UNIVERSIDADE REGIONAL DE BLUMENAU

CENTRO DE CIÊNCIAS EXATAS E NATURAIS

CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

PROTÓTIPO DE SOFTWARE PARA GERÊNCIA DE PATRIMÔNIO DOS EQUIPAMENTOS DE UMA REDE UTILIZANDO *SESSION MESSAGE BLOCK*

EDMUNDO NOLAS DE OLIVEIRA JUNIOR

BLUMENAU
2005

2005/1-12

EDMUNDO NOLAS DE OLIVEIRA JUNIOR

PROTÓTIPO DE SOFTWARE PARA GERÊNCIA DE PATRIMÔNIO DOS EQUIPAMENTOS DE UMA REDE UTILIZANDO *SESSION MESSAGE BLOCK*

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Ciências da Computação — Bacharelado.

Prof. Francisco Adell Péricas - Orientador

BLUMENAU
2005

2005/1-12

PROTÓTIPO DE SOFTWARE PARA GERÊNCIA DE PATRIMÔNIO DOS EQUIPAMENTOS DE UMA REDE UTILIZANDO *SESSION MESSAGE BLOCK*

Por

EDMUNDO NOLAS DE OLIVEIRA JUNIOR

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Membro:	Prof. Franciso Adell Péricas, Orientador – FURB
Membro:	Prof. Paulo Fernando da Silva – FURB
Membro:	Prof. Sérgio Stringari – FURB

Blumenau, 01 de junho de 2005

Dedico este trabalho a todos os que direta ou indiretamente ajudaram e colaboraram com estudos e ensinamentos para que o objetivo final deste trabalho fosse atingido.

AGRADECIMENTOS

À Deus, pelo seu imenso amor e graça, que me guia e me ilumina.

Aos meus pais, Edmundo e Olinda pela confiança, carinho e ensinamentos de toda uma vida.

A minha querida esposa Simone pelo apoio, paciência e confiança que depositou em mim em todas as horas dedicadas à este trabalho.

Ao meus filhos, pela alegria que me trazem todos os dias.

A minha mais nova filha Melissa, que veio ao mundo durante a elaboração deste trabalho e me motivou mais ainda e encheu a minha vida com muito mais alegria.

Aos meus amigos e familiares que sempre torcem por mim.

E por fim, ao meu orientador, Francisco Adel Péricas, por ter acreditado na conclusão deste trabalho.

Obrigado à todos.

RESUMO

Este trabalho apresenta um estudo sobre gerência de patrimônio de rede. Dentre os diversos protocolos existentes para executar a gerência de rede, neste projeto foi utilizado o protocolo *Session Message Block* (SMB) que junto ao Registro do Windows colhe as informações necessárias para a gerência dos dispositivos com sistema operacional Windows da Microsoft de uma rede. Também é apresentado como foi implementado um protótipo de *software* que faz a gerência de patrimônio de rede e suas funcionalidades.

Palavras-chave: Inventário. Gerência. Patrimônio. Redes de computadores.

ABSTRACT

This work presents a study on management of net patrimony. Amongst the diverse existing protocols to execute the management of net, in this project was used the protocol Session Message Block (SMB) that with the Register of the Windows it harvests the necessary information for the management of the devices with Windows operational system of Microsoft in a net. It is also presented as was implemented a software archetype that makes the management of net patrimony and its functionalities.

Key-Words: Inventory. Management. Patrimony. Computer networks.

LISTA DE ILUSTRAÇÕES

LISTA DE SÍMBOLOS

ARP – *Address Resolution Protocol*

ASN.1 – *Abstract Syntax Notation One*

CIM – *Common Information Model*

CMIP – *Common Management Information Protocol*

CMIS – *Common Management Information Service*

ISO – *International Standars Organization*

LAN - *Local Area Network*

MIB – *Management Information Base*

OID – *Object Identifier*

OSI – *Open Systens Interconnection*

PID – *Process Identifier*

SMAE – *System Management Application Entity*

SMAP – *System Management Application Proccess*

SMB – *Session Message Block*

SNMP – *Simple Network Management Protocol*

TCP – *Transmission Control Protocol*

TID – *Tree Identifier*

UDP – *User Datagram Protocol*

UML – *Unified Modeling Language*

USB – *Universal Serial Bus*

WMI – *Windows Management Instrumentation*

SUMÁRIO

1 INTRODUÇÃO

Hoje em dia, com o crescimento das redes de computadores nas empresas, e com a facilidade que se tem de adicionar e remover periféricos nas estações de uma rede, fica cada vez mais difícil gerenciar todos os dispositivos que compõe cada estação da rede.

A facilidade de conexão de periféricos em computadores através de portas *Universal Serial Bus* (USB), paralelas e seriais, torna as estações de uma rede bem mais vulneráveis, possibilitando conectar qualquer tipo de dispositivo, algumas vezes para uso indevido. Por um lado é fácil conectar estes dispositivos aos computadores e inserir ou extrair dados e informações, mas por outro, é muito difícil controlar estes acessos quando a rede de computadores da empresa é extensa e principalmente quando as estações estão espalhadas em diversos locais.

Os administradores de rede precisam fazer levantamentos da infra-estrutura da rede com certa freqüência para que os componentes desta rede possam ser atualizados, bem como verificar se as estações não estão sendo lesadas ou danificadas ou até mesmo se não estão sendo extraídas informações e dados da rede. Estes levantamentos são chamados de Gerência de Patrimônio de Rede.

Como já existem diversas formas de controlar e bloquear o acesso das estações à internet e a outros meios, precisa-se também ter controle para evitar que periféricos insiram ou extraiam informações e dados.

