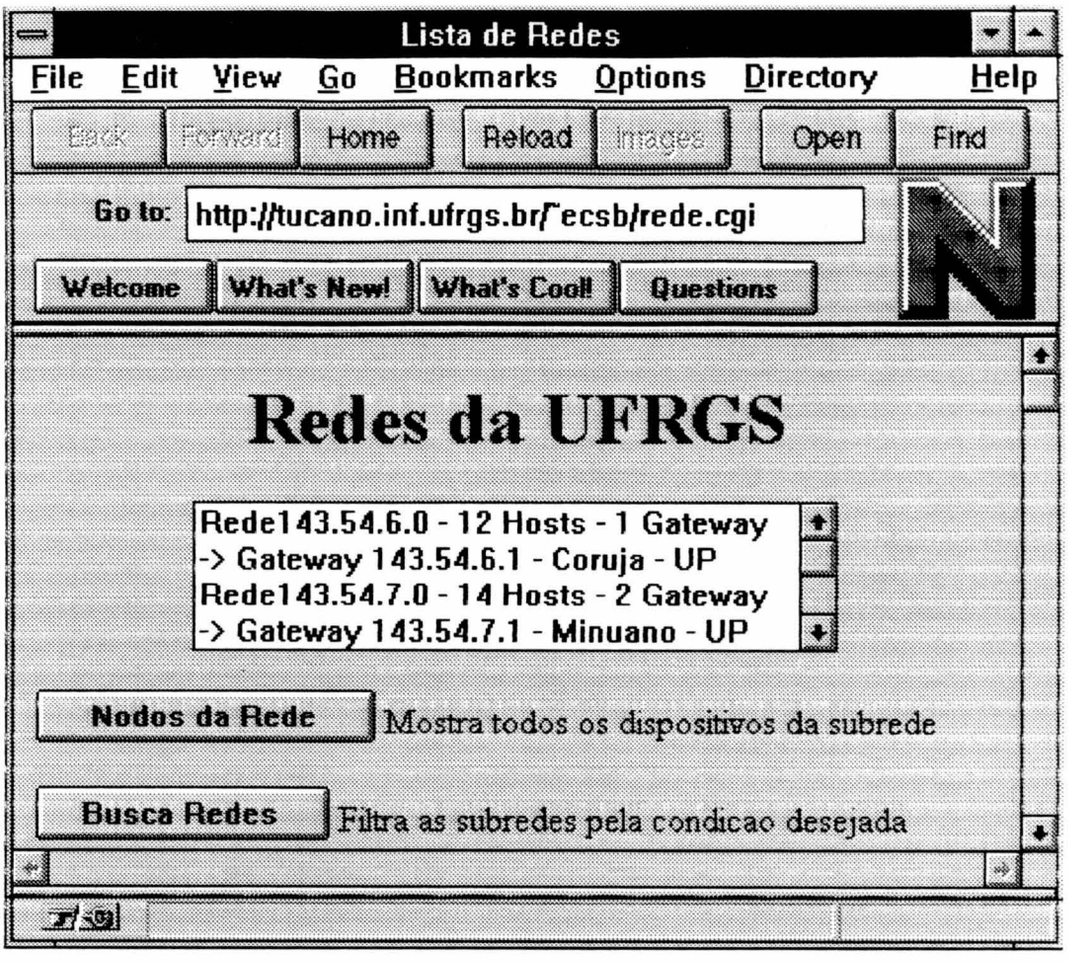

Figura 5.2 : Página WWW para consulta de Redes

O segundo nível representa a visualização de todos os dispositivos que estão conectados a uma determinada sub-rede selecionada no primeiro nível. Para cada dispositivo serão mostradas suas principais informações, tais como seu nome, seu número IP, sua função(Host/Gateway), seu número de interfaces e o seu estado operacional no momento(Figura 5.3).

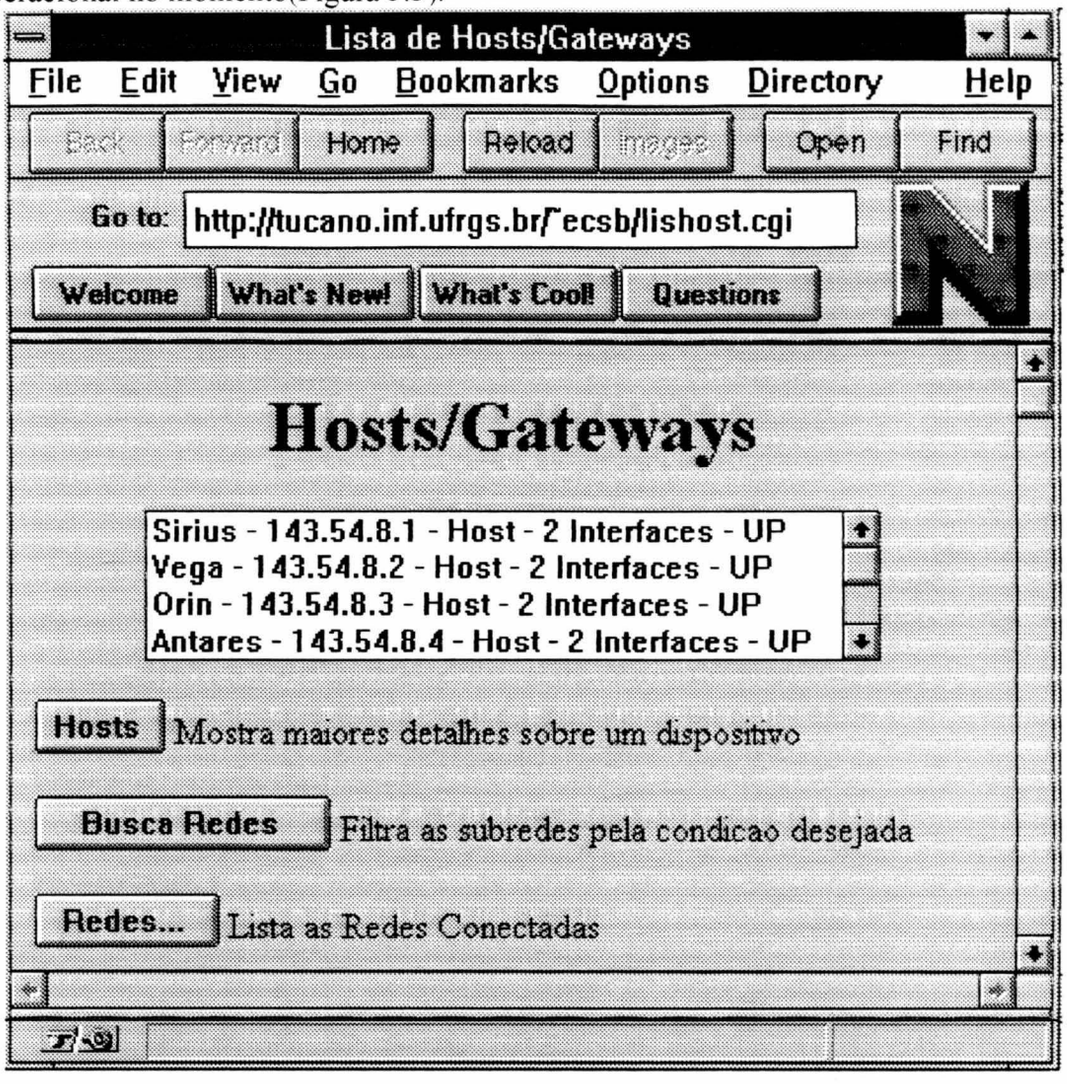

Figura 5.3 : Página WWW com a lista de Hosts de uma sub-rede

Já o terceiro nível mostrará informações completas a respeito de um determinado dispositivo da sub-rede atual. Neste momento, todas as informações que se dispõem a respeito de um determinado nodo, bem como de todas as suas interfaces, são visualizadas no WWW(Figura 5.4).

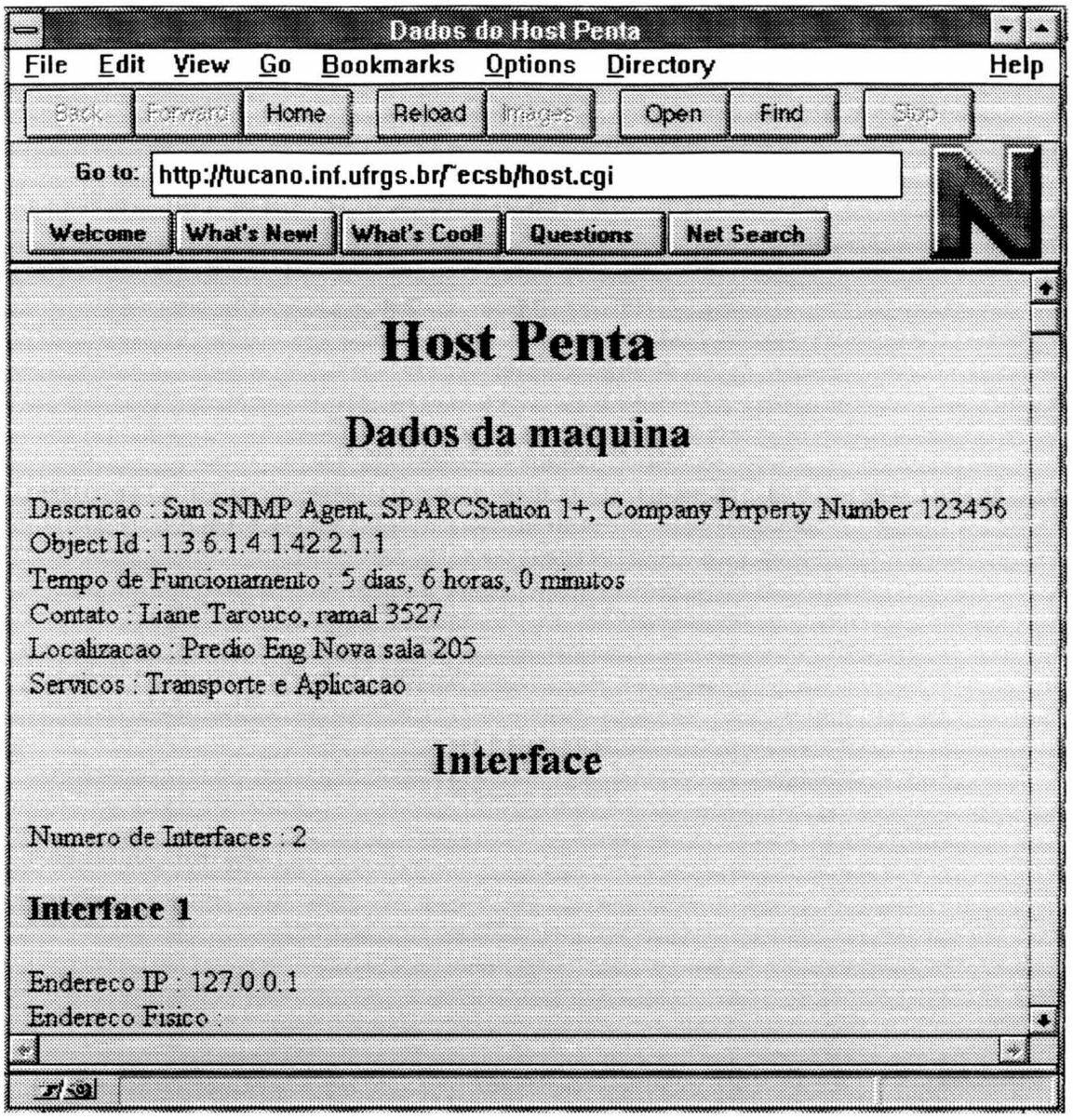

Figura 5.4 : Informações completas sobre a configuração de um dispositivo

A consulta elaborada visa buscar a informação gerenciada pelo sistema pelos mais diversos caminhos possíveis. A intenção do sistema é facilitar essa busca, permitindo ao usuário criar filtros de consulta à base da dados por qualquer um dos atributos existentes nas entidades. A principal idéia é a de o usuário ter um formulário com todos os atributos de uma sub-rede ou dispositivo, e dependendo da forma como ele preenche ou não os campos deste formulário, o sistema trará os registros cujos atributos se enquadraram dentro da condição solicitada no formulário.

Desta forma, pode-se selecionar quais sub-redes se deseja visualizar através dos atributos de classe, número de hosts e número de gateways. Como o preenchimento do número de hosts e de gateways especializaria por demais a consulta, foi optado por

requisitar o número mínimo e máximo de hosts/gateways que a rede deve possuir para ser mostrada no resultado. Esta tela de consulta, mostrada na figura 5.5, retorna como resultado uma página de primeiro nível, porém mostrando somente as sub-redes que atenderam aos requisitos solicitados.

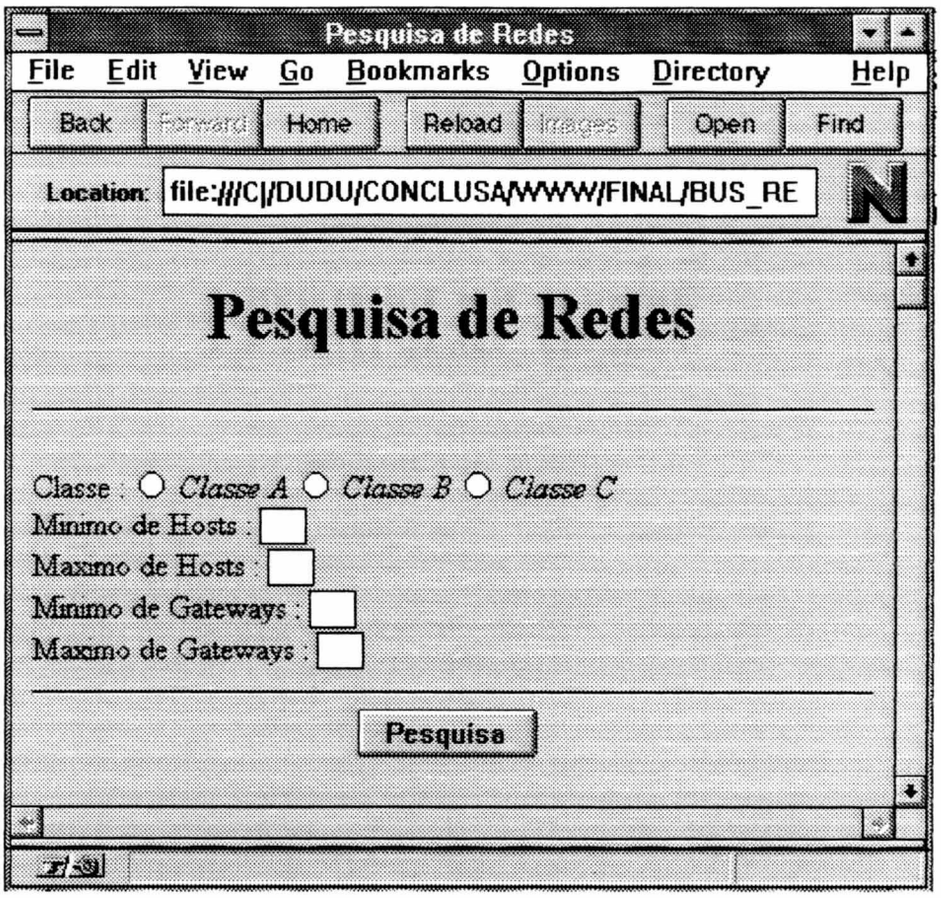

Figura 5.5 : Condições de Busca de Sub-redes

Já a busca de hosts, por sua vez, permite inicialmente que o usuário escolha com qual escopo de pesquisa ele deseja trabalhar : sobre todas a rede ou somente sobre uma determinada sub-rede. Definido esse escopo, pode-se consultar os nodos pelo seu nome, descrição ou função, bem como buscar dispositivos através das características de alguma das suas interfaces, como o seu nome, tipo, velocidade, status administrativo e status operacional. O resultado desta pesquisa traz uma tela compatível como o nível dois, com a informação adicional da rede a qual o dispositivo pertence (Figura 5.6).