Baseado nesta dificuldade de gerenciar os componentes das estações de uma rede, o *software* proposto neste trabalho, utilizando o protocolo *Session Message Block* (SMB) como forma de comunicação entre estação de gerência e estações gerenciadas, visa disponibilizar de forma clara, rápida e objetiva para o administrador, todos os periféricos e componentes das estações que estão sendo administradas dentro de uma rede. A disponibilização das

informações será feita de forma automática na inicialização da estação ou através de requisições feitas pelo administrador para as estações.

1.1 MOTIVAÇÃO

O assunto Gerência de Patrimônio de Redes foi o principal fator de motivação deste trabalho visto a dificuldade que se tem hoje com esta área da gerência de redes. O desafio de utilizar um protocolo fora do convencional também foi decisivo, porque precisava testar todas as suas funcionalidades e limitações e também porque torna mais amplo o escopo nos estudos de protocolos de gerência de redes. Esta também vem sendo uma dificuldade nas empresas que possuem uma rede de tamanho médio e que precisam fazer o levantamento de patrimônio com baixo custo. O desafio de fazer um *software* de fácil instalação, sem complicações e que possa ser utilizado por qualquer usuário mais experiente em redes e informática foi fator decisivo para e elaboração deste, ou seja, apresentar o inventário dos componentes das estações de uma rede de forma clara e objetiva, sem complicações para que qualquer usuário com um nível de experiência mínima possa alcançar os seus objetivos de saber o que tem instalado em cada máquina da sua rede.

Como citado por JACOBOWSKI (2004), a utilização da plataforma WMI no seu trabalho, limitou-se a utilização do serviço WMI, ou seja, as estações precisavam estar rodando o serviço WMI. Esta limitação motivou ainda mais o desafio de apenas instalar uma aplicação nas estações da rede e esta já fornecer as informações necessárias para fazer o levantamento de patrimônio de rede.

1.2 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver a especificação e a implementação de um protótipo de *software* para gerência de patrimônio de rede utilizando o protocolo *Session Message Block* (SMB).

Especificadamente pretende-se atingir os objetivos descritos abaixo:

a) disponibilizar nas estações da rede um aplicativo para rastrear os periféricos e dispositivos instalados nas mesmas;
b) disponibilizar um aplicativo para centralizar numa estação de gerência as informações colhidas nas estações da rede;
c) ter disponível a qualquer momento de cada estação da rede o inventário de *hardware* de cada estação. Isso deve ser possível através de requisição feita pelo administrador de uma estação de gerência;
d) notificar o administrador da rede quando algum dos componentes acima for retirado ou trocado de uma determinada estação, informando qual estação e o que foi alterado.

1.3 ESTRUTURA DO TRABALHO

O capítulo 1 apresenta a estrutura geral do trabalho, a introdução, os objetivos que se quer alcançar, a localização dos assuntos abordados e a organização do trabalho.

No capítulo 2 é apresentado um estudo sobre Gerência de Redes, abordando a importância do gerente de rede, de como é formado uma equipe de gerência, do que se trata a gerência de patrimônio de redes e também como funciona o Registro do Windows. Os

protocolos para gerência de redes também são apresentados neste capítulo, dando uma visão geral das várias maneiras que se pode administrar uma rede.

No capítulo 3 é apresentado o protocolo de gerência *Session Message Block* (SMB) com todas as suas propriedades e funcionalidades, bem como suas aplicações e forma de operação.

O capítulo 4 descreve como foi desenvolvido o protótipo, apresentando os seus requisitos, a especificação através dos diagramas de classe, de atividades e de casos de uso. Também são apresentadas as suas funcionalidades e todo o esquema de funcionamento.

O capítulo 5 apresenta as conclusões sobre o trabalho e sugestões para extensões e trabalhos futuros.

2 GERÊNCIA DE REDES

O objetivo de Gerência de Redes é monitorar e controlar os elementos da rede, assegurando um certo nível de qualidade de serviço. Para realizar esta tarefa, os gerentes de rede são geralmente auxiliados por um sistema de Gerência de Redes. Um sistema de gerência de rede pode ser definido como uma coleção de ferramentas integradas para a monitoração e controle da rede. Este sistema oferece uma interface única, com informações sobre a rede e pode oferecer também um conjunto poderoso e amigável de comandos que são usados para executar quase todas as tarefas de gerência da rede.

A arquitetura geral dos sistemas de gerência de redes apresenta quatro componentes básicos: os elementos gerenciados, as estações de gerência, os protocolos de gerência e as informações de gerência (LOPES; SAUVÉ; NICOLLETTI, 2003).

Os **elementos gerenciados** possuem um *software* especial chamado agente. Este *software* permite que o equipamento seja monitorado e controlado através de uma ou mais estações de gerência. Os elementos gerenciados constituem os componentes da rede que precisam operar adequadamente para que a rede ofereça os serviços para os quais foi projetada.

Exemplos de elementos gerenciados incluem:

a) *Hardware*: equipamentos de interconexão, enlaces de comunicação, hospedeiros, *nobreaks*, *modems*, impressoras etc;
b) *Software*: sistemas operacionais, servidores de bancos de dados, servidores *Web*, servidores de e-mail etc.

Em um sistema de gerência de redes deve haver pelo menos, uma **estação de gerência.** As estações de gerência são hospedeiros munidos de *software* necessário para gerenciar a rede. Para facilitar a vida dos especialistas em gerência, as estações de gerência são

normalmente centralizadas; aliás, é muito freqüente que haja uma única estação de gerência. Só se recorre a várias estações de gerência quando a escala da rede impede que seja gerenciada por uma única estação.

O *software* presente na estação de gerência que conversa diretamente com os agentes nos elementos gerenciados é chamado de “gerente”. A estação de gerência pode obter informação de gerência presente nos elementos gerenciados através de uma sondagem regular dos agentes ou até mesmo recebendo informação enviada diretamente pelos agentes; a estação também pode alterar o estado de elementos gerenciados remotos. Adicionalmente, a estação de gerência possui uma interface com o servidor especialmente projetada para facilitar a gerência da rede. Em sistemas de gerência distribuídos existem duas ou mais estações de gerência. Em sistemas centralizados, o que é mais comum, existe apenas uma. Chama-se de gerente o *software* da estação de gerência que conversa diretamente com os agentes nos elementos gerenciados, seja com o objetivo de monitorá-los, seja com o objetivo de controlá-los. A estação de gerência oferece uma interface através da qual servidores autorizados podem gerenciar a rede (LOPES; SAUVÉ; NICOLLETTI, 2003).

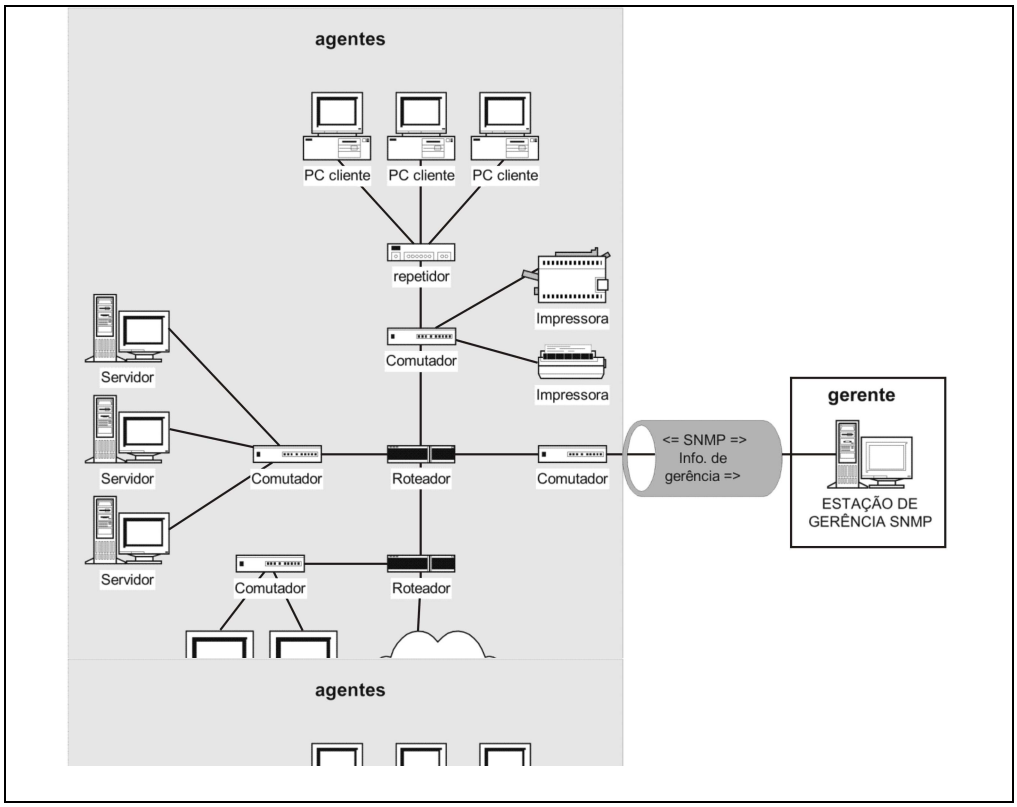
Para que a troca de informações entre gerente e agentes seja possível é necessário que eles falem o mesmo idioma. O idioma que eles falam é um **protocolo de gerência**. Este protocolo permite operações de monitoramento e controle.

O gerente e os agentes trocam informação de gerência usando um protocolo de gerência. O protocolo inclui operações de monitoramento (leitura de informação de gerência) e operações de controle (alteração de informação de gerência presente no elemento gerenciado). Um exemplo de operação de monitoramento ocorre quando o gerente pergunta a um roteador: “Qual é a quantidade de erros ocorrendo no fluxo de entrada na interface número 17?” Um exemplo de uma operação de controle ocorre quando o gerente diz ao roteador: “Desligue sua interface número 17.”

Gerentes e agentes podem trocar informações, mas não qualquer tipo de informação. As **informações de gerência** definem os dados que podem ser referenciados em operações do protocolo de gerência, isto é, dados sobre os quais gerente e agente conversam. As conversas entre gerente e agentes envolvem informação de gerência. Essa informação define os dados que podem ser referenciados em conversas gerente-agente (LOPES; SAUVÉ; NICOLLETTI, 2003).

Exemplos incluem: informação de erro de transmissão e recepção em enlaces de comunicação, status de um enlace de comunicação, temperatura de um roteador e tensão de entrada de um equipamento *nobreak*.

Na figura 1 há roteadores, comutadores, repetidores, impressoras, servidores e estações clientes. Todos estes equipamentos podem ter agentes instalados. A estação de gerência deve obter informações de gerência destes agentes usando o protocolo *Simple Network Management Protocol* (SNMP), ou qualquer outro implementado pelo *software* gerenciador.



Fonte: Lopes, Sauvé, Nicolletti (2003, p. 5)

Figura 1 – Elementos de uma arquitetura geral de solução de gerência

A padronização de solução de gerência mais usada chama-se *Internet-Standart Network Management Framework*. Esta solução é mais conhecida como gerência SNMP. *Simple Network Management Protocol* (SNMP) é o protocolo de gerência deste padrão. Este padrão descreve não apenas o protocolo de gerência, mas também um conjunto de regras que são usadas para definir as informações de gerência e um conjunto inicial de informações de gerência que já podem ser utilizadas (SOUZA, 1999).