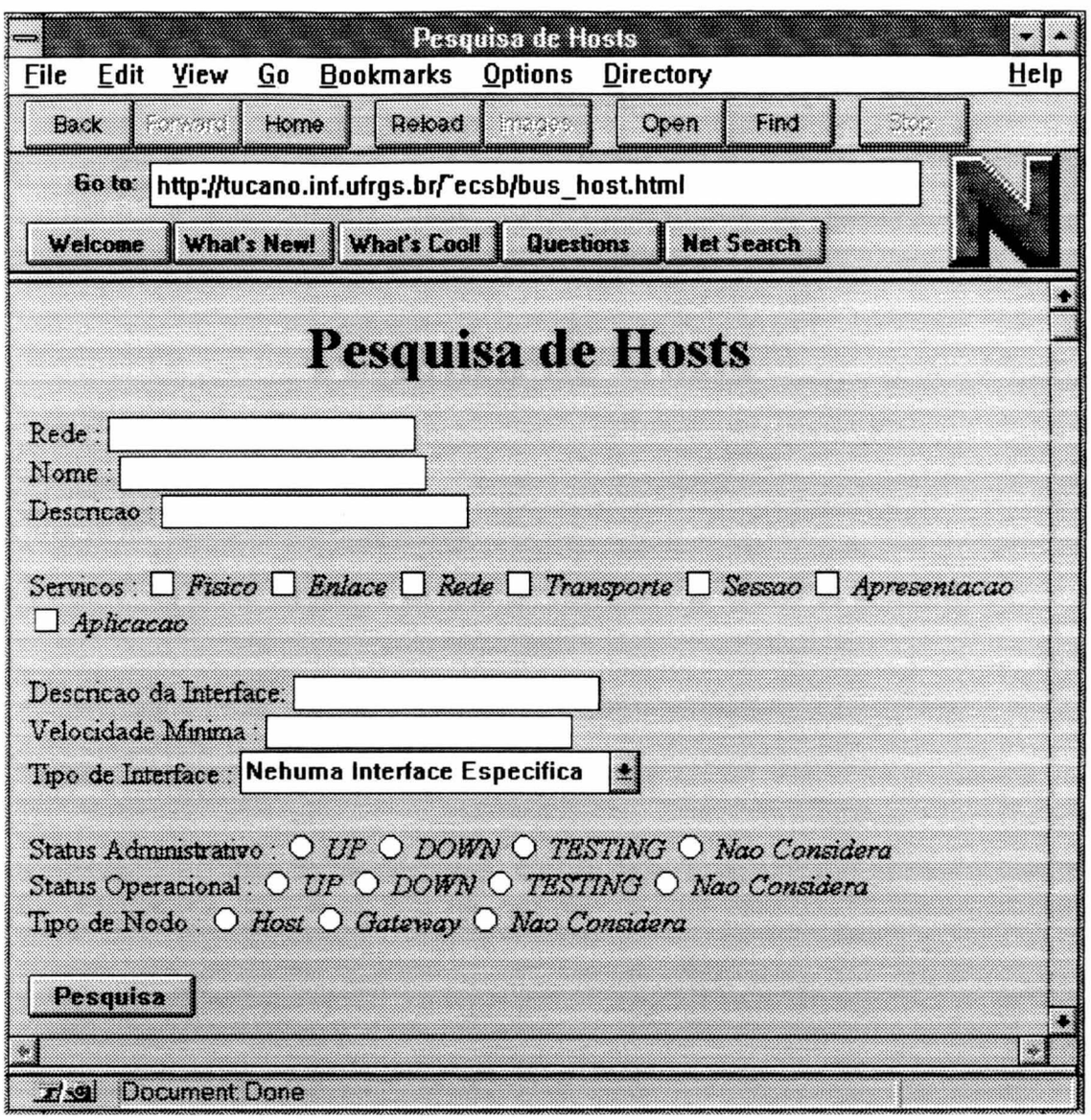

Figura 5.6 : Página WWW com as condições de busca de um dispositivo

A interface sobre a utilização da rede mostra os seus dados classificados por dia ou por dispositivo. Para tanto, será necessário que se tenha uma tela que pergunte o tipo de consulta desejada e os seus parâmetros. Na classificação por dia é mostrado todos os dispositivos que foram detectados num determinado dia escolhido pelo usuário. Já na classificação por dispositivo, o usuário escolhe um determinado dispositivo e o sistema mostrará todos os dia/hora em que esse dispositivo já foi detectado(Figura 5.7, 5.8 e 5.9).

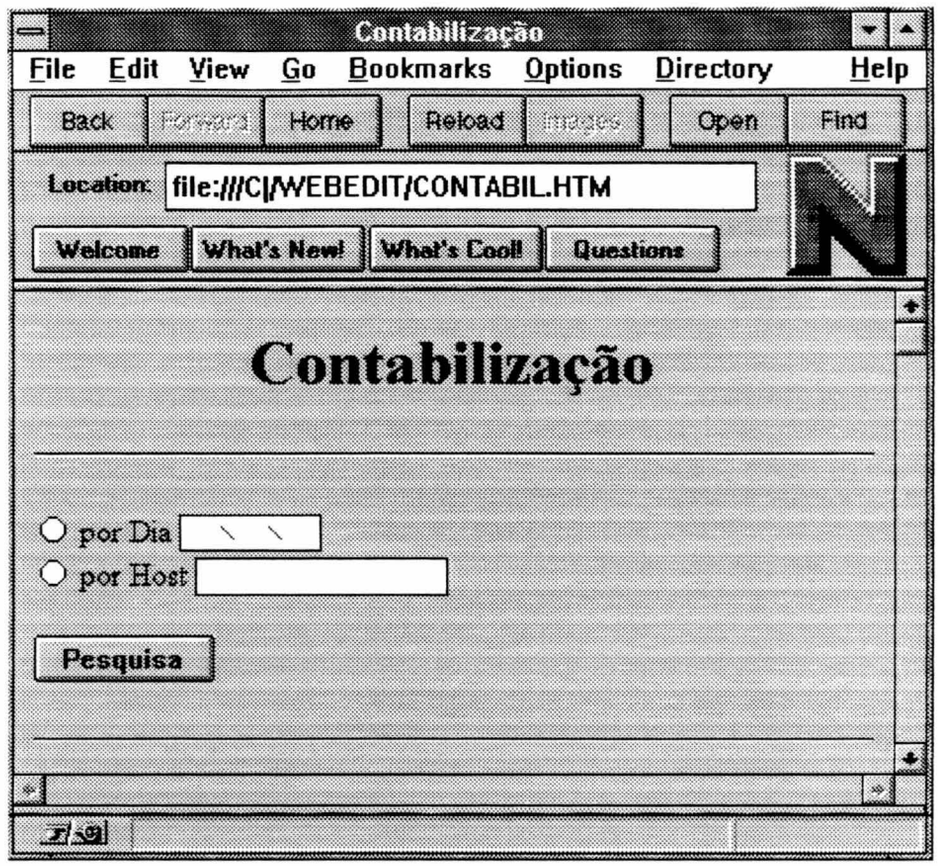

Figura 5.7 : Página WWW para escolha do modo de contabilização

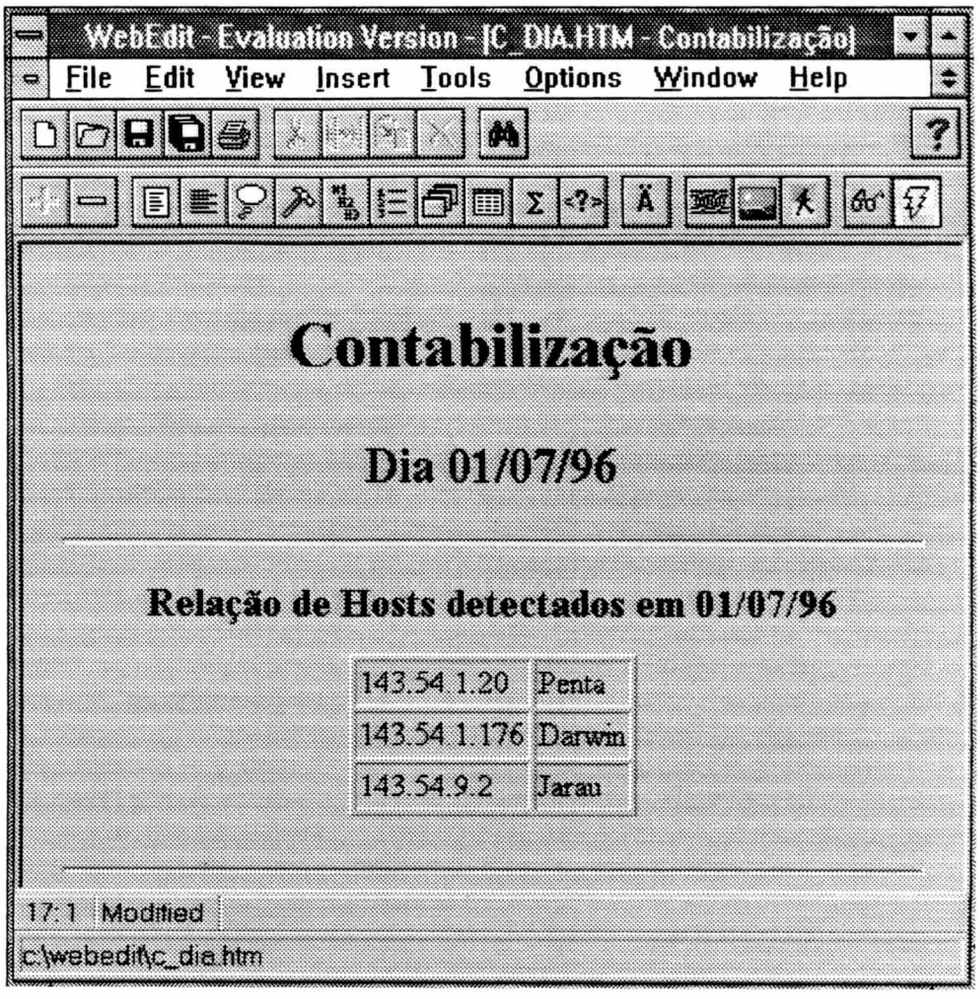

Figura 5.8 : Página WWW de contabilização por Dia

Figura 5.9 : Página WWW de contabilização por Host

Finalmente, a interface gerencial se divide em três módulos : de configuração, de manutenção e de Relatórios. O primeiro módulo, representado pela figura 5.10, permite que o administrador selecione a forma como o seu ambiente deve se comportar. Estão disponíveis as seguintes opções :

- Faixa de Domínio : Possui um endereço IP mínimo e máximo a ser aceito pelo WatchDog. Quaisquer outros endereços que não se encontrem dentro desta faixa serão ignorados pelo sistema.
- Número máximo de Processos ativos : Especifica qual o grau de paralelismo que o algoritmo poderá assumir.
- Tempo máximo de ping : Especifica o tempo limite de espera que o WatchDog esperará por um pedido de ping, conforme explicado mais à frente.

O segundo módulo é responsável pela exclusão de nodos cadastrados no WatchDog(figura 5.11). A exclusão pode ser feita tanto a nível de host, onde o administrador especifica qual o nome do nodo que ele deseja excluir, como de toda a base de dados. Caso a exclusão seja global, é necessário que se marque esse desejo numa checkbox, a fim de evitar exclusões acidentais.

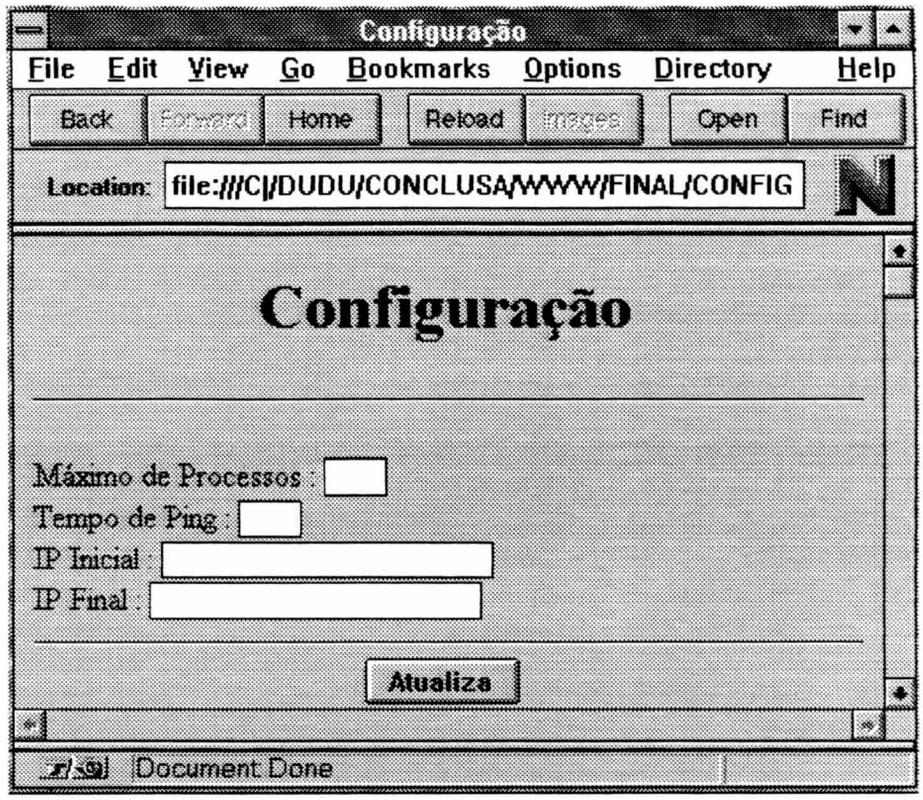

Figura 5.10 : Página WWW para configuração do WatchDog

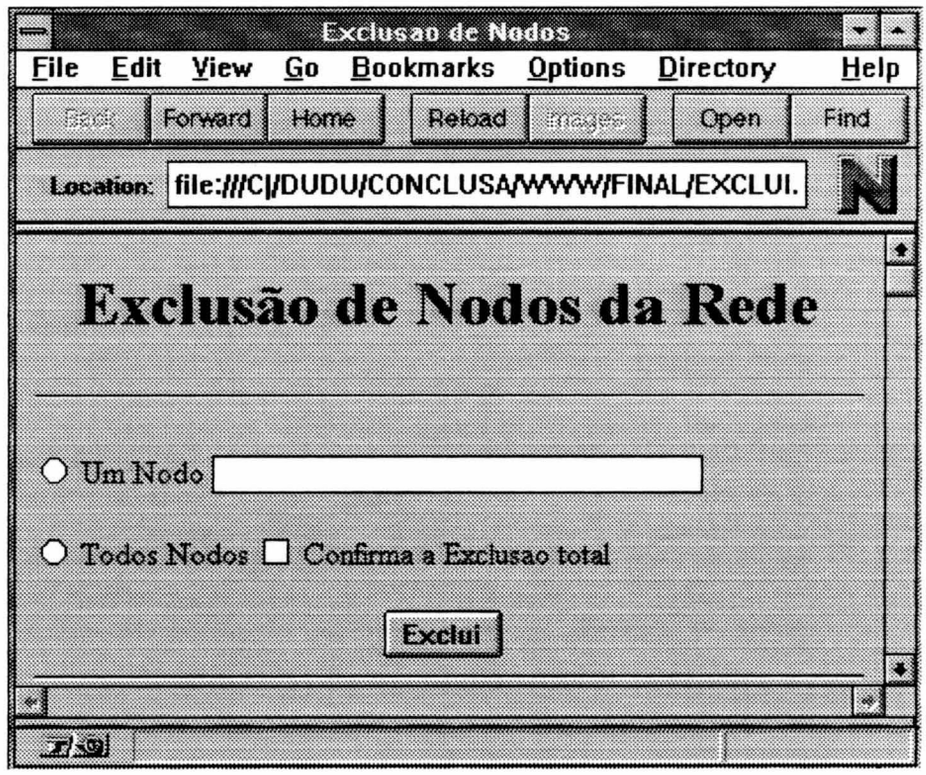

Figura 5.11 : Página WWW de Exclusão de Nodos

O terceiro módulo possui um relatóriocom todas as alterações encontradas pelo WatchDog na última vez em que ele foi executado. As modificações estão agrupadas por rede. Estão relacionados aqui :

a) Alterações encontradas na configuração de um Host,

b) Alterações encontradas na configuração da interface de um Host,

c) A detecção de nodos hosts na rede,

d) A detecção de novas interfaces em um dispositivo,

e) A falta de comunicação com certos nodos da rede a um período considerável de tempo.

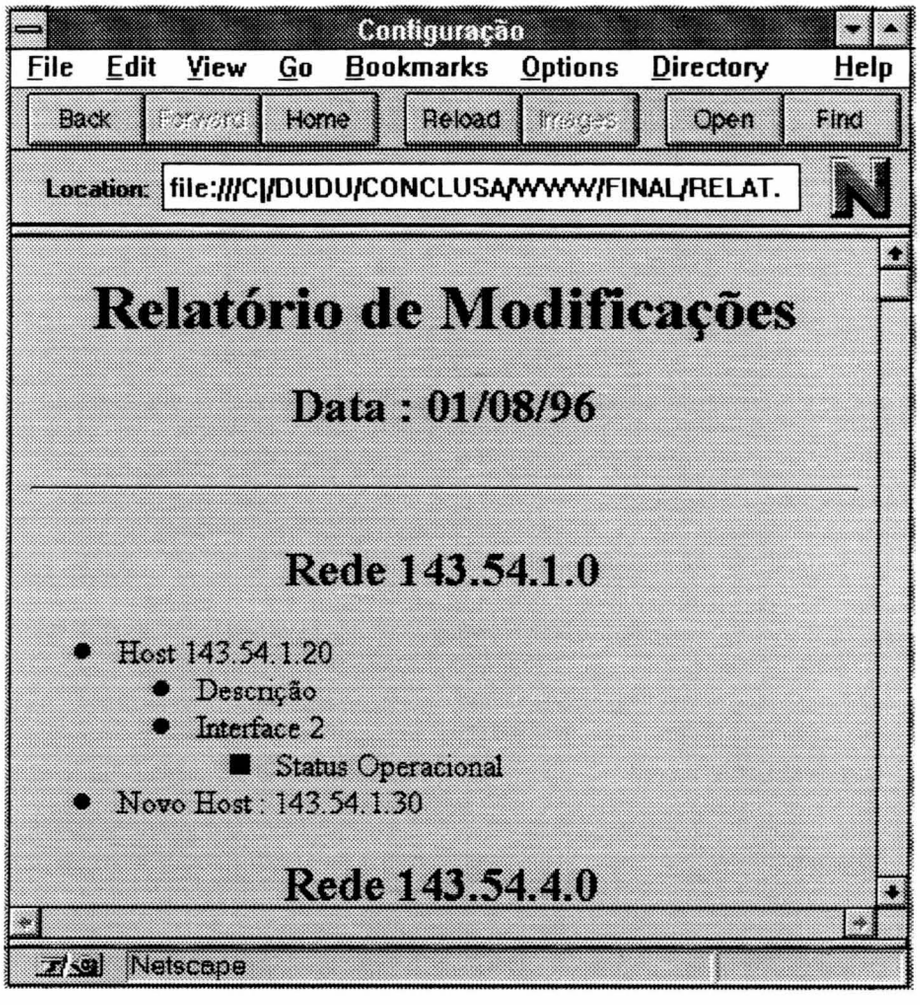

Figura 5.12 : Página WWW com Relatório de modificações

A figura 5.13 mostra um diagrama com a hierarquia e ordem de ativação das interfaces do sistema. As setas que saem de cada página mostram os caminhos que se pode percorrer na navegação entre as páginas, onde os retângulos sombreados indicam páginas e os círculos referenciam uma página que já apareceu no diagrama. O diagrama mostra o título de cada página, a forma como a página é criada (através de um arquivo HTML ou como resultado de um processo cgi) e o seu número de referência. A página que se encontra mais no topo do diagrama serve como apresentação do sistema e interliga os dois módulos principais, de contabilização e de consulta a rede, dentro do sistema.