Através da estação de gerência pode-se obter informações tais como: taxa de erros, estado operacional de enlaces e equipamentos, utilização de enlace, dentre outras. Tão importante quanto obter estas informações é saber interpretá-las. Por exemplo, em um determinado momento, a estação de gerência informa que a taxa de erros de um certo enlace é 1%.

Para muitas informações de gerência seta-se valores limites. Se o valor da informação obtida for maior que o limite estabelecido significa que algo anormal está ocorrendo na rede.

Estes limites são chamados de limiares (*thresholds*). Assim, quando diz-se que limiares foram excedidos, quer-se dizer que obtive-se valores de informações de gerência que não estão dentro da faixa de normalidade e, portanto, são indicativos de problemas. Limiares excedidos e outros eventos podem gerar alarmes na estação de gerência. Quando a estação de gerência percebe que uma interface parou de operar, por exemplo, um alarme pode ser gerado.

Além do sistema de gerência de redes, outras ferramentas nos auxiliam a gerenciar a rede. Dentre elas encontram-se analisadores de protocolos, e outras ferramentas mais simples, como os comandos *ping*, *traceroute* e *netstat*, disponíveis para vários sistemas operacionais.

Com os analisadores de protocolos, pode-se ver quais dados estão trafegando na rede. Eles nos permitem tirar um raio-x da rede, sendo portanto, ferramentas importantes de gerência. Certas tarefas da gerência só podem ser realizadas com o auxílio de um analisador de protocolos (LOPES; SAUVÉ; NICOLLETTI, 2003).

2.1 O PAPEL DE GERENTE DE REDES.

Um dos objetivos da gerência de redes é prevenir e solucionar problemas na rede. Geralmente esta tarefa é realizada por uma equipe. Não existe uma regra rígida sobre os profissionais que fazem parte desta equipe. Cada organização tem autonomia para criar seu próprio time de gerência de redes de acordo com suas conveniências. Porém, é comum que nesta equipe existam profissionais que executem quatro tarefas distintas: o pessoal do *help desk*, o operador de rede, a equipe de suporte técnico e o gerente da equipe de gerência. Quando os servidores enfrentam problemas relacionados à tecnologia de informação, eles pedem auxílio ao *help desk*. Em algumas organizações o *help desk* é composto por apenas uma pessoa, que atende chamadas telefônicas de usuários e tem certo grau de conhecimento para lidar com alguns problemas que forem reportados. Em organizações maiores, o *help desk*

é composto por um grupo de pessoas um pouco mais especializadas, auxiliadas por aplicações que ajudam a gerenciar os problemas reportados. Além disso, esta equipe pode ser auxiliada por outras ferramentas que ofereçam informações que possam ajudar a localizar e/ou solucionar problemas. Por exemplo: ferramentas que apresentam o estado operacional das interfaces e equipamentos da rede. Geralmente, esta equipe é capaz de solucionar os problemas mais simples e os erros cometidos pelos próprios usuários. Quando o *help desk* existe, os usuários nunca têm contato com a equipe de suporte técnico ou com o operador de rede.

O gerente da equipe de gerência de rede não é, necessariamente, um técnico em redes. O gerente tem um certo conhecimento em redes, mas não no nível do suporte técnico. Dentre as atividades desde gerente encontram-se: avaliar o desempenho da sua equipe de suporte, solicitar compra de equipamentos, aplicações ou outros recursos necessários, providenciar treinamento adequado para a equipe e reescalonar a solução de problemas para outros membros da equipe quando a solução demora. Para avaliar o desempenho da equipe de gerência, o gerente pode se valer de certas métricas tais como: o tempo médio entre falhas e o tempo médio para correção de falhas na rede, percentual de problemas resolvidos em menos de 1 hora, entre outras (LOPES; SAUVÉ; NICOLLETTI, 2003).

2.2 GERÊNCIA DE PATRIMÔNIO DE REDE

Gerenciadores de redes são sistemas e aplicativos que permitem detectar e controlar problemas que ocorram em hardwares e softwares que fazem parte da rede. Sistemas de gerenciamento permitem tanto o gerenciamento de problemas como também o controle de equipamentos e *softwares* que fazem parte da rede (inventário dos componentes da rede). A

utilização de sistemas de gerenciamento de redes é uma forte tendência, que visa monitorar e detectar problemas de uma forma pró-ativa, resolvendo-os antes que gerem prejuízos operacionais para a empresa. Os sistemas de gerenciamento funcionam por meio de sub-rotinas incluídas nas aplicações que são processadas nos equipamentos. Essas sub-rotinas geram pacotes de informações, chamados de alertas, que são transmitidos a um computador central (estação de gerência) responsável pelo gerenciamento, quando ocorrem problemas. Os alertas de gerenciamento são coletados pelo gerenciador central que mostra num monitor o estado dos diversos componentes da rede. Se o gerenciador central recebe um alerta indicando falha em algum componente, esse componente pode aparecer no monitor na cor vermelha e piscando, indicando falha a ser corrigida. Ou seja, quando ocorre falha em um programa ou componente monitorado, a aplicação detecta e envia o alerta para a estação de gerenciamento. O operador de gerenciamento de rede pode, portanto, perceber defeitos e tomar ações corretivas antes mesmo que o servidor perceba a falha. O gerenciamento pode ajudar na administração, planejamento e expansão da rede, pois além do envio de alerta também efetua outros controle como gerar dados para as mais diversas análises possíveis e quais os pontos da rede devem ter mais assistência.