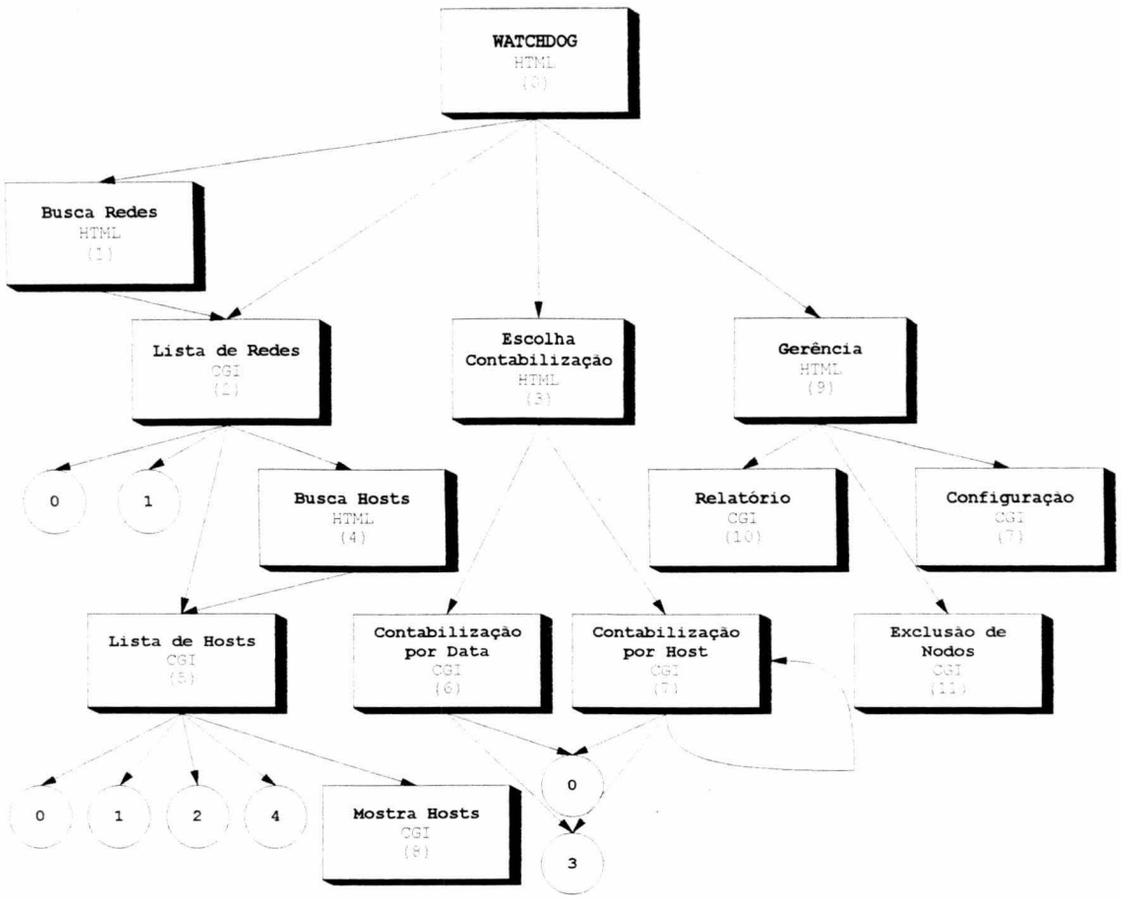

Figura 5.13 : Ordem de ativação das páginas WWW do WatchDog

Como pode-se notar, existem dentro do sistema cinco programas cgi que realizam a consulta a base de dados e montam dinamicamente as páginas html com o resultado da pesquisa. De uma maneira geral, os programas cgi podem ser divididos em três classes : programas de consulta simples, programas com utilização de consultas elaboradas e programas para a execução de tarefas.

Na primeira classe têm-se os dois programas de contabilização, o programa que detalha as informações sobre um determinado host e os programas que listam os dispositivos e as sub-redes, quando utilizados sem nenhuma condição de busca. Para montar a página com o resultado da pesquisa, os programas cgi primeiramente obtém os parâmetros indispensáveis enviados pelo form. O sistema a seguir monta o cabeçalho da página html e realiza uma consulta sobre a base de dados, formatando e retornando o seu resultado ao WWW. Finalmente, ele monta o rodapé da página, onde são disponibilizados os botões ao usuário que permitem que ele realize novas consultas.

Na segunda classe têm-se novamente os programas de listagem de sub-redes e de dispositivos, só que desta vez utilizando expressões de busca que realizam uma consulta elaborada sobre a base de dados. O fato desses dois programas poderem ser disparados de páginas de classes distintas obriga a eles terem uma dupla funcionalidade, tomando atitudes diferentes conforme for a classe de quem o disparou.

No caso da listagem de sub-redes, pode-se ter duas situações distintas :

a) O seu chamador foi uma tela de consulta elaborada, : o programa cgi deve analisar os atributos de entrada e montar um filtro que encontre somente as sub-redes que atendam aos requisitos solicitados pelo usuário.

b) O seu chamador foi uma página de consulta simples : o programa cgi deve listar todas as sub-redes existentes.

Já no caso da página de listagem de dispositivos, pode-se ter as seguintes situações :

a) O seu chamador foi uma tela de consulta elaborada : o programa cgi deve mostrar somente os dispositivos que atendem aos requisitos pedidos pelo usuário, mostrando a rede na qual faz parte cada dispositivo mostrado.

b) O seu chamador foi uma tela de consulta simples : o programa cgi neste caso recebe como entrada o endereço IP de uma rede e mostra todos os dispositivos que fazem parte da rede requisitada.

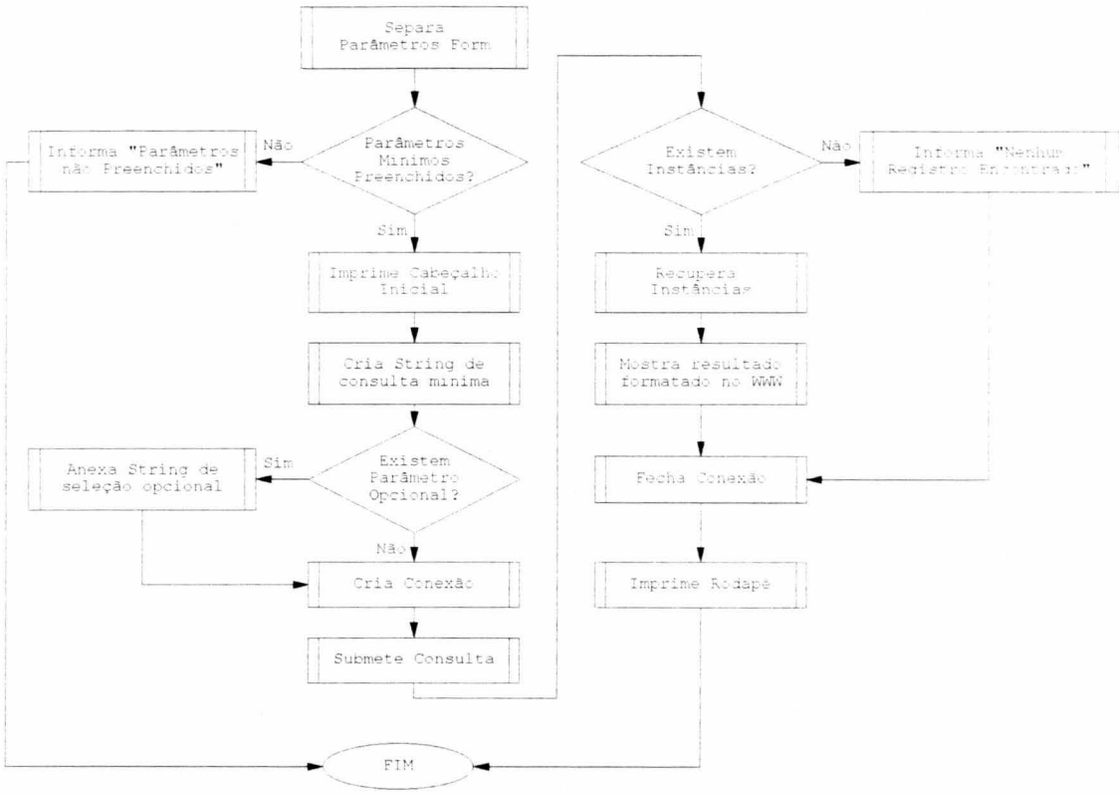

Figura 5.14 : Algoritmo para geração das páginas WWW

Os passos seguidos pelos programas dessa segunda classe são bastante semelhantes aos passos da primeira classe. A principal diferença são os parâmetros opcionais que são passados nessa classe e a dependência do procedimento de montagem da expressão de busca em relação a entrada que foi submetida ao usuário. A figura 5.14 mostra um algoritmo generalizado para a montagem das páginas html. Nele, a página em HTML está dividida em três partes : Cabeçalho, onde estão todas as informações anteriores aos dados da consulta como título, descrição do contexto, etc... ; os dados resultantes da consulta, que podem estar tanto em forma de texto como contidos numa listbox; e o rodapé, que possui algum comentário necessário e os botões para navegação entre as páginas. O termo impressão é utilizado para expressar o envio de informações ao browser do cliente. A consulta mínima é tratada como sendo a seleção de registros utilizando somente os parâmetros obrigatórios, enquanto que a consulta opcional faz a seleção utilizando todos os atributos possíveis.

A implementação das páginas de consulta sobre a utilização da rede possui uma estrutura de funcionamento muito parecida com as anteriores. Apesar destas páginas possuírem consultas bem menos complexas, envolvendo no máximo duas entidades

(Dispositivos e Contabilização), elas devem suportar quatro tipos diferentes de consulta :

a) Consulta de histórico por máquina, disparada através da página de seleção da contabilização,

b) Lista de máquinas ativas num determinado dia, disparada através da página de seleção da contabilização,

c) Lista de máquinas ativas no dia seguinte ao que foi passado como parâmetro, disparada através do botão de avanço da página de lista de máquinas ativas, e

d) Lista de máquinas ativas no dia anterior ao que foi passado como parâmetro, disparada através do botão de retrocesso da página de lista de máquinas ativas.

Todas as quatro consultas possuem um mesmo padrão de chamada, composto por um flag indicando a opção selecionada e um parâmetro de referência para a consulta.

A criação de todos esses programas descritos acima se deu através da união de três tecnologias distintas : Postgres, C e WWW.

A ligação do Postgres com o C se dá de forma fácil, uma vez que o próprio Postgres disponibiliza uma API para sua interligação com C. Essa API é composta por funções simples que não exigem grandes conhecimentos para serem compreendidas. Apesar da documentação disponibilizada pelo Postgres para a sua conexão com o C ser bastante sucinta, ela está muito bem descrita e possui alguns programas exemplos no seu final que tornam bem mais fácil a compreensão do mecanismo de comunicação.

A API disponibiliza funções para conexão, execução de comandos e encerramento de uma sessão. Todas as funções executadas em C retornam com algum tipo de flag que indica qual foi o resultado da execução, permitindo assim ao programador decidir quais serão os seus próximos comandos.

A troca de dados entre o C e o Postgres se dá através de uma estrutura de dados chamada de Portal, que não passa de um buffer gerenciado pelo Postgres onde as instâncias de consultas submetidas pelo usuário são disponibilizadas. Esses portais são utilizados somente por funções de consulta a base de dados que retornam instâncias do banco de dados. Em cima destes portais existe um conjunto de funções que manipulam os dados disponibilizados, permitindo ao usuários recuperar as suas informações das mais variadas maneiras.

A comunicação entre o C e o WWW é da mesma forma fácil de ser implementada. As informações disponibilizadas através da internet, bem como o tutorial difundido dentro da UFRGS, explicam claramente como deve ser feito o processo de conexão entre o C e o WWW. A implementação de páginas dinâmicas via programas cgi não exige nenhum conhecimento muito aprofundado sobre o funcionamento da internet ou de servidores WWW. Uma vez seguidos os passos descritos no tutorial, a obtenção de parâmetros de entrada digitados no formulário é feita de forma automática, e o envio das informações para o browser do cliente é feito de modo transparente através da saída padrão do C (stdout - standard out).

Apesar de ser fácil a implementação da comunicação entre C e Postgres e entre C e WWW, a união destes dois serviços simultaneamente cria alguns problemas. O primeiro problema diz respeito à relação entre quem executa o processo e os seus direitos de acesso dentro do Postgres. Dentro de um programa com ligação somente entre Postgres e C, o usuário que dispara esse sistema deve estar cadastrado no Postgres com direitos de acesso suficiente para consultar e talvez manipular as tabelas utilizadas pela aplicação. Desta forma, para uma pessoa de user britto no unix possuir uma conta no Postgres, deve existir no Postgres um usuário de nome também britto e que possua

direitos sobre as tabelas que a aplicação que deseje rodar utilize para que se possa utilizar esta aplicação sem problemas.

No caso de um programa cgi que utilize o Postgres, a aplicação que realiza consultas à base de dados é disparada através de uma consulta no WWW feita por um usuário que pode estar conectado a um computador em qualquer parte do mundo. Dentro desse ambiente, o processo cgi que executa as operações desejadas não é mais de propriedade do usuário que solicitou a consulta, e sim de um usuário determinado pelo servidor WWW como dono do processo cgi. Desta forma, deve-se descobrir qual o nome do usuário que o servidor de WWW utiliza para disparar um programa cgi. De posse desta informação, deve-se cadastrar este usuário no Postgres e dar os direitos de acesso necessários para que a aplicação realize as suas atividades.

No nosso caso, o gerente de configuração foi instalado inicialmente no servidor da máquina tucano da UFRGS, localizada no instituto de informática. Este servidor dispara os seus processos cgi através do usuário nobody. Logo, para o gerente funcionar, deve-se cadastrar um usuário de nome nobody no Postgres e dar direitos de consulta sobre as tabelas do gerente. O sistema cgi será executado seguinte os seguintes passos :

a) O usuário clica num botão de uma página WWW que dispara um programa cgi.

b) O servidor WWW cria um processo cgi através do usuário nobody.

c) O programa cgi pede conexão com o Postgres.

d) O Postgres verifica se nobody está cadastrado como usuário e, caso esteja, cria uma sessão de conexão com o cgi.

e) O cgi pede para consultar/atualizar alguma tabela.

f) O Postgres verifica se nobody possui direitos suficientes para realizar as operações pedidas sobre aquela tabela e, caso possua, permite que a operação seja realizada.

Um grande problema que surge nesse mecanismo é o da segurança. Para permitir que o programa cgi possa realizar alguma operação sobre uma tabela, deve-se liberar os direitos de executar essa ação sobre a tabela para o usuário determinado pelo servidor WWW. Isso permitirá que qualquer programa cgi irá tenha direito de acesso a esses dados, tornando o sistema extremamente vulnerável. Quando esses direitos são somente de consulta, não existem maiores problemas se as informações não forem confidenciais. Porém, quando esses direitos são também de atualização, corre-se o risco de se perder a confiabilidade dos dados armazenados.

Outro problema que surge com a integração entre Postgres e WWW é a localização do SGBD. Uma vez que o processo disparado via WWW rodará na máquina servidora de WWW, deve-se prever um mecanismo que permita rodar o processo cliente de banco de dados em uma máquina diferente de onde está o servidor, ou então instalar o banco de dados na própria máquina do servidor de WWW.

O Postgres permite que processos remotos sejam conectados a ele. Para isso, basta setar no processo cliente o valor de uma variável chama PQhost com o nome da máquina servidora, e noutra chamada PQport o número da porta nessa máquina. No caso deste trabalho, o Postgres está residente na máquina vega (PQhost = "vega") e utiliza a porta número 1234 (PQport = "1234").

5.2 Processo WatchDog

O WatchDog é o processo dentro do sistema que preenche o banco de dados do gerente com as informações da rede. Ele se divide em três tarefas distintas :

a) Detecção das sub-redes presentes e seus dispositivos,

b) Obtenção da configuração de cada sub-rede e dispositivo detectado, e

c) Cadastramento dos novos dispositivos e sub-redes encontrados bem como a atualização dos dados já existentes na base de dados.

A primeira tarefa é sem dúvida a mais complexa, e exige um estudo mais aprofundado sobre a melhor maneira de se detectar sub-redes e dispositivos na rede. Existem várias alternativas para se realizar esta tarefa, e cada alternativa apresenta um conjunto de ferramentas que podem ser utilizadas, tais como consultas SNMP, a utilização de ping's ou o uso de arquivos que se encontram nos dispositivos e que possuem algumas "dicas" sobre as sub-redes e nodos da rede.

A segunda tarefa possui um nível de complexidade menor, e consiste em um conjunto de consultas à MIB que visa o recolhimento de dados sobre cada dispositivo presente na rede e sua análise, a fim de se montar a estrutura da rede como um todo, obedecendo a hierarquia rede -> sub-rede -> dispositivo -> interface. Além disso, essa tarefa também exige que alguns valores obtidos pela MIB sejam convertidos, de forma a se tornar uma informação clara para o usuário final que consultará o gerente através do WWW.