O controle de inventário evita despesas desnecessárias e gera economia para a empresa, pois os recursos são melhor aproveitados, evitando desperdícios. Estudos mostraram que as despesas com redes de computadores são divididas percentualmente da seguinte forma:

a) 34% das despesas são para administrar a rede;
b) 23% das despesas são para efetuar o gerenciamento físico dos equipamentos;
c) 21% das despesas são gastos na identificação de falhas;
d) 21% das despesas são gastos no gerenciamento como um todo.

A administração e o acompanhamento freqüente do inventário são de fundamental importância, apesar de poucas empresas fazerem este trabalho. É preciso ter uma ou mais

pessoas responsáveis por esses controles e acompanhamentos. Este procedimento poderá poupar muitas despesas que são geradas pela falta de controle dos recursos existentes, tanto em nível de recursos de *hardware*, de *software*, de serviços, quanto de canais de comunicação contratados para atender às redes de computadores (SOUZA, 1999).

A gerência de patrimônio baseia-se nas informações contidas em cada estação gerenciada da rede, e com estas informações centraliza-as numa estação de gerência. As informações das estações da rede estão dentro do Registro do Windows, que é visto a seguir.

2.3 REGISTRO DO WINDOWS

O Registro do Windows é uma espécie de banco de dados, onde são armazenadas as informações sobre todos os programas instalados, estrutura de diretórios, informações do computador e *drivers*. Ele existe desde as versões do sistema operacional Windows 3.x da Microsoft, mas passou a ser utilizado como padrão a partir do Windows 95.

O Registro, numa comparação grosseira, pode ser entendido como "sangue do Windows", pois todas as atividades no sistema operacional dependem da sua existência. Um exemplo simples é que, através do Registro, o sistema consegue saber onde os programas estão armazenados e quais arquivos se relacionam a eles. É por isso que um simples corrompimento do Registro do Windows faz o sistema parar. Se isso ocorre, não será possível encontrar programas, *drivers* e configurações. Por ser uma parte crítica do Windows, a Microsoft preferiu não disponibilizar um acesso fácil ao Registro do Windows. Isso porque usuários não preparados poderiam facilmente causar danos ao sistema, tentando fazer reparos ou querendo saber para que servem as linhas estranhas do Registro (INFOWESTER 2004).

2.3.1 A estrutura do Registro do Windows

O Registro do Windows é composto por cinco chaves raiz, onde cada uma tem uma finalidade:

a) HKEY_CLASSES_ROOT - esta chave, na verdade, não é muito importante a princípio. É um atalho para a chave HKEY_LOCAL_MACHINE\SOFTWARE\Classes e sua finalidade é manter compatibilidade com programas antigos, que rodam a 16 bits;

b) HKEY_CURRENT_USER - esta chave é um atalho para a chave HKEY_USERS\infowester, onde *infowester* deve ser o nome do servidor do Windows. Ela mostra somente informações do servidor atual do sistema, como configurações personalizadas;

c) HKEY_LOCAL_MACHINE - esta é a chave mais importante do Registro, pois nela é que estão as informações sobre programas e hardware. Para se ter noção da importante desta chave, seus dados são guardados num arquivo chamado system.dat. Esta chave é dividida numa estrutura que indica onde estão os dados. As informações estão organizadas por tipo: em *HARDWARE* estão informações relativas ao hardware do computador, como portas paralelas, interfaces SCSI, etc; em *SECURITY*, estão informações de segurança; em *SOFTWARE* estão as informações sobre os programas instalados no computador;

d) HKEY_USERS - no Windows é possível ter vários usuários num único computador. A função desta chave é guardar informações de cada um deles. Quando o sistema está configurado apenas para um servidor (muito comum no Windows 95/98), esta chave possui apenas uma entrada, de nome *default* ou padrão. Todas as limitações dos usuários, assim como todas as suas configurações

podem ser manipuladas nesta chave;

e) HKEY_CURRENT_CONFIG - é um atalho que contém configurações do servidor atual do computador relativas ao hardware. Este atalho é útil quando é necessário procurar informações do servidor que está logado, pois todas as suas informações aparecem nesta chave.

É comum ter que alterar algumas informações no Registro do Windows para executar uma configuração em especial, ou mesmo para guardar informações de um programa específico que está instalado no computador e utilizar o Registro do Windows como base de informação para configuração (INFOWESTER 2004).

2.4 PROTOCOLOS DE GERÊNCIA

A seguir serão analisadas as funcionalidades de dois dos principais protocolos de gerência que existem, o *Simple Network Management Protocol* (SNMP) e o protocolo *Common Management Information Protocol* (CMIP).