Finalmente, a terceira tarefa é a que menos envolve aspectos de gerência de redes e entra um pouco mais a fundo em aspectos de banco de dados, tais como a verificação de quais informações são novas para o gerente e quais estão desatualizadas, aspectos de concorrência na inclusão dos dados por processos que estão executando paralelamente, etc...

5.2.1 Detecção dos dispositivos presentes

Esta fase consiste em descobrir quais dispositivos e sub-redes estão presentes na rede que está sendo pesquisada. Existem diversas maneiras de se realizar esta detecção, e cada maneira possui o seu grau de confiabilidade, suas próprias ferramentas de busca e suas vantagens e desvantagens.

Serão avaliados neste trabalho duas alternativas e feita a opção ao seu final por uma delas.

5.2.1.1 Detecção por comunicação

Cada dispositivo presente na rede possui informações que permitem que se saiba quem ele é, onde ele está conectado, quais caminhos ele utiliza para se conectar, etc... Estas informações estão guardadas em um conjunto de tabelas que possuem funcionalidades distintas dentro do dispositivo. As seguintes tabelas são de vital importância para a detecção de dispositivos e sub-redes :

- Tabela de Interfaces : Possui informações sobre todas as interfaces correntes do dispositivo, quantas são, quais suas características físicas e configurações, qual o seu estado corrente, etc... Através dela pode-se descobrir quais são os meios que o dispositivo tem para se comunicar com o mundo exterior.
- Tabela de Endereços IP : Esta tabela fornece informações sobre todos os endereços IP do dispositivo, tais como o seu número IP, a interface responsável por aquele endereço, o endereço de rede daquele IP, seu endereço de broadcast, etc... O endereço IP de um dispositivo informa onde ele se encontra conectado. Através desta tabela tem-se condições de ver todas as sub-redes na qual o dispositivo está conectado. Como um dispositivo pode estar conectado em mais de uma sub-rede, pode-se chegar

em um dispositivo através de uma determinada sub-rede e descobrir através dele novas sub-redes.

• Tabela ARP (Address Resolution Protocol) : Esta tabela armazena um conjunto de endereços IP conhecidos pelo dispositivo e o endereço físico correspondente a esses endereços IP. As informações armazenadas por esta tabela são obtidas através do protocolo ARP.

O protocolo ARP visa solucionar o problema que surge quando uma máquina deseja se comunicar com outra máquina na qual ela só conhece o endereço IP. Para descobrir o endereço físico desta máquina, o nodo de origem emite um pacote de broadcast na sub-rede perguntando qual o endereço físico do endereço IP requisitado. A máquina que possui esse endereço recebe a mensagem e envia como resposta um par (End-IP, End-Físico), permitindo assim a comunicação entre as duas.

Como um pacote de broadcast possui um custo alto para a rede, cada dispositivo mantém uma tabela ARP em memória cache onde é guardada uma lista com todos os últimos endereços IP adquiridos e seus endereços físicos equivalentes, a fim de evitar o uso do protocolo ARP repetidamente.

Uma consulta à tabela é um meio prático e eficiente de se descobrir nodos conectados à rede recentemente.

• Tabela de Roteamento : A troca de mensagens entre dispositivos pode ser feita de maneira direta, quando eles se encontram numa mesma rede física, ou de maneira indireta, quando eles se encontram em redes distintas. O roteamento direto de informação é resolvido facilmente através do encapsulamento de um datagrama em um frame físico que mapeia o endereço IP destino em um endereço físico com ajuda do protocolo ARP.

Já o roteamento indireto é mais difícil, uma vez que o nodo origem deve enviar o pacote a um gateway que faça parte de uma rota de nodos na qual o pacote deverá passar para chegar ao seu destino. O pacote deverá então percorrer uma série de gateways até alcançar um que esteja na mesma sub-rede do seu nodo destino e portanto enviar o seu pacote diretamente.

A fim de realizar essa operação de roteamento entre os dispositivos, cada host ou gateway possui uma tabela de roteamento onde são armazenados informações sobre possíveis nodos ou sub-redes destinos e qual é o próximo gateway da rota para qual o pacote deve ser enviado para alcançá-lo.

Esta tabela de roteamento acaba se tornando uma excelente fonte de informação a respeito das sub-redes que compõem a rede que está sendo pesquisada. Através dela é possível descobrir tanto novas sub-redes através da informação de destino da rota, como novos gateways através do próximo nodo da rota sendo percorrida.

É através destas tabelas que a primeira alternativa se embasa para detectar a estrutura da rede. Ela parte do princípio que todo o nodo que compõem a rede já se conectou alguma vez com outro nodo, tendo portanto um registro na tabela ARP ou de rotas de alguma das máquinas visitadas.

A coleta de informações da rede começa através da escolha de um gateway qualquer da rede. Uma vez que os gateways são os principais dispositivos de comunicação de uma rede por assumirem o papel de roteadores de pacotes, o algoritmo irá visitar cada gateway descoberto a fim de obter novas informações sobre a rede. O algoritmo armazena cada dispositivo descoberto na rede em duas listas : LstGate, que armazenará os gateways descobertos, e LstHost, que armazenará os hosts descobertos. As sub-redes, por sua vez, são guardadas numa lista chamada LstRede.

Para realizar a detecção, o algoritmo inicialmente insere o gateway recebido como parâmetro de entrada na sua lista. Após, ele entre num loop onde cada interação analisa um gateway buscando novos componentes.

Para cada gateway descoberto, são verificadas as sub-redes a qual o gateway pertence e as sub-rede que não possuem registro ainda são inseridas em LstRede. Após, é analisada a tabela ARP do gateway e para cada nodo encontrado é verificado se ele é um host ou um gateway, inserindo-o na lista correspondente caso ele não tenha sido incluído ainda.

A próxima tabela a ser analisada é a de Rotas, onde se descobrirá novos endereços de rede através do destino de cada rota (ip.ipRouteTable.ipRouteEntry.ipRouteDest, ip.ipRouteTable.ipRouteEntry.ipRouteMask), bem como novos nodos gateways, através da informação de próximo nodo (ip.ipRouteTable.ipRouteEntry.ipRouteNextHop).

O algoritmo faz a análise destas tabelas para cada gateways que ele encontra, conseguindo ao final do algoritmo um conjunto de listas contendo todos os host, gateways e sub-redes encontrados. A figura 5.15 mostra os passos traçados pelo algoritmo.

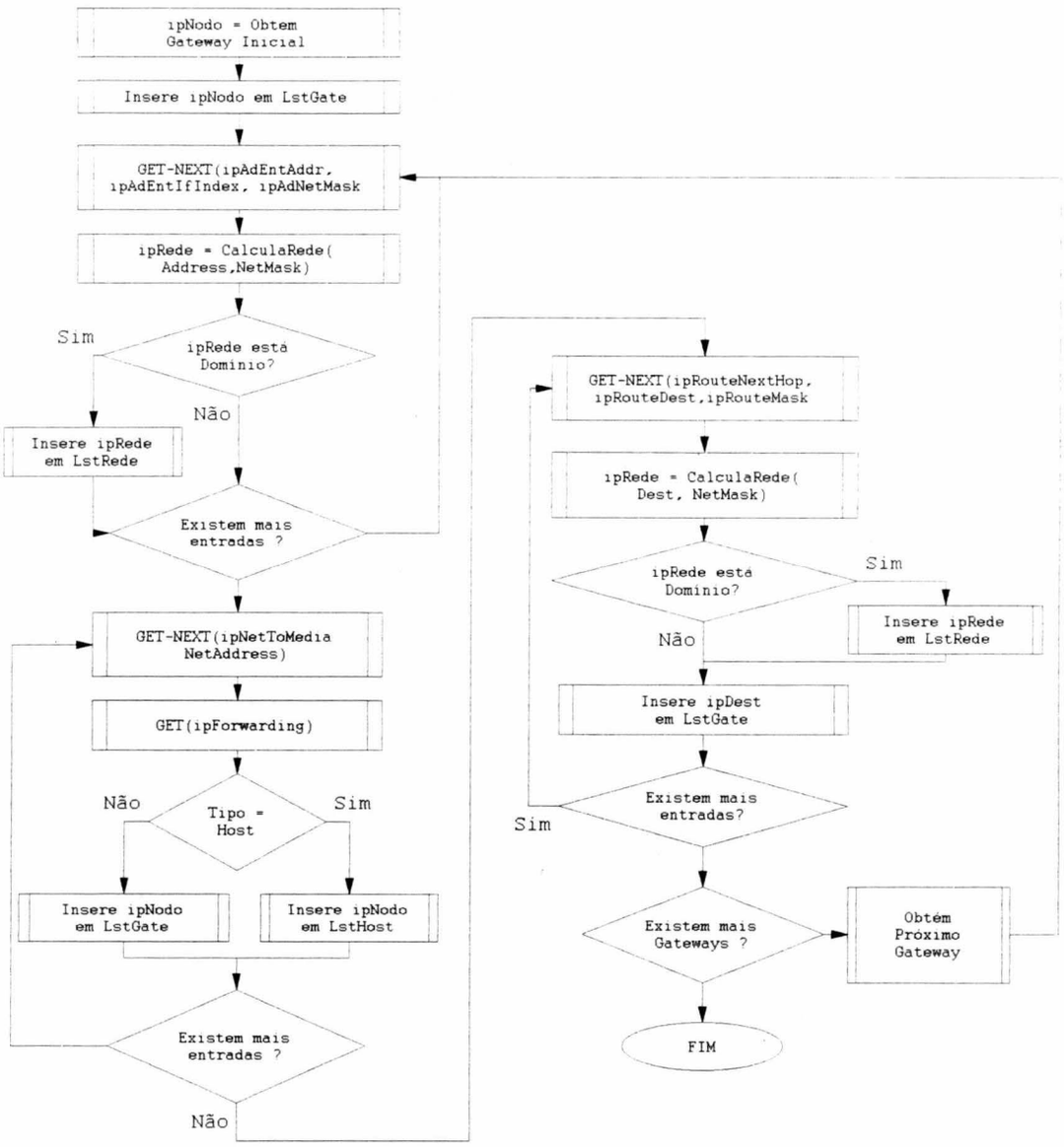

Figura 5.15 : Algoritmo de Detecção por Comunicação

Por se basear somente nas tabelas dinâmicas dos dispositivos para montar a sua rede, diz-se que esse algoritmo detecta os dispositivos com base nas necessidades de comunicação que os mesmos tiveram. Ele parte do princípio que, se um dispositivo está conectado à rede, é porque ele possui uma necessidade freqüente de comunicação.

Porém, nem sempre isso é verdade. Dispositivos que estão ligados a rede, mas que estão sendo utilizados somente para realizar processamento local e que não necessitam de nenhum recurso ou arquivo de outros equipamentos, não apareceram nas tabelas dinâmicas, não sendo portanto detectados pelo algoritmo.

5.2.1.2. Detecção por Comunicação e Ping

A segunda alternativa tenta resolver os problemas da primeira, não se embasando tanto nas tabelas e tentando descobrir por si só o que realmente existe na rede. Para isso, ela utiliza um mecanismo de obtenção de informações semelhante ao da alternativa anterior, porém ao invés de se embasar na lista de hosts, gateways e sub-redes, e ele somente utiliza as sub-redes encontradas como ponto de partida. Cada sub-rede detectada será percorrida atrás de nodos ativos através de ping's a todos os endereços possíveis dentro daquela sub-rede, encontrando assim todos os nodos da mesma que estavam ligados no momento em que o algoritmo executou, inclusive aqueles que não se encontram presente em nenhuma tabela de dispositivo.

Para implementar este mecanismo, o algoritmo continuará utilizando um loop que busca informações em todos os gateways conhecidos, porém as informações armazenadas serão somente as sub-redes e os gateways encontrados. O armazenamento dos gateways é de caráter temporário, utilizando-os somente como instrumento de busca de novas informações. Uma vez que não é mais necessário cadastrar os hosts, a consulta a tabela ARP não é mais realizada.

Após analisar todos os gateways, o algoritmo irá dar ping's em todos os endereços possíveis de cada sub-rede, e a cada ping que receber resposta será criada uma entrada na lista de hosts ou de gateways, conforme o caso, registrando o dispositivo encontrado(Figura 5.16).

Desta forma, ao final deste algoritmo se terá as mesmas três listas do algoritmo anterior, porém com a grande vantagem desta abordagem ter uma maior confiabilidade na completude dos dados encontrados.

Porém toda vantagem têm um custo, e neste caso este custo é de tempo de processamento. A busca da existência de um nodo pode ser uma procedimento demorado. Muitas vezes falta de resposta a um ping não se deve a não existência do nodo procurado, e sim a uma demora excessiva no recebimento da resposta do nodo questionado. Essa demora pode ser ocasionada devido diversos fatores, como o congestionamento da rede, o número elevado de nodos intermediários, etc...

Logo, o tempo de espera pela resposta do nodo é um fator crítico do algoritmo. Quanto mais tempo se esperar pela resposta, mais confiável será o algoritmo, porém maior também será o tempo que ele levará para terminar. Logo, a escolha deste tempo de espera vai depender das necessidades de confiabilidade e de tempo que se possui. O tempo default de espera utilizado em um ping é de 20 segundos.

Para visualizar melhor este problema, basta analisar um exemplo típico da rede da UFRGS. Nela, cada sub-rede costuma ter uma média de 254 endereços de hosts possíveis. O tempo necessário para percorrer essa rede, considerando que existem por exemplo uma média 20 nodos por sub-rede, seria de (234 nodos ausentes) x (20 segundos por nodo) = 4680 segundos = **1 hora e 18 minutos**.

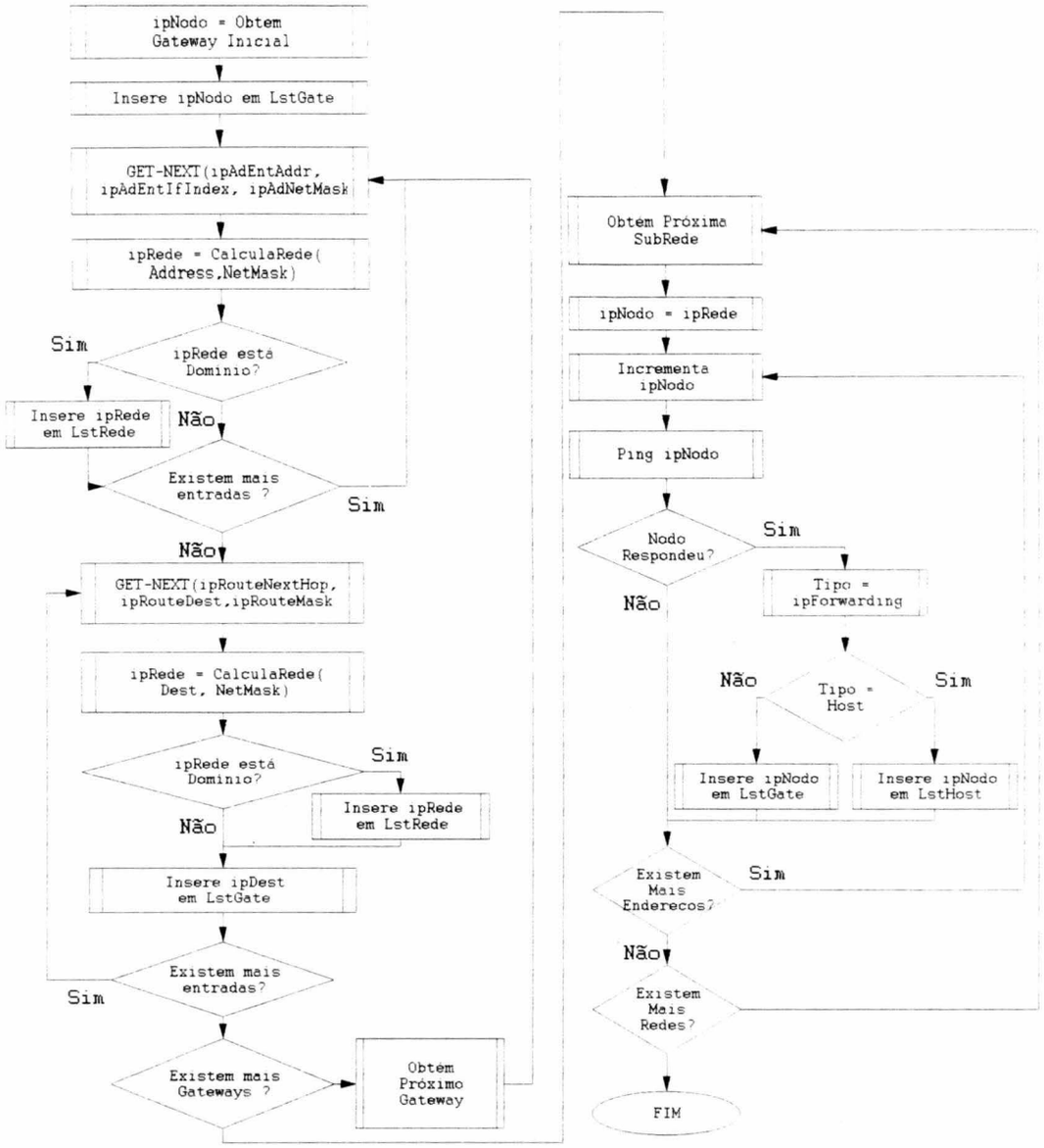

Figura 5.16 : Algoritmo de Detecção por Comunicação e Ping

Esse tempo calculado é apenas utilizado para detectar quais nodos não se encontram na sub-rede. Não estão sendo considerados outras fases que despendem uma quantidade razoável de tempo, tais como o tempo gasto para detectar os 20 nodos que realmente existem nessa sub-rede, para se realizar as consultas SNMP nesses dispositivos encontrados e para se fazer a consulta e atualização no banco de dados, bem como o tempo que é gasto no início do algoritmo para se identificar as sub-redes existentes.

Considerando que esse tempo é utilizado somente para analisar uma sub-rede, a análise de uma rede com 15 sub-redes, por exemplo, levaria um tempo 15 vezes maior (cerca de 20 horas), podendo assim conforme o caso levar alguns dias para detectar toda a rede desejada, e tornando desta forma esse algoritmo inviável(Figura 5.17).

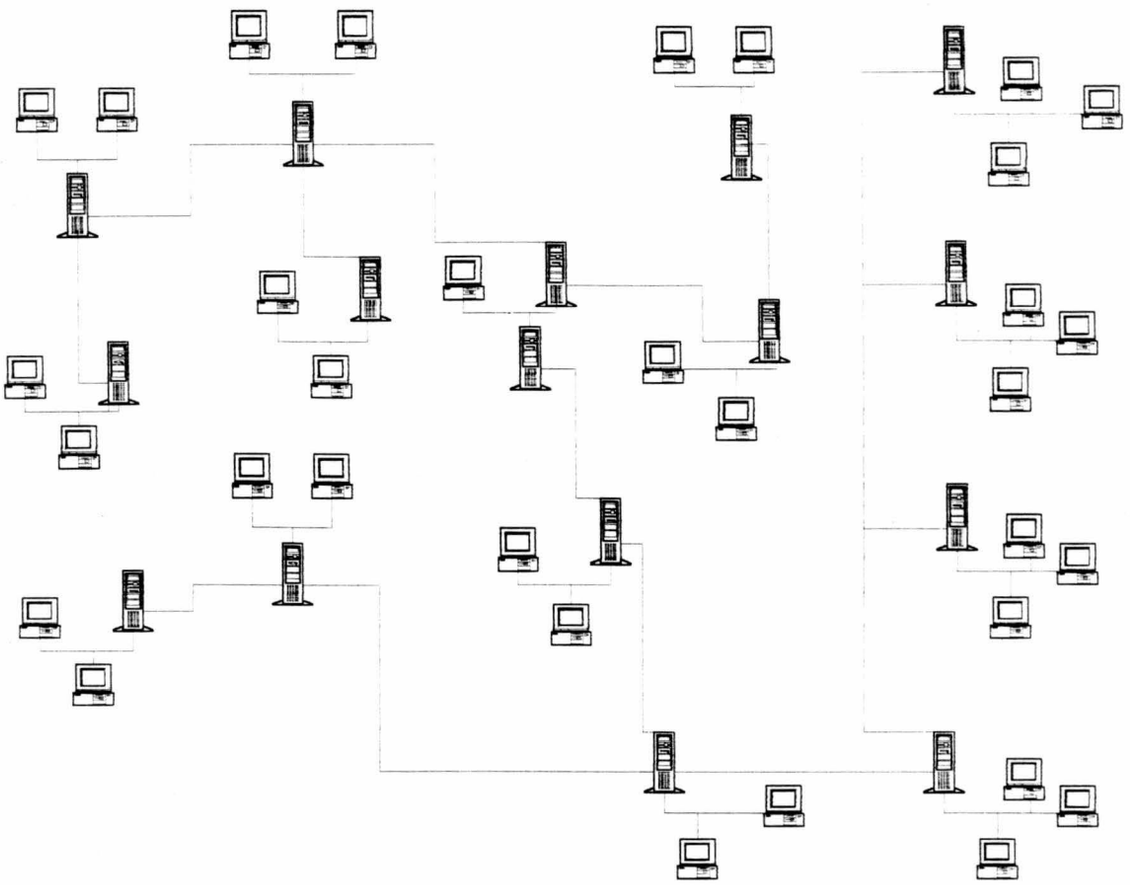

Figura 5.17 : Detecção de um Rede composta por 15 sub-redes.

Para amenizar esse problema, podemos aplicar duas estratégiasa que, quando empregadas juntas, tornam os tempos de respostas bem mais aceitáveis. A primeira, já discutida anteriormente, consiste em diminuir o tempo de espera. A implantação desta alternativa deve ser precedida de um estudo que avalie todas as variáveis relevantes na determinação do tempo de espera, tais como a abrangência que a consulta deve tomar (consultas a redes dispersas irão necessitar de um tempo de resposta maior), o tempo médio de resposta dos dispositivos e a confiabilidade desejada para o sistema.

Contudo, mesmo uma redução significativa sobre o tempo de espera não torna o tempo de processamento do algoritmo aceitável. A continuidade desse problema se deve à repetição realizada sobre o mesmo procedimento de detecção para cada sub-rede que o algoritmo deve percorrer. No exemplo acima, caso se reduzísse o tempo de espera para 3 segundos, ainda sim se teria um tempo de aproximadamente 12 minutos para percorrer cada sub-rede e de cerca de 9 horas para percorrer a rede como um todo (composta por 15 sub-redes).

Considerando que a detecção de cada sub-rede independe das demais, uma solução para esse problema é executar os processos de detecção de forma paralela. Desta maneira, cada sub-rede será analisada por um processo diferente que roda simultaneamente com os demais. Tendo em vista que o algoritmo de detecção não possui um processamento pesado, uma vez que ele passa a maior parte do tempo esperando por respostas de possíveis nodos, o disparo de um processo para cada sub-rede que se deseje percorrer não acarretaria num grande overhead para o processador.

Seguindo o exemplo, após detectar a existência de 15 sub-redes, o gerente dispararia 15 processos paralelos que analisariam essas 15 sub-redes, e ao final de um pouco mais de 12 minutos, levando em consideração o esforço extra do processador em

gerenciar estes quinze processos simultaneamente, teria todas as informações sobre a rede.

5.2.2 Obtenção da Configuração

Uma vez descobertos quais são os dispositivos existentes na rede, o próximo passo é obtenção dos dados a respeito da configuração destes dispositivos. Esses dados são obtidos através do uso de consultas SNMP get e get-next, que permitem que o gerente percorra a MIB de cada dispositivo e obtenha os objetos desejados.

O processo de coleta de dados de um dispositivos consiste realizar consultas à MIB sobre os objetos previstos pelo modelo. Alguns destes objetos necessitarão ser convertidos de modo que o seu conteúdo seja mais claro para o usuário. Existem também alguns dados que são obtidos através da análise de um conjunto de objetos retornados.

A seguir serão analisadas as três principais entidades detalhadamente, estudando como é obtido cada um dos seus atributos.

a) Dispositivo

- Código - Valor obtido dinamicamente através de uma rotina da parte de banco de dados.
- Nome : Possui o nome lógico do dispositivo, é obtido a partir do objeto system.sysName.
- Descrição : Possui uma descrição do dispositivo, é obtido a partir do objeto system.sysDescr.
- IdObjeto : Possui um Id que identifica o fabricante, é obtido a partir do objeto system.IdObjeto.
- TimeUp : Indica a quanto tempo o dispositivo se encontra operacional. É obtido a partir do objeto system.sysUpTime, que armazena o valor em milésimos de segundo, necessitando por isso ser convertido para dias, horas, minutos e segundos.
- Localização : Indica aonde o dispositivo está localizado, obtido a partir do objeto system.sysLocalization.
- Serviços : Indica quais os níveis do modelo OSI que o dispositivo atende. Obtido a partir do objeto system.sysServices, o valor retornado da MIB é um número que representa a soma de valores de cada camada utilizando a fórmula $2^{(L-1)}$, onde L é o número da camada do protocolo. Esse número é analisado de forma a se montar uma string que enumere quais os serviços que estão disponibilizados pelo dispositivo.
- SNMP : Indica se o dispositivo fala ou não SNMP. Esse valor é obtido através da resposta ou não do dispositivo a consultas SNMP. Caso o dispositivo responda às consultas, é retornado o valor TRUE. Caso contrário, esse atributo recebe FALSE.
- Tipo : Indica se o dispositivo é um host(Valor H) ou um gateway(Valor G). Essa informação é obtida através do objeto ip.ipForwarding, e é de vital importância durante a execução do algoritmo para decidir se o dispositivo pode ou não fornecer maiores informações sobre a rede.
- nInterface : Número de interfaces do dispositivo. Essa informação é obtida através do atributo interface.ifNumber, podendo também ser derivada do número de interfaces retornados pelo dispositivo à consulta SNMP.

b) Interface

• CodMáquina - Código da máquina que possui a interface. Esse valor é obtido através de um valor do atributo Código de Dispositivo, que já terá sido calculado na hora de se incluir as interfaces, uma vez que a inclusão da interface se dará após a inclusão do dispositivo.

• Número : Índice da interface dentro do dispositivo. É um número seqüencial que vai de 1 a n, onde n é o total de interfaces do nodo. Não podem existir duas interfaces com o mesmo número para um mesmo dispositivo.

• IP : Endereço IP da interface, obtido através do objeto ip.ipAddrTable.ipAddrEntry.ipAdEntAddr. Para se descobrir a qual interface pertence o endereço IP armazenado neste objeto, se consulta o objeto ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex, que liga o endereço IP à interface que possui ifIndex igual ao número deste objeto. Podem existir interfaces que não possuem nenhum endereço IP ligados a elas, estando assim com este campo vazio.

• Físico : Endereço físico da interface, obtido através do objeto interfaces.ifTable.ifPhysAddress.

• Rede : Código da rede no banco de dados na qual o endereço IP está conectado. A descoberta de a qual rede o endereço IP pertence é feita através de um AND lógico entre o endereço IP da interface e a máscara de rede contida no objeto ip.ipAddrTable.ipAddrEntry.ipAdEntNetMask . Caso a interface não possua endereço IP, logicamente ela também não possuirá um código de rede.

• Descrição : Descrição textual da interface, obtido através do objeto interfaces.ifEntry.ifDescr .

• Tipo : Tipo de interface, distinguida através do protocolo de nível 1 e 2. O objeto interfaces.ifEntry.ifType possui um número que classifica a interface em um dos tipos pré-definidos na MIB, necessitando portanto que se converte esse valor antes de se incluir o nome do tipo dentro do banco de dados.

• MTU : Possui o tamanho em octetos do maior datagrama possível de ser enviado ou recebido pela interface, obtido através do objeto interfaces.ifEntry.ifMTU .

• Velocidade : Velocidade da interface em bits por segundo, obtido pelo objeto interfaces.ifEntry.ifSpeed .

• Status Administrativo : Indica o estado desejado para a interface, sendo obtido pelo objeto interfaces.ifEntry.ifAdminStatus . Ele está representado na MIB por números de 1 a 3, que são convertido para os estados up, down e testing.

• Status Operacional : Indica o estado corrente da interface, sendo obtido pelo objeto interfaces.ifEntry.ifOperStatus . Sofre o mesmo tipo de conversão descrito acima.

• Tempo Status : Indica a quanto tempo o dispositivo se encontra no status corrente. Obtido pelo objeto interfaces.ifEntry.ifLastChange .

c) Rede

• Código : Número seqüencial atribuído pelas rotinas de banco de dados para criar os relacionamentos.

• Endereço : Endereço IP da rede. Através deste endereço os dispositivos descobrirão a que rede pertencem e obterão o código da rede.

• NumHosts : Número total de hosts existentes na rede. Apesar de ser uma informação derivada que pode ser obtida através de um procedimento de contagem, seu valor é armazenado a fim de se otimizar a consulta aos dados de redes. A contagem é feita durante o algoritmo de detecção de hosts. Para cada dispositivo encontrado, é feito um teste para verificar se o dispositivo age como um host e, em caso afirmativo, é incrementado o contador de hosts encontrados naquela rede.

• NumGateways : Número total de gateways existentes na rede, obtido de forma semelhante a NumHosts.

• Classe : Classe da rede, obtido através de uma análise dos dois primeiros bits do endereço IP da rede, conforme tabela abaixo :

Tabela 5.1 : Obtenção da Classe de uma rede

bit 0	bit 1	Classe
0	-	A
1	0	B
1	1	C

A figura 5.18 mostra o processo de obtenção de configuração.

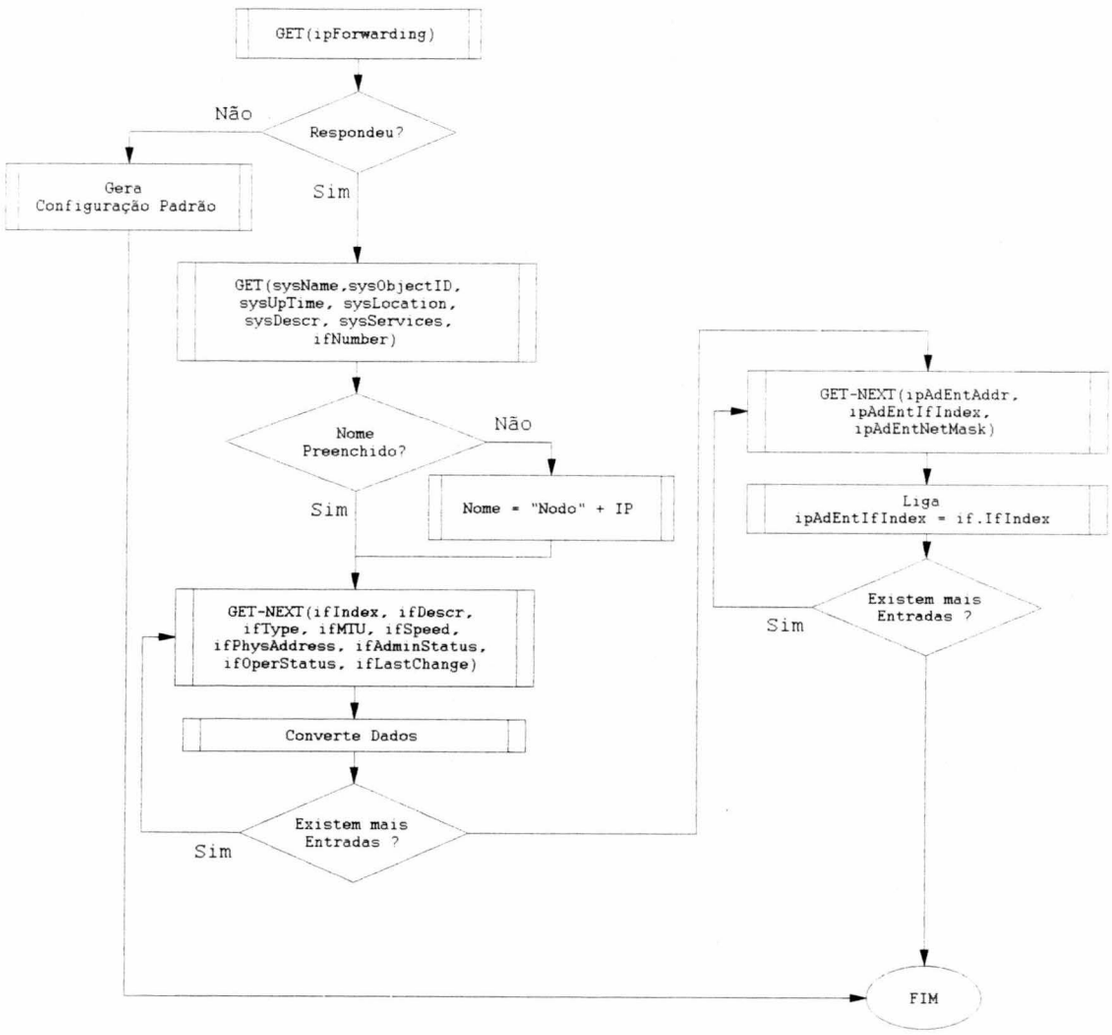

Figura 5.18 : Processo de Obtenção de Configuração do Dispositivo

Alguns problemas podem ocorrer neste procedimento de busca de informações. Um deles é a consulta a um nodo que não possui agente SNMP. Nodos sem agentes não dão nenhum tipo de informação ao gerente, se tornando uma caixa preta para o sistema. Nesses casos, o gerente irá preencher os dados do dispositivo somente com as seguintes informações :

- SNMP : Sem Agente
- Nome : Cria-se o nome "Nodo Nº IP"
- Tipo : Atribui-se a condição de Host. Esta informação pode não ser correta, mais é atribuída admitido-se que é mais provável que um host não possua um agente SNMP do que um gateway.
- Num. Interfaces : Uma
- Endereço IP interface : Número IP utilizado na busca do dispositivo
- Rede : Código da rede na qual se está percorrendo.

Outro problema grave que deve ser resolvido é o caso do dispositivo não possuir nome. O nome é um atributo bastante útil na identificação de um nodo em relação a um conjunto. Por isso, para cada máquina que não tiver seu nome cadastrado na MIB, será atribuído um nome especial criado na seguinte maneira : nome = "Nodo" + End IP. Como o endereço IP é único para toda a rede, é garantido que esse nome não se repetirá em nenhum outro nodo da rede.

5.2.3 Atualização dos Dados

Uma vez já estando de posse das informações sobre o dispositivo, só falta agora atualizar o banco de dados com essas informações. Essa atualização poderá tanto gerar a inclusão de novos registros no banco de dados, como a alteração de registros já existentes. O procedimento normal provavelmente será a modificação dos dados, uma vez que a inclusão e exclusão de nodos na rede são eventos que não acontecem a todo instante com todos os nodos.

Devido aos aspectos de paralelismo que o algoritmo assume, deve-se pensar seriamente nos problemas de concorrência ao estudar as formas de atualizações no banco de dados. O acesso e a modificaçãos dos dados cadastrais até que não são fatores preocupantes, uma vez que mesmo que se dois processos desejarem atualizar os dados sobre o mesmo hosts, ambos obterão seus dados da mesma origem, o próprio host. O problema surge a nível de inclusão de dados : como garantir que dois processos distintos não irão incluir o mesmo dispositivo simultaneamente, ou que não serão incluídos hosts ou redes com o mesmo código ?