2.4.1 SNMP

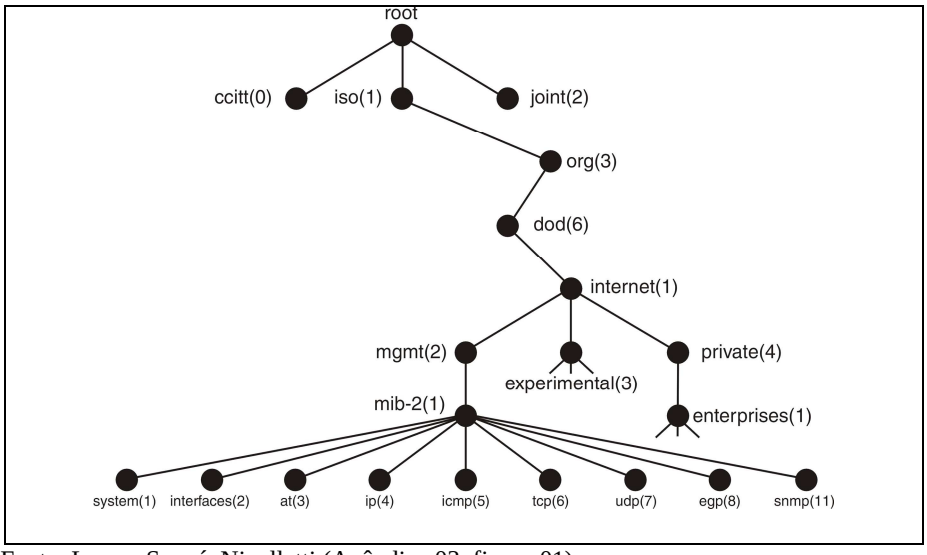
O SNMP é essencialmente um protocolo de solicitação/resposta especializado que suporta dois tipos de mensagens de solicitação: *GET* e *SET*. O primeiro tipo é usado para obter alguma informação sobre o estado de um certo nó da rede, e o outro é usado para armazenar uma nova informação sobre o estado de algum outro nó da rede. O SNMP também suporta uma terceira opção, o *GET-NEXT*. Por exemplo, há uma variável de gerência chamada *ifAdminStatus* que determina o estado desejado para um enlace de comunicação. Se o valor “1” for escrito nessa variável num determinado agente e para uma determinada

interface de comunicação, isso significa que desejamos que o enlace esteja "*up*" ou "em funcionamento". Ao escrever o valor "2" correspondendo a "*down*" nessa variável, o enlace é desabilitado.

O SNMP é usado de uma maneira simples. Um administrador interage com um programa cliente que mostra informações sobre a rede. Esse programa cliente geralmente tem uma interface gráfica, a qual tem o mesmo papel de um navegador *web*. Sempre que um administrador seleciona uma informação específica que deseja ver, o programa cliente usa o SNMP para solicitar essa informação do nó em questão. Um servidor SNMP que está rodando naquele nó recebe a solicitação, localiza a informação correta e a retorna ao programa cliente, que então a mostra para o servidor.

Existe apenas uma complicação nesse cenário bastante simples: exatamente como o cliente indica que informação ele deseja, e, do mesmo modo, como o servidor sabe qual a variável da memória ele deve ler para satisfazer a solicitação? A resposta está no fato de que o SNMP depende de uma especificação adicional chamada de *Management Informatino Base* (MIB). A MIB define informações específicas que podem ser obtidas de um nó da rede (SOUZA, Apêndice 3, 1999).

As variáveis de gerência que podem ser manipuladas pelo protocolo SNMP formam uma base de dados virtual acessível ao agente de um elemento gerenciado. Devido ao grande número de variáveis de gerência (existem milhares de variáveis de gerência no mundo SNMP), o espaço de nomes dessas variáveis está estruturado hierarquicamente, como mostra a Figura 2.



Fonte: Lopes, Sauvé, Nicolletti (Apêndice 03, figura 01)

Figura 2 – Estrutura hierárquica das variáveis de gerência

Nessa árvore, alguns órgãos de padronização internacional (ISO, CCITT) têm seu espaço logo abaixo da raiz. Cada objeto da árvore possui um rótulo com uma descrição textual e um número. Por exemplo, há um objeto chamado “mgmt” com número 2, abaixo de um objeto com rótulo “internet” cujo número é 1. A árvore mostrada na Figura 2 exibe apenas o topo da árvore de todas as variáveis de gerência do mundo SNMP.

Os objetos da árvore que não são folhas agrupam um conjunto de objetos relacionados. Os objetos descrevem a informação mantida nos agentes. Uma instância de um objeto (também chamada de variável) é o que realmente é manipulado pelo protocolo SNMP. Para entender a diferença entre objeto e instância de objeto, devemos observar que, no mundo SNMP, objetos podem ser simples (ou escalares) ou podem fazer parte de uma linha de uma tabela.

Vejamos primeiro os objetos simples. Um objeto é identificado unicamente através do caminho percorrido da raiz da árvore até o objeto em questão. Por exemplo, o objeto com *Object Identifier* (OID) identifica um objeto simples que possui uma única instância. A instância tem o nome: iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0. Nesse caso, a instância (única) do objeto representa a descrição do elemento gerenciado. Um valor para esta variável poderia ser “IBM 8271 EtherStreamer Switch”. O nome numérico 1.3.6.1.2.1.1.1.0 é

equivalente ao nome simbólico dado anteriormente.

Em muitos casos, um determinado objeto possui várias instâncias. Consideramos, por exemplo, o objeto: iso.org.dod.internet.mgmt.mib-2.interfaces.iftable.ifEntry.ifInOctets. Esse objeto informa o número de bytes que foram recebidos numa determinada interface de rede do elemento gerenciado em questão.

A versão atual da MIB, chamada de MIB-II, organiza as variáveis em 10 diferentes grupos, dentre eles:

a) *system*: parâmetros gerais de um sistema (nó) como um todo, incluindo onde ele está localizado, a quanto tempo ele funciona e o nome do sistema;

b) *interfaces*: informações sobre todas as interfaces de rede presentes no nó, como por exemplo, o endereço físico de cada interface, quantos pacotes foram enviados e recebidos em cada interface;

c) *address translation*: informações sobre o *Address Resolution Protocol* (ARP), e em particular, sobre o conteúdo e sua tabela de tradução de endereços;

d) IP: variáveis de IP, incluindo sua tabela de roteamento, quantos datagramas foram encaminhados com sucesso e as estatísticas sobre a remontagem de datagramas. Inclui também a contagem de datagramas descartadas pelo IP, por qualquer que seja a razão;

e) TCP: informações sobre as conexões TCP, tais como o número de aberturas passivas e ativas, o número de *resets*, o número de *timeouts*, a configuração padrão para o *timeout* e assim por diante. A informação referente a uma conexão existe apenas durante o tempo em que a conexão existir;

f) UDP: informações sobre o tráfico UDP, incluindo o número total de datagramas UDP que foram enviados e recebidos.