Por exemplo, imagine um cenário composto por 15 processos sendo executados, cada um buscando nodos numa rede específica. Num determinado momento, dois nodos chegam a uma mesma máquina que acaba se ser conectada em diversas sub-redes, só que com endereços IP diferentes. Tome como exemplo um processo chegou ao dispositivo através de endereços IP 143.54.11.6 da sua rede, e o outro através do número 143.54.10.2. Ao procurar simultaneamente esses dois endereços no banco de dados para verificar a sua existência, ambos verificarão que o mesmo não existe, gerando a inclusão duplicada do mesmo registro. Esse problema pode ocorrer tanto a nível de host/interface, como a nível de rede(Figura 5.19).

Para solucionar este problema a nível de rede, a inclusão de todas as redes conhecidas são feitas antes do disparo dos processos de detecção de hosts. Desta forma, quando os processos estiverem competindo, não haverá mais inclusão de redes, impedindo desta maneira problemas com a concorrência.

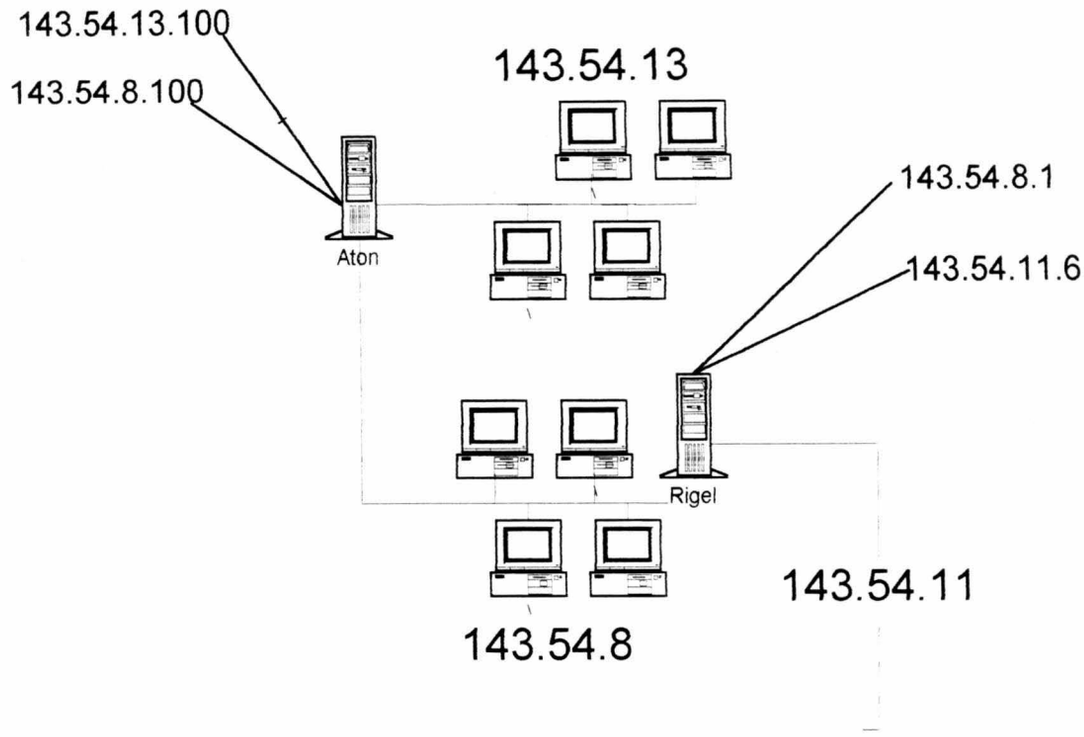

Figura 5.19 : Exemplo de Gateway que faz parte de mais de uma rede

Já a nível de host, este problema é resolvido através da inclusão de interfaces ordenadas pelo seu número IP, ou seja, as interfaces de um dispositivo são incluídas em ordem crescente do seu número IP. Desta maneira, inclusão de um mesmo dispositivo por dois processos concorrentes deverá ser feita na mesma ordem, fazendo que um dos dispositivos não consiga ser incluído devido a existência de uma interface de mesmo número. Isso gerará o aborto do processo que tentou incluir por último o dispositivo.

Voltando ao algoritmo, para cada dispositivo descoberto o gerente irá procurar se o mesmo já se encontra cadastrado no banco de dados. Esta procura é feita através de todos os endereços IP do host. Essa busca garante que dispositivos que receberem novas interfaces ou que modificarem o número IP de uma das suas interfaces não serão tratados como novos dispositivos, uma vez que alguma das suas outras interfaces será encontrada. O ideal seria a verificação dos dispositivos através de seus nomes, porém o fato de alguns dispositivos não possuírem o seu nome cadastrado na MIB inviabiliza esse tipo de processo, uma vez que se o mesmo dispositivo for achado através de dois endereços IP distintos, dois nomes diferentes serão atribuídos ao mesmo dispositivo, criando assim duas instâncias no banco de dados.

Continuando o algoritmo, se for verificado que o dispositivo já se encontra cadastrado, deve-se disparar uma função que irá comparar os dados armazenados no banco de dados com os dados obtidos, verificando se houve alguma modificação tanto a nível de nodo como a nível de interface. Caso hajam alterações, elas são atualizadas no banco de dados e as modificações realizadas são cadastradas do modo que se possa relata-las ao final do algoritmo.

Caso o dispositivo não exista no banco de dados, ele é incluído com todas as suas informações. Ao final da atualização de cada dispositivo, é obtido a data e a hora atual do sistema e é gerada uma entrada de contabilização indicando que o host estava ativo naquele momento.

Finalmente, quando o sistema termina de percorrer todos os nodos possíveis da rede, a contagem feita durante o processamento com o número de hosts e de gateways cadastrados é atualizada no banco de dados(Figura 5.20).

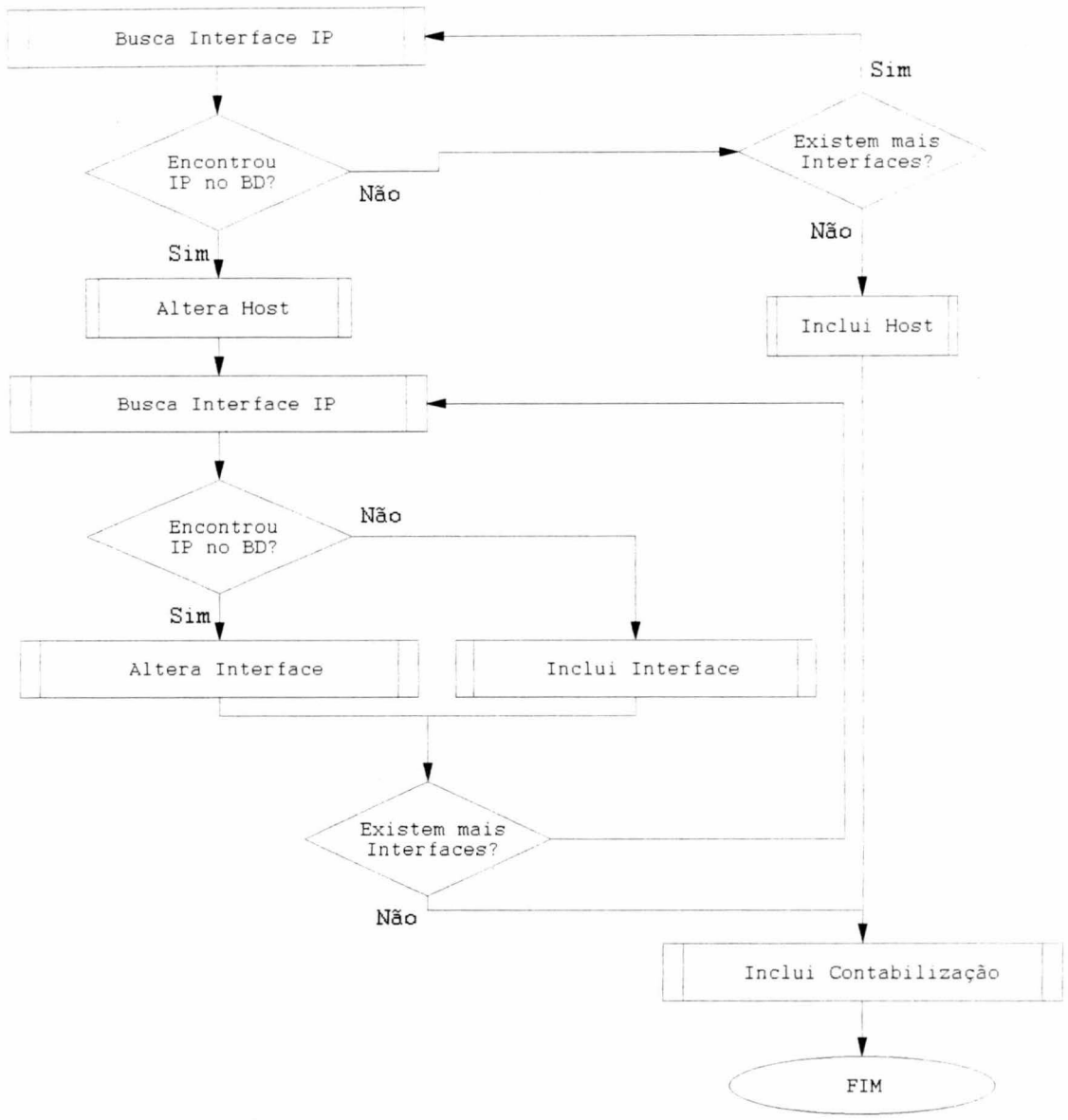

Figura 5.20 : Algoritmo para Atualização dos dados

5.2.4 Geração do relatório com as modificações da Rede

Todas as modificações detectadas pelo WatchDog em relação a configuração antiga que ele possui armazenada são registradas individualmente por todos os processos que percorrem as redes. Devido ao fato de não serem utilizados nenhum mecanismos de intercomunicação entre processos, essas modificações são armazenadas num formato próprio gravado em um arquivo de nome 'R'+Endereço_Rede+'.rel' .

Ao receber o aviso de término de todos os subprocessos de rede, o processo coordenador coleta as informações espalhadas nos diversos arquivos e junta-as num arquivo só, que recebe o nome de 'Relatorio.txt'.

Além disso, é registrado também neste arquivo todos os hosts registrados no banco de dados e que não são detectados a um período expressivo de tempo. Deste

modo o administrador pode verificar se o dispositivo não têm sido detectado devido a problemas de comunicação, por ter sido desconectado da rede ou por se encontrar desligado nas vezes que o WatchDog passou. O tamanho deste período é designado na configuração do WatchDog e é expresso em dias.

A exclusão definitiva do nodo deve ser feita através do módulo de exclusão via WWW, onde o administrador apaga os registros um a um. Esta exclusão não é feita automaticamente pelo algoritmo devido às duvidas em relação ligação ou não do dispositivo à rede.

5.2.5 Visão Geral do sistema

Seguindo o algoritmo descrito nas seções acima, o sistema de WatchDog pode ser dividido em processos com tarefas bem definidas que se comunicam e cooperam entre si a fim de contemplar a funcionalidade do sistema. Essa modelagem foi feita tomando como base os diagramas em SDL. A figura 5.21 mostra os agentes projetados e a troca de serviços realizada entre eles.

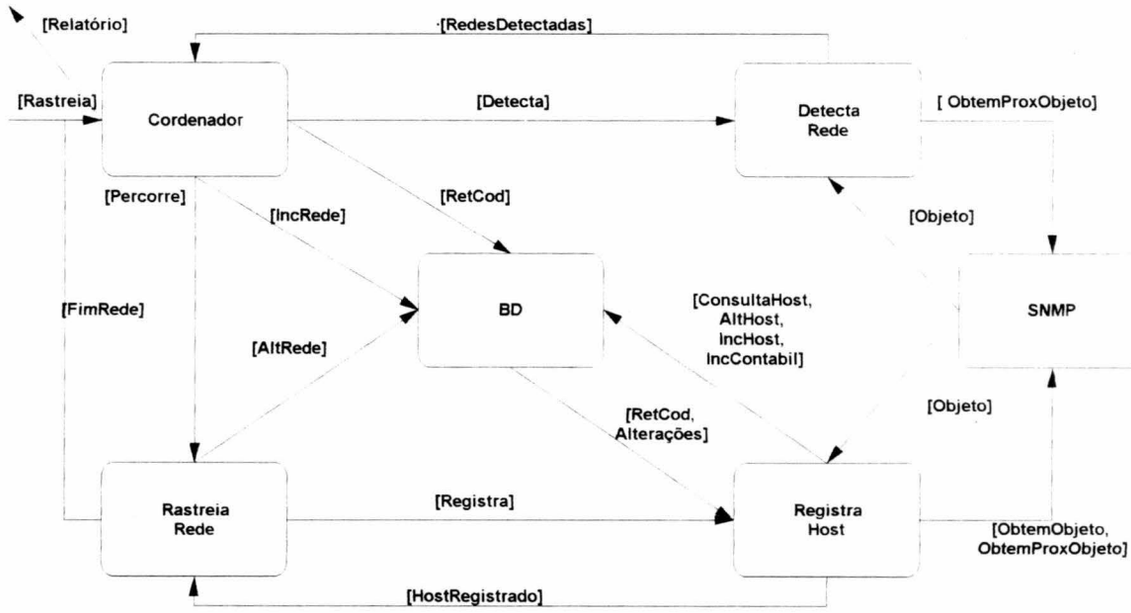

Figura 5.21 : Visão Geral do Sistema e a cooperação entre os seus agentes

O sistema é composto de seis processos que implementam as seguintes funções :

- Coordenador : Processo responsável por controlar toda a atividade de detecção e registro de banco de dados, ele é quem gerencia todo o processamento do WatchDog.
- DetectaRede : Processo que varre a rede atrás de endereços de sub-rede que estejam dentro do domínio especificado na configuração do sistema.
- RastreaRede : Processo que detecta todos os nodos presentes na rede de sua responsabilidade. Cada sub-rede encontrada pelo processo anterior gera um evento para este processo.
- RegistraHost : Processo responsável por obter todas as informações de configuração do host e garantir que as mesmas sejam armazenadas no banco de dados.
- SNMP : Processo responsável por realizar as consultas SNMP.

• BD : Processo responsável por se comunicar com o banco de dados e gerenciar os seus dados.

Vamos a seguir analisar cada um destes módulos.

Primeiramente o coordenador tem como função básica gerenciar o algoritmo como um todo. Para isso, ele parte de alguns nodos iniciais e detecta as sub-redes presentes no seu domínio de busca. Após, ele coordena a execução do rastreamento em cada sub-rede, não permitindo nunca que o número de processos rastreadores em execução passe do limite máximo atribuído pelo administrador. Finalmente, após a execução em todas as sub-redes, o sistema gera um relatório com todas as modificações detectadas.

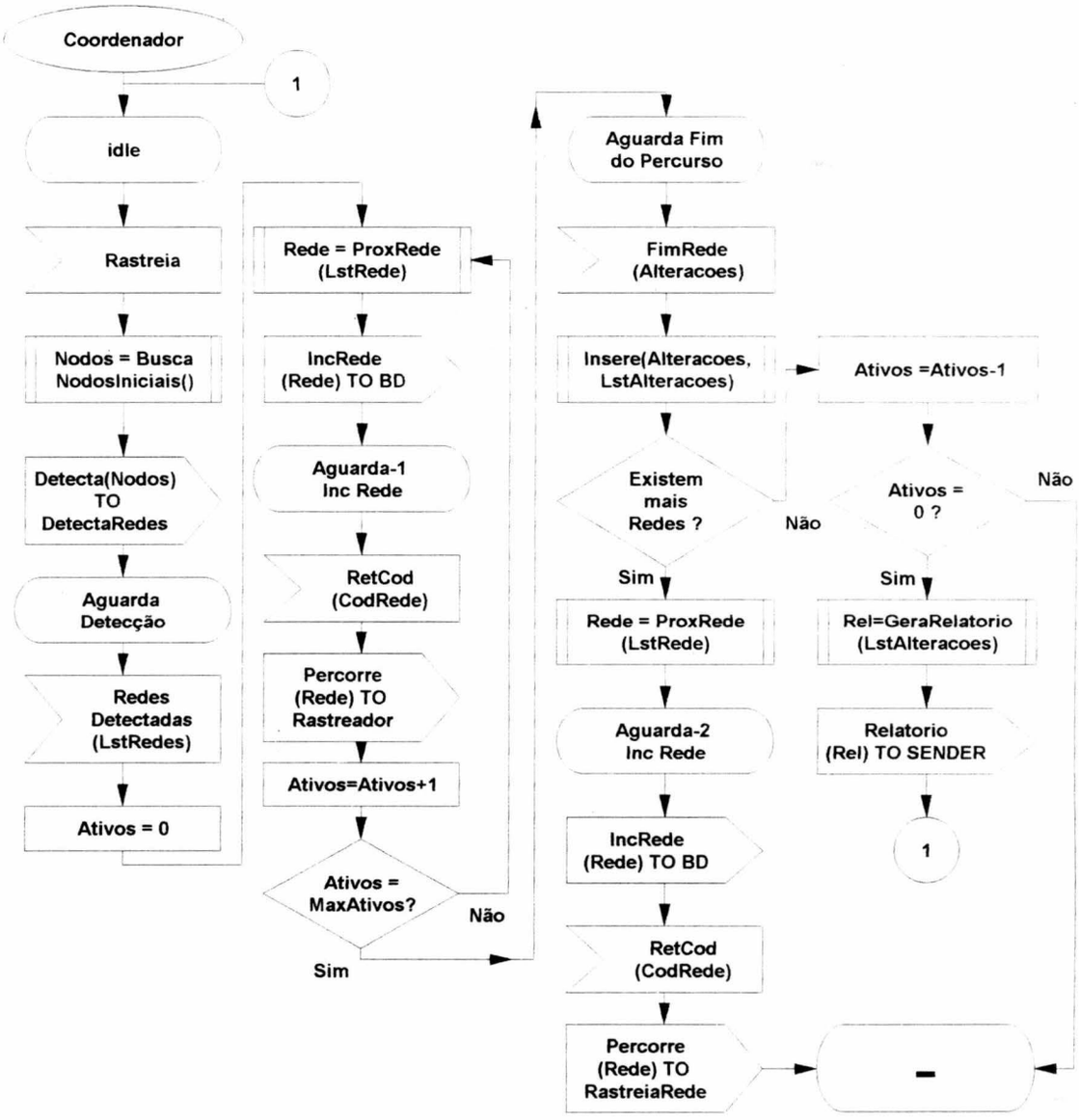

Figura 5.22 : Processo de Coordenação do sistema

O processo DetectaRedes encontra as redes presentes através de consultas às tabelas de roteamente e de endereçamento de cada máquina que ele encontra. Para isso ele se utiliza das funções de consulta do SNMP.

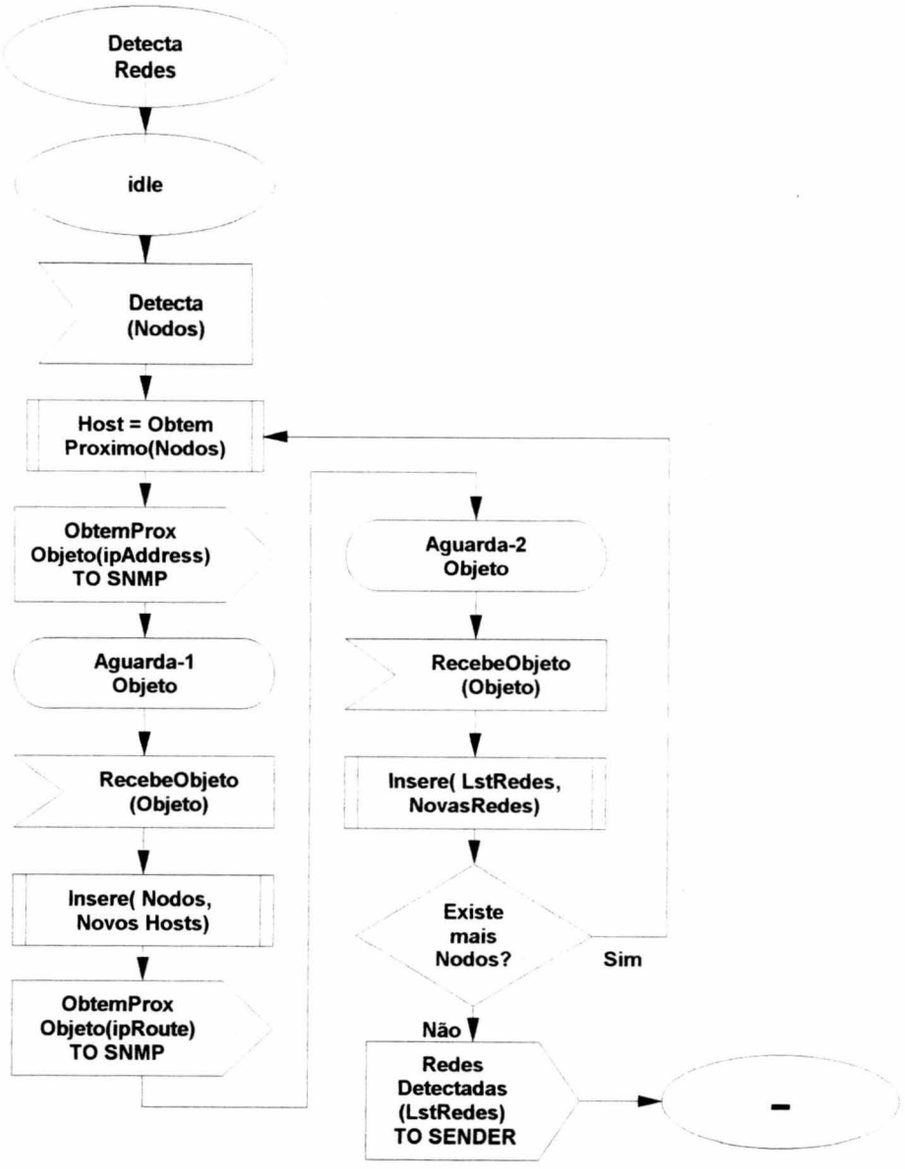

Figura 5.23 : Processo de Detecção de Redes

O processo RastreaRede procura através de pings todos os hosts presentes na sub-rede, disparando um evento de RegistraHost para cada nodo encontrado. Após a busca por toda a rede, ele salva as informações coletadas no banco de dados.

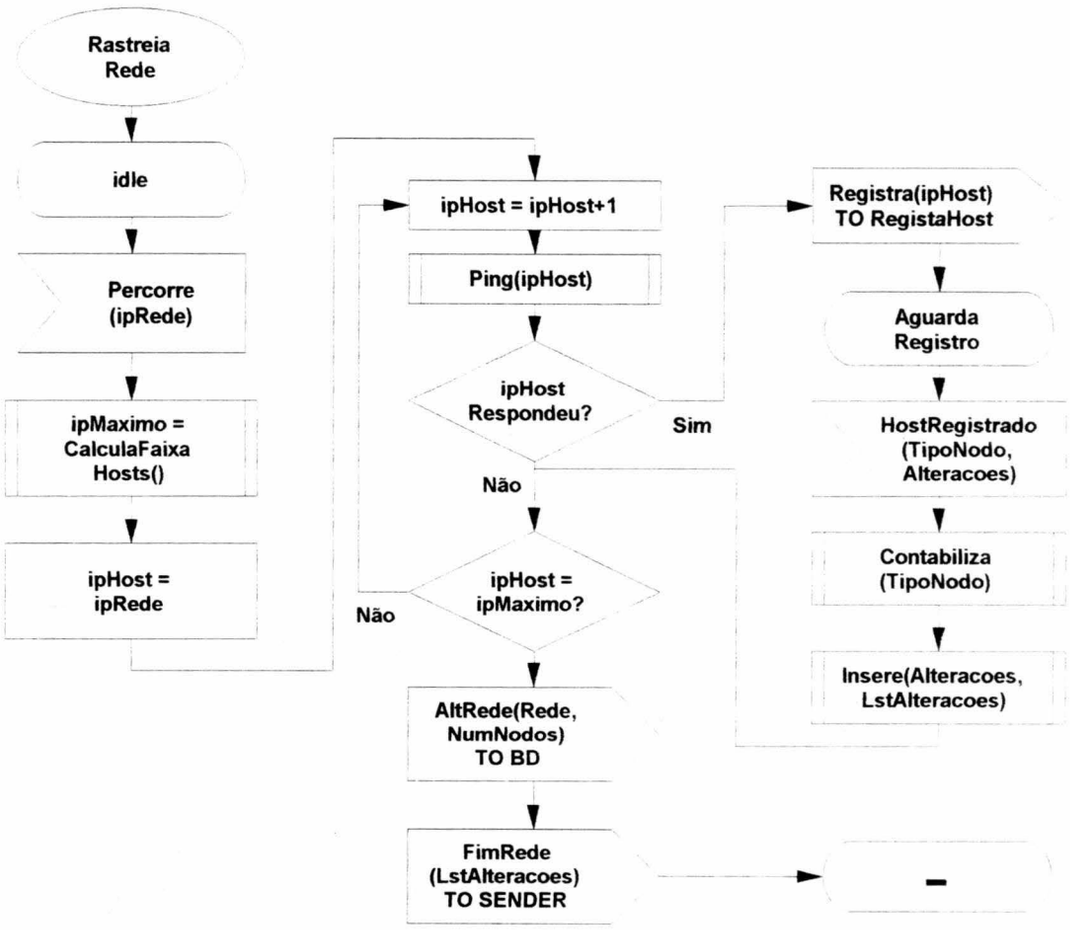

Figura 5.24 : Processo de rastreamento de nodos por rede

O processo RegistraHost obtém todas as informações sobre um nodo pertinentes ao sistema através de consultas SNMP. Após, ele verifica a existência ou não do host no banco de dados, incluído-o caso o nodo não esteja cadatrado aidna ou verificando as informações obtidas com as armazenadas a fim de verificar se houve alguma modificação. Após, ele registra a detecção do nodo no banco de contabilização.

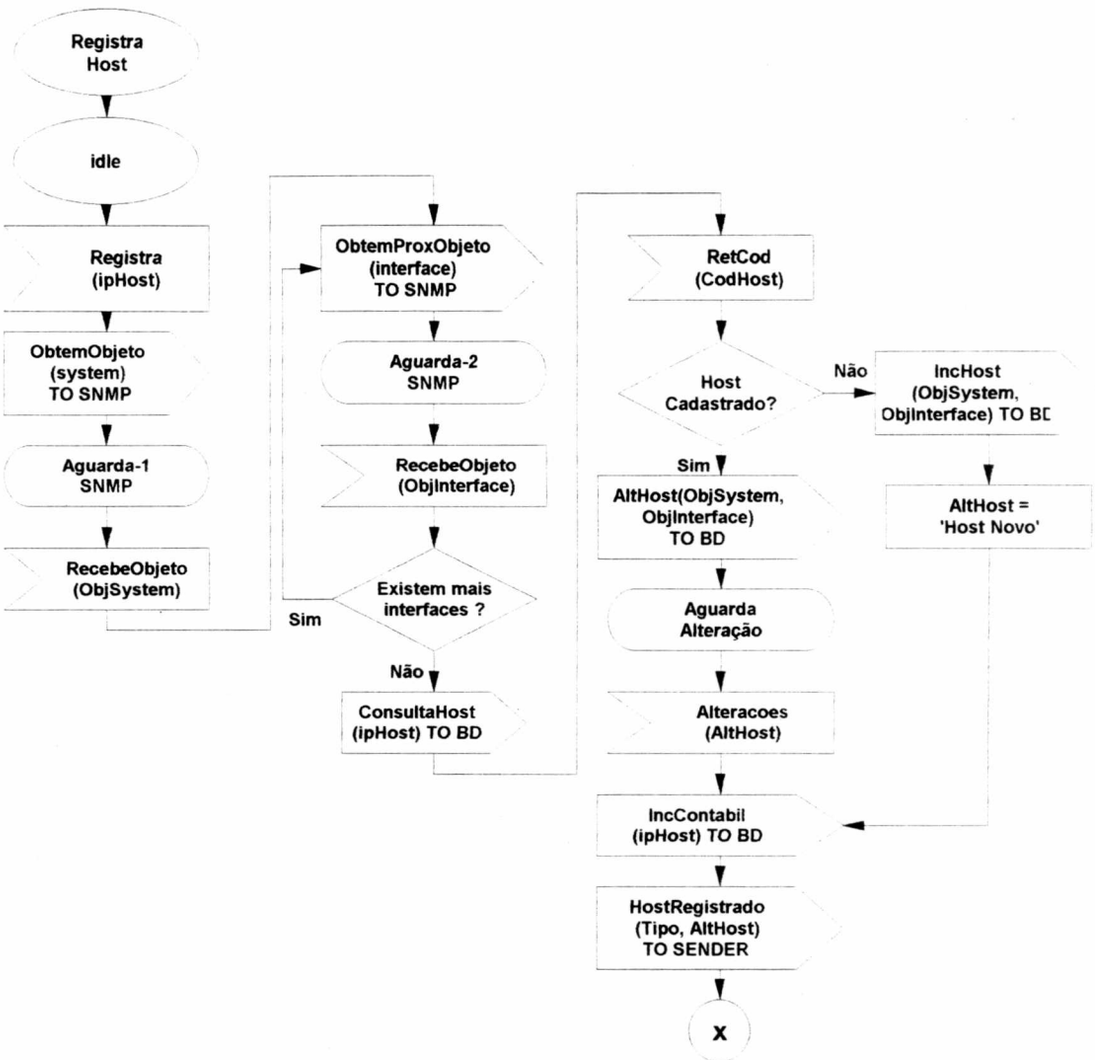

Figura 5.25 : Processo que Obtém os dados do host e o registra no BD

Finalmente, o processo de BD simplesmente submete comandos de consulta e atualização ao banco de dados, conforme for requisitado.

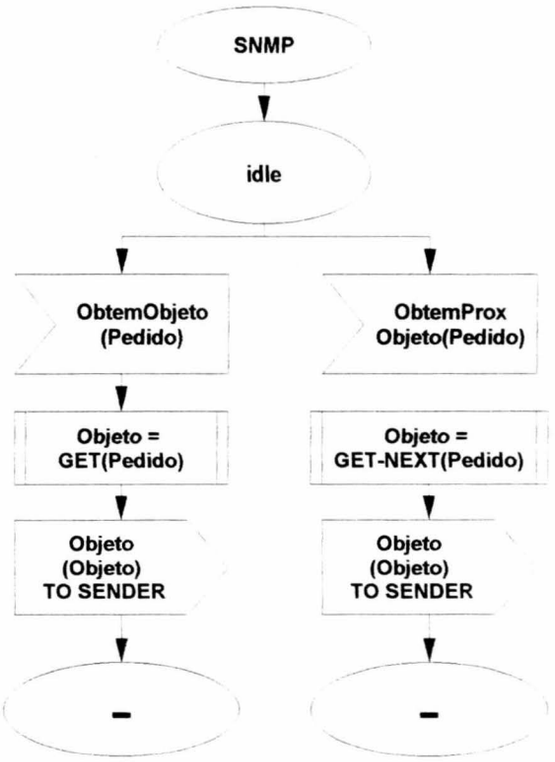

Figura 5.26 : Processo responsável por realizar as consultas SNMP

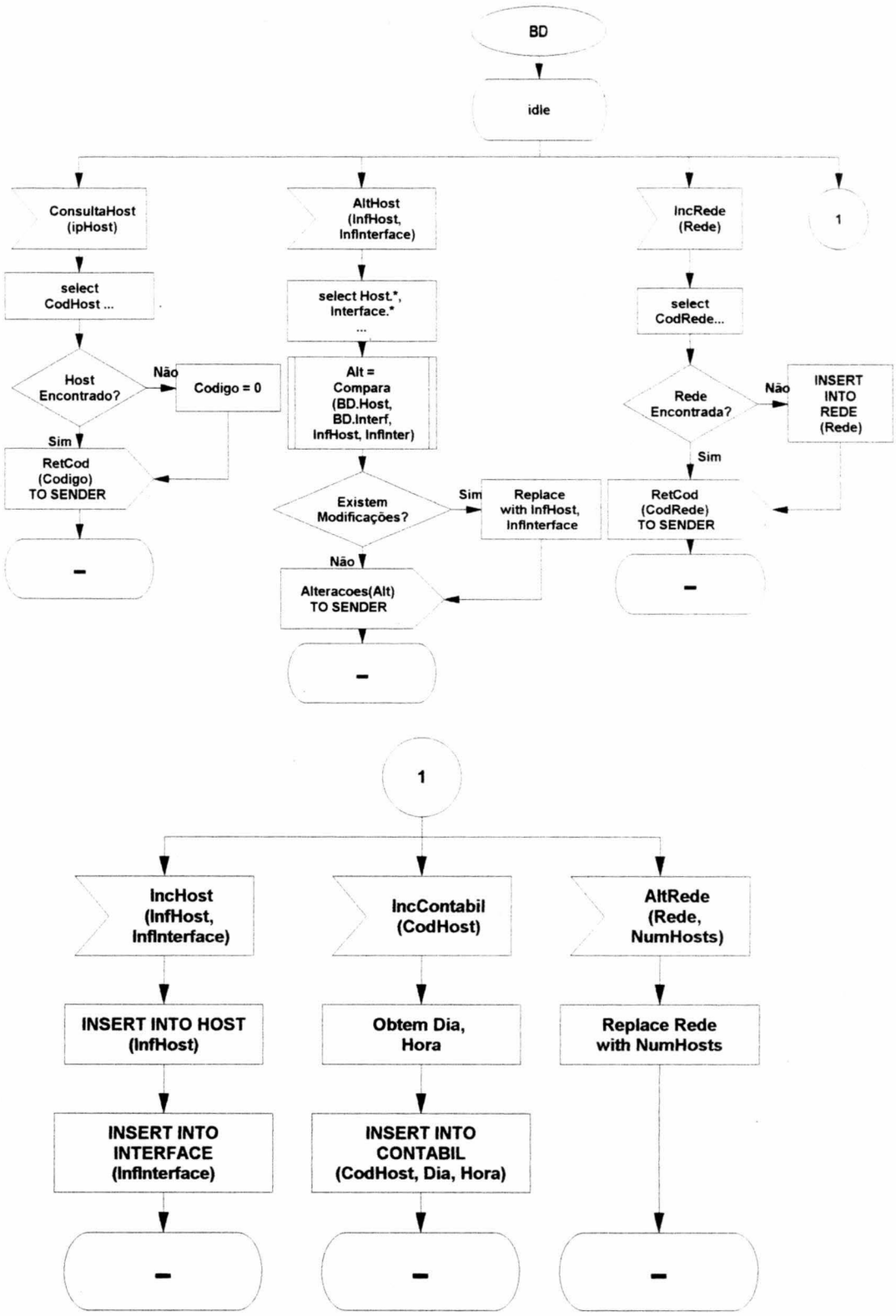

Figura 5.27 : Agente de Banco de Dados

5.2.6 Implementação

A implementação do WatchDog foi feita utilizando uma série de ferramentas que permitiram a implantação de toda a sua funcionalidade. Entre essas ferramentas,

devem ser citadas a interface para linguagem C do Postgres, bem como a biblioteca de funções SNMP da CMU.