Sendo assim, o cliente SNMP coloca o identificador ASN.1 para a variável MIB que ele deseja na mensagem de solicitação, e envia essa mensagem para o servidor. O servidor então mapeia esse identificador em uma variável local, obtém o valor atual contido nesta

variável e usa o *Abstract Syntax Notation One* (ASN.1) para codificar o valor que envia de volta ao cliente. Muitas das variáveis MIB são tabelas ou estruturas. Estas variáveis compostas explicam a razão para a operação GET-NEXT do SNMP. Essa operação, quando aplicada a um identificador de uma variável em particular, retorna o valor daquela variável e o identificador da próxima variável, por exemplo, o próximo item em uma tabela ou próximo campo de uma estrutura. Essa operação auxilia o cliente a passar por todos os elementos de uma tabela ou de uma estrutura.

O SNMP existe em 3 versões, embora a versão inicial seja a única versão realmente difundida até hoje (SOUZA, Apêndice 3, 1999).

2.4.2 CMIP

CMIP é um modelo da OSI (*Open Systems Interconnection*) que define como criar um sistema de gerência comum da rede. Comparando o CMIP ao SNMP, o CMIP é mais complexo. De fato, o CMIP é usado realmente por fornecedores de serviço de algumas operadoras de telecomunicações para a gerência de rede. Já o SNMP é um protocolo de internet projetado especificamente para redes de TCP/IP para ser utilizado em redes corporativas. O modelo de gerência da OSI define os sistemas que são controlados, e define sistemas de gerência. Nos sistemas controlados funcionam os agentes que recolhem a informação sobre processos e se comunicam com os sistemas de gerência. Um processo funciona em nós que coletam a informação de gerência dos processos que funcionam em cada camada da OSI. As mudanças podem também ser aplicadas nas camadas. Cada nó tem um *Management Information Base* (MIB), que é uma coleção dos objetos que contém a informação do nó. Um *System Management Application Process* (SMAP) fornece a interface

que dá a informação por parte de MIBs. As SMAPs conversam com outras SMAPs através da rede. Um *System Management Application Entity* (SMAE) suporta uma comunicação de SMAP, e SMAEs usam o CMIP para trocar dados entre os nós. A CMIP dá forma a um mapa de caminhos para projetar um sistema de gerência da rede, mas as especificações de relação reais estão em *Common Management Information Service* (CMIS).

A *International Standars Organization* (ISO) define as funções de gerenciamento em cinco classes:

a) gerência de contabilidade: monitora a e carga de uso da rede para fins de tarifação;
b) gerência da configuração: analisa e gerencia recursos do sistema e gera informações de gerência;
c) gerência de falha: detecta e corrige falhas na rede;
d) gerência de desempenho: monitora e ajusta o desempenho da rede;
e) gerência de segurança: autentica usuários, detecta intrusos, e transmite dados com segurança.

O CMIS fornece os caminhos para compartilhar o gerenciamento da informação no contexto do CMIP. A *Common Information Model* (CIM) traz a interoperabilidade entre os protocolos de gerência em geral. Existem várias características novas que vão além das características do SNMP e do CMIP, quando provêm compatibilidade inversa. (LINKTIONAY.COM, 2001).

A especificação de CMIP para redes TCP/IP é chamada *CMIP over TCP* (CMOT) e a versão para IEEE 802 LAN's é chamada *CMIP over LLC* (CMOL). CMIP/CMIS são propostos como protocolos competindo com o SNMP no conjunto TCP/IP. A informação da gerência é trocada entre a aplicação de gerência de rede e os agentes de gerência através dos objetos controlados. Os objetos controlados são uma característica de um dispositivo controlado que possa ser monitorado, modificado ou controlado e podem ser usados para

executar tarefas.

O CMIP não especifica a funcionalidade da aplicação de gerência de rede, define somente o mecanismo da troca de informação dos objetos controlados e não como a informação deve ser usada ou interpretada:

Conforme o site JAVVIN (2005), as principais vantagens do CMIP sobre o SNMP são:

a) As variáveis do CMIP não armazenam somente informação, elas também podem ser usadas para executar tarefas. Isto é impossível no SNMP;
b) O CMIP é um sistema mais seguro porque suporta a autorização, o controle de acesso, e os registros da segurança;
c) O CMIP fornece potencialidades poderosas que permitem que as aplicações de gerência realizem mais com um único pedido;
d) CMIP fornece um relatório de condições incomuns da rede mais esclarecedor.

3 SMB

O *Session Message Block* (SMB) é um protocolo para compartilhar arquivos, impressoras, portas seriais, e para comunicação entre computadores. O SMB é um protocolo cliente-servidor e *request-response.* Na figura 3 pode-se observar como o SMB trabalha.