Todas as funções utilizadas pelo WatchDog foram separadas por funcionalidade em arquivos distintos, visando criar um código claro e bem estruturado. A seguir será analisado cada um dos módulos criados, e sua iteração com outras ferramentas quando for o caso.

5.2.5.1 Algoritmo WatchDog

O algoritmo descrito até aqui está implementado no arquivo WatchDog.c . Esse algoritmo é usuário de todas as demais funções que serão descritas nas próximas seções. A separação das diversas tarefas que o algoritmo deve cumprir foram separados nos seguintes módulos :

BuscaNodoInicial() - Consulta o arquivo de hosts localizado no diretório /etc, a obter sugestões de nodos a serem consultados inicialmente pelo algoritmo de busca de redes.

DetectaRedes() - Implementa o algoritmo de detecção de redes explicado acima.

ProcessoRede(End_Rede, Cod_Rede) - Processo que é disparado em paralelo e é responsável por detectar os nodos na rede e cadastrá-los.

RegistraHost(End_IP, Cod_Rede) - Processo que busca informações sobre o host encontrado na rede pelo ping e o registra no banco de dados.

5.2.5.2 Funções de Processamento e Conversão

Contido no arquivo converte.c, essas funções são as responsáveis por transformas os valores trazidos da MIB em valores claros para o usuário e processar os valores obtidos para obter os valores dos atributos que são calculados pelo sistema. Estão definidas as seguintes funções :

ObtemClasseRede (End_Rede) - Retorna a classe da rede.

VerificaNome(Nome, End_IP) - Verifica se o nodo possui um nome. Caso não tenha, monta um nome padrão do endereço IP que foi utilizado para encontra-lo.

ConverteServiço(iServiço) - Retorna com uma string contendo todos os serviços disponibilizados por um dispositivo, conforme procedimento de conversão explicado acima.

5.2.5.3 Funções de Acesso ao SNMP

A implementação das funções de consulta SNMP foram bastante dificultadas devido a falta de material explicativo sobre a biblioteca disponibilizada pelo CMU. A única documentação que eles fornecem é um arquivo que possui uma explicação sucintíssima a respeito de um subconjunto de funções disponibilizados por eles. Quando se olha realmente o código dos programas deles, o que se vê é um grande número de funções que nunca foram citadas e cuja única pista que se têm sobre o sua funcionalidade é através da análise do seu nome. Realmente o que torna o pacote utilizável são as aplicações prontas que disponibilizam os funções básicas do SNMP.

Além disso, o trabalho foi muito dificultado devido à problemas encontrados no pacote de SNMP 1.0 . Devido a uma má administração de memória, muitas vezes o espaço alocado pelo pacote para transmitir e receber as PDUs não eram desalocadas. Em programas que utilizam poucas chamadas ao SNMP este problema não era detectado, uma vez que a memória que deixava de ser desalocada era pouca. Já num

sistema que requer uma grande quantidade de consultas como o WatchDog, a memória que não era desalocada ia se acumulando lentamente entre os pedidos, causando uma interrupção no sistema no meio da sua execução. A detecção deste problema no pacote requeriu uma quantidade significativa de tempo para que se chegasse a conclusão que o problema era realmente no pacote e não no sistema.

A resolução dele começou com uma procura pelo WWW, ftp e News atrás de documentos que expusessem este problema e indicasse como resolvê-lo, de novas versões do SNMP ou de pacotes de outros fabricantes. Após pesquisar exaustivamente e testar alguns outros pacotes tais como um da MIT, se chegou á versão 2 do SNMP da CMU, que além de dar suporte ao SNMP 1.0, manteve as mesmas estruturas da versão anterior, tornando assim muito mais fácil a migração de uma versão para outra.

As funções responsáveis por realizar todas as consultas a objetos utilizaram somente dois comandos SNMP : get e get-next. Esses comandos foram implementados com base nos fontes disponíveis no diretório snmp/apps : snmpget.c e snmpgetnext.c .

Esses arquivos geraram duas funções que se encontram no arquivo snmp-gerente.c :

ObtemObjeto(End_IP, Id_Objeto) - Obtém um objeto da MIB.

ObtemProximoObjeto(End_IP, Id_Objeto) - Obtém o objeto seguinte ao pedido.

Os objetos da MIB podem ser de diversos tipos de dados. Foram necessários no escopo deste trabalho apenas dois tipos básicos : string e OID. O OID é composto por uma matriz de longs, permitindo dessa maneira que objetos da MIB do tipo long sejam retornados num tipo OID. Para obter esses dados, as funções descritas acima sempre retornam uma estrutura de dados alocada dinamicamente e que possui essas informações. Essa estrutura deve, sempre depois que for lida, ser desalocada.

A fim de otimizar o trabalho com consulta de objetos, outras funções foram definidas para se obter os objetos diretamente na variável do atributo de uma entidade sem precisar se preocupar com alocação de memória. São elas :

ObtemDadoString(End_IP, Objeto) - Retorna uma string com o objeto da MIB pedido.

ObtemDadoLong(End_IP, Objeto) - Retorna um long com o objeto da MIB pedido.

5.2.5.4 Funções para gerência de listas

A fim de gerenciar as listas dinâmicas necessárias pelo WatchDog, tais como a de hosts, gateways e redes, foram criados um conjunto de funções que realizam essa tarefa sobre uma estrutura de dados pré-definida em gerlista.c .

São elas :

AlocaLista(Lst) : Inicializa uma lista para armazenar os informações desejadas.

InsereInformação(Lst, Informação) - Insere uma informação caso ela não se encontre na lista ainda.

PesquisaInformação(Lst, Informação) - Retorna um flag Boolean indicando se uma informação já se encontra cadastrada.

ObtemProximaInformação(Lst) - Retorna o próximo nodo da lista que não foi consultado ainda.

5.2.5.5 Banco de Dados

Durante o percorrer do trabalho, o Postgres se mostrou ser um SGBD não muito confiável no tratamento de concorrência entre processos. Os erros apresentados por ele foram dos mais variados tipo, desde a quebra da conexão com o banco de dados por motivos desconhecidos até problemas de timeout. Muitas vezes o Postgres falhava e passava a não aceitar mais requisições de ninguém, sendo necessário então matar à mão todos os processos criados por ele, desativar o postmaster e voltar a ativá-lo novamente. O problema se tornou tão crítico que num determinado momento foi necessário desistalá-lo da máquina Penta e instalá-lo novamente.

O crescimento dos erros se mostrou diretamente proporcional ao número de processos ativos se conectados ao postgres no momento da falha. A incidência seguida destes erros exigiu que fossem reforçadas as rotinas de comunicação com o banco de dados através da realização de testes a cada operação executada. Caso a operação não seja executada com sucesso, o sistema desfaz a conexão, cria uma nova e tenta executar novamente a operação. O número de vezes que estes passos são realizados é definido por uma constante MAX_EXECUCAO_BD. Caso o sistema não consiga realmente executar a operação, ele desiste dela.

As funções de banco de dados são as responsáveis por realizar a comunicação entre o WatchDog e o Postgres. Para isso ela utiliza a biblioteca de funções de conexão C e Postgres que vem com o Banco de dados. Essas funções se localizam no arquivo bd.c . Devido ao fato de todas as operações de banco de dados se localizarem num só local, a biblioteca de conexão também é somente declarada neste arquivo.

As funções desenvolvidas se dividem em funções genéricas e em funções de atualização e consulta específicas para uma determinada entidade. As primeiras são utilizadas para realizar operações genéricas sobre o banco de dados. Dentre elas têm-se as seguintes funções :

AbreConexão() - Abre uma conexão entre o programa cliente e o banco de dados. Depois de executada esta operação, podem ser realizadas operações sobre o banco de dados.

FechaConexão() - Fecha uma conexão aberta pelo função acima.

PesquisaBD(szConsulta) - Realiza uma consulta sobre o banco de dados e retorna com o valor do primeiro atributo da primeira instância do resultado.

AtualizaBD(szOperacao) - Realiza uma operação de atualização sobre o banco de dados (append, replace ou delete), retornando um valor booleano que indica se a operação foi executada com sucesso.

Já as funções específicas levam em conta os atributos e condições de uma determinada entidade, utilizando muitas vezes as funções genéricas para realizar as suas operações. Dentre elas tem-se as funções :

BuscaRede(szIPRede) - Busca o código de uma determinada rede no Banco de dados. Um código inválido indica que a rede não se encontra cadastrada.

IncluiRede(REDE) - Inclui uma rede no banco de dados.

BuscaHost(szIPHost) - Busca o código de um determinado host no banco de dados. Um código inválido indique o host não está cadastrado.

IncluiHostSemSNMP(szIPRede, szNome, szIPHost) - Inclui um dispositivo na qual não foi possível obter informações por não possuir agente SNMP. A sua inclusão é feita com o mínimo de informações.

IncluiHost(HOST, INTERFACE) - Inclui um dispositivo no banco de dados.

AlteraHost(szIPHost, HOST) - Altera as informações de um dispositivo Cadastrado.

BuscaInterface(szIPInterface) - Retorna com o número da interface no dispositivo na qual ela faz parte.

InsereInterface(szCodHost, INTERFACE) - Inclui uma interface no banco de dados e liga essa interface ao dispositivo correspondente a szCodHost.

AtualizaInterface(szCodHost, szNumInterface, INTERFACE) - Atualiza os dados de uma determinada interface presente no banco de dados.

IncluiContabilizacao(szCodHost) - Registra o dispositivo referenciado por szCodHost como ativo no momento da inclusão do registro de contabilização no banco de dados.

6. Conclusão

O presente trabalho apresentou uma visão geral do ambiente de redes atual e a sua expansão cada vez mais crescente. Juntamente com essa expansão, crescem as dificuldades de se gerenciar um ambiente que se encontra espalhado nas mais diversas localidades físicas. A fim de estudar melhor estes problemas, foram apresentados alguns conceitos de gerenciamento de rede, particularmente no que diz respeito a problemas de configuração.

Dentro deste contexto, foi proposta a implementação de um gerente de configuração chamado de WatchDog que visa resolver dois dos principais problemas : o controle dos dispositivos presentes na rede e da sua configuração. Para resolvê-los, esse gerente realiza a detecção automática dos dispositivos presentes na rede e armazenas as suas configurações.

A fim de implementar este gerente, o trabalho fez um estudo sobre quais objetos que deveriam ser armazenados pelo WatchDog e como se daria o armazenamento dos mesmos. Foi realizado um estudo sobre quais as melhores maneiras do usuário realizar consultas ao gerente sobre as configurações e a partir disto foi criada uma interface com o usuário.

O sistema desenvolvido disponibiliza uma ferramenta útil para se detectar os dispositivos que compõem a rede, bem como uma excelente fonte de consulta sobre o estado e configuração dos dispositivos, permitindo que se acesse tanto a informações de um determinado dispositivo, como se consulte dispositivos que atendem a determinados requisitos solicitados pelo usuário.

Através da sua interface WWW, o sistema criado permitiu que o seu acesso não tenha limites, podendo ser realizado de qualquer máquina e lugar que esteja conectada a internet. Isso permite uma popularização das informações, fornecendo uma ferramenta importante de gerência não somente para um determinado departamento ou CPD, mas para todos os institutos que compõem a UFRGS.

Uma vez implantado o WatchDog, existem uma série de funcionalidades que podem ser incrementados ao seu modelo. Sua utilização, por exemplo,funciona com um banco de dados somente, forçando que todos os usuários do sistema vejam e atualizem o mesmo repositório. A implantação de um mecanismo baseado em usuário, que permitisse a cada usuário do sistema ter a sua base de dados e o seu registro de configuração tornaria o sistema mais personalizado. Além disso, poderia ser implantada a opção de cada usuário manter diversas configurações de rede. Por exemplo, poderia-se ter uma configuração onde ficasse armazenada somente as redes de um determinado instituto, outra composta somente pelas redes que compõem a UFRGS e outra que varresse toda a RNP. É claro que o armazenamento de várias configurações acarretaria numa atualização periódica de todas as configurações, além de duplicar as informações entre as configurações distintas. Se cada usuário, além disso, tiver configurações diferentes, o custo de processamento e de armazenamento para manter esse tipo de estrutura começaria a se tornar caro.

O sistema pode também ter um mecanismo de aviso via mail das alterações encontradas entre as diversas ativações dos processos de detecção, de modo que o usuário não necessite ficar esperando o processo terminar para ver as modificações encontradas.

Outro recurso interessante seria o armazenamento dos alterações encontradas num banco de dados, de modo a permitir que o administrador acompanhe as mudanças ocorridas e tenha um histórico sobre elas.

7. Bibliografia Consultada

[BRI 93] BRISA. Gerenciamento de Redes : Uma aboradagem de sistemas abertos. São Paulo: Makron Books, 1993, 364p.

[BRI 96] BRISA. Arquitetura de Redes de Computadores OSI e TCP/IP. Makron Books, São Paulo, 1996.

[BYT 96a] BYTE BRASIL. INTERNET - O mundo a seu alcance. Fevereiro de 1996. Editora REVER. São Paulo (SP), p. 41.

[CAR 96] Cardoso, Carlos. HTML : Truques Espertos. Axcel Books do Brasil, Rio de Janeiro, 1996.

[CGI 93] COMMOM GATEWAY INTERFACE. The Commom Gateway Interface. 1995. (Disponível via WWW em http://hohoo.ncsa.edu/cgi/intro.html E-mail : cgi@ncsa.uiuc.edu)

[CMU 96] CMU SNMPv2 Archives (Disponível via ftp em lancaster.andrew.cmu.edu no diretório /pub/snmp)

[COM 91a] COMER, Douglas E. . Internetworking with TCP/IP Vol 1: Principles, Protocols, and Architeture. Englewook Cliffs, New Jersey: Prentice Hall, 1991, 547p.

[COM 91b] COMER, Douglas E. . Internetworking with TCP/IP Vol 2 : Principles, Protocols, and Architeture. Englewook Cliffs, New Jersey: Prentice Hall, 1991, 547p.

[FAN 93] FANG, Karen; Leinwand, Allan. Network Management : A pratical perspective. Addison-Wesley, 1993, 222p.

[FOX 95] FOX, David; Downing, Troy. Dominando o editor HTML Web. Rio de Janeiro : Ciência Moderna, 1995, 574p.

[INT 96] Internet World. Volume 1 No 8.Mantel Media Editora. Rio de Janeiro (RJ).

[KUH 95] KUHN, Rafael Vilarinho. A integração de Servidores WWW com Banco de Dados Visando a Atualização Automática de Hiperdocumentos. Porto Alegre. Universidade Federal do Rio Grande do Sul - UFRGS, 1996.

[LAN 96] LAN TIMES BRASIL. A solução não pode dar problema(46) e Monitoração Remota(50). Editora REVER. São Paulo (SP), Março de 1996.

[ODA 94] ODA, Cybelle Suemi. Desenvolvimento de um sistema monitor gráfico baseado em protocolo de gerenciamento SNMP. São Carlos, Instituto de Ciências Matemáticas de São Calor, 1994, 110p.

[MAD 94] MADRUGA, Ewerton Longoni. Ferramentas de Apoio à Gerência de Falhas e Desempenho em contexto distribuído. Porto Alegre: CPGCC da UFRGS, 1994, 135p.

[MIT 96] MIT´s SNMPv1 Package. 1995. (Disponível via ftp em mercury.lis.mit.edu no diretório /pub/snmp)

[NET 95] Network Management Archives. Julho de 1995. (Disponível via WWW em http://smurfland.cit.buffalo.edu/NetMan/Archives.html#MibBrowse

[POS 94] POSTGRES MANUAL. The POSTGRES User Manual. Universidade da California, 1994.

[POS 94a] POSTGRES release 4.2. 1994. (Disponível via WWW em http://s2k-ftp.CS.Berkeley.EDU:8000/postgres/postgres.html)

[ROS 91] ROSE, Marshall. The Simple Book: an introduction to management of TCP/IP-based internets. Englewook Cliffs: Prentice-Hall, 1991. 347p.

[SCH 91] SCHILDT, H. C Completo e Total. São Paulo, Makron Books do Brasil, Editora Ltda, 1991.

[SCH 93] SCHMITT, Marcelo Augusto Rauh. Um modelo de gerência de configuração de redes locais. Porto Alegre: CPGCC da UFRGS, 1993, 132p.

[SCH 95] SCHOSSLER, Alexandre & HEUSER, Carlos Alberto. Guia de desenvolvimento de aplicação hipermídia para o ambiente da rede Internet. Porto Alegre. Universidade Federal do Rio Grande do Sul - UFRGS, 1995.

[SUN 90a] SUN MANUAL. Networking Programing Guide. Sun Microsystem, 1990.

[SUN 90b] SUN MANUAL. SunOS Reference Manual-Maintenance commands. Sun Microsystem, 1990.

[TAN 94] TANENBAUM, Andrew S.. Redes de Computadores. Rio de Janeiro : Campus, 1994, 786p.

[TUT 96] Tutorial - Como Criar Documentos Interativos no WWW. (Disponível via WWW em http://penta.ufrgs.br/edu/forms/tut0.html)

[VEN 96] Venetianer, Tomas. HTML : Desmistificando a linguagem da internet. Makron Book, São Paulo, 1996.

[YOU 90] YOURDON, Edward. Análise Estruturada Moderna. Rio de Janeiro : Campus, 1990, 836p.