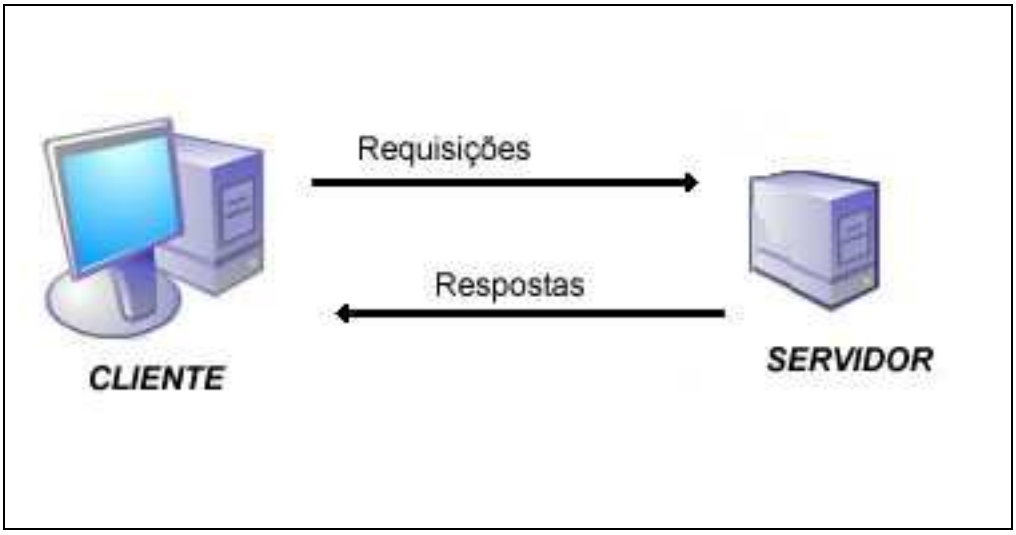


Figura 3 – Forma de trabalho do protocolo SMB.

Os clientes conectam-se ao servidor usando TCP/IP ou NetBEUI. Uma vez estabelecida a conexão, os clientes podem enviar comandos (SMB) para o servidor para que acessem pastas compartilhadas, leiam e escrevam arquivos e todas as possível funcionalidades que se pode fazer com um sistema de arquivos.

O SMB é definido por dois modelos diferentes de segurança. O ***Sharel Level,*** onde a proteção é aplicada na parte compartilhada de um servidor, ou seja, cada parte pode ter uma senha e o cliente só precisa ter uma senha se quiser acessar arquivos desta parte do servidor, e este foi o primeiro modelo de segurança do SMB. O outro modelo é o ***User Level***, onde a proteção é aplicada nos arquivos individuais de cada parte baseados nos direitos de acesso do servidor. Cada cliente precisa se autenticar no servidor. Depois de autenticado, o cliente ganha uma identificação única (UID) que deve ser utilizada em todos os próximos acesso ao

servidor (SAMBA, 2002).

Um cliente é um sistema que solicita serviços de arquivos da rede e um servidor é um sistema que entrega serviços de arquivos da rede. Os clientes e os servidores são sistemas lógicos: um cliente e um servidor podem coexistir em um único sistema físico, ou seja, no mesmo computador. Os clientes são responsáveis por dirigir seus pedidos ao servidor apropriado. O mecanismo de endereçamento de rede ou convenção de nomes com que o servidor é identificado é tratado pela própria rede e não faremos abordagem sobre isso. Cada servidor disponibiliza à rede uma estrutura *self-contained* de arquivos. Não há dependência de outros servidores, nem por armazenamento nem por serviço. Um arquivo deve ser alocado inteiramente por um único servidor. Um arquivo compartilhado requer uma autenticação do servidor antes que os acessos pelos clientes sejam permitidos. Cada processo no servidor autentica seu próprio cliente. Um cliente deve efetuar a autenticação em cada servidor que deseja utilizar. Este modelo de autenticação supõe que a LAN conecta os sistemas autônomos que disponibilizarem algum subconjunto de seus arquivos locais aos servidores remotos. Os seguintes ambientes existem no ambiente de compartilhamento de arquivo do protocolo:

a) *Virtual Circuit Environment* - consiste em um Circuito Virtual estabelecido entre um sistema do cliente consumidor e o sistema do servidor. Os clientes podem ter somente um único pedido ativo em qualquer tempo, isto é, um segundo pedido não pode ser iniciado até que a resposta ao primeiro esteja recebida. Um Circuito Virtual representa os dados que usam o serviço de transporte;
b) *Logon Environment* - é representado por uma árvore de ID (TID). Um TID identifica excepcionalmente um arquivo que compartilha uma conexão entre um cliente e um servidor. Identifica também o espaço e o tipo de acessos permitidos através da conexão;
c) *Ambiente Process* – é representado por um identificador de processo (PID). Um

PID identifica excepcionalmente um processo do cliente dentro de um ambiente;

d) *File Environment* - é representado por um *file handle* (FID). Um FID identifica um arquivo aberto que é único dentro de um ambiente.

Quando um destes ambientes é terminado, todos os ambientes contidos dentro dele estarão terminados. Por exemplo, se um canal virtual for encerrado, todos os PIDs, TIDs e FIDs dentro dele serão invalidados.

Os sistemas podem usar este protocolo para obter ou fornecer serviços de arquivos remotoss em um ambiente da rede. Este protocolo é projetado para permitir que os sistemas acessem os arquivos que residem em sistemas remotos. Quando duas máquinas fazem o primeiro contato pela rede, podem negociar o uso de um nível mais elevado.

Como e quando os servidores criam e destroem processos é, naturalmente, uma implementação da execução e não há nenhuma exigência que este processo seja amarrado à gerência de processo do cliente. Entretanto é necessário que o servidor esteja ciente das atividades da gerência do processo do cliente porque os arquivos são acessados pelo nome do cliente. Conseqüentemente, o arquivo que compartilha o protocolo inclui notificações apropriadas. Todas as mensagens, exceto negociações, incluem uma identificação do processo (PID) para indicar que processo do servidor iniciou um pedido. Os clientes informam aos servidores da criação de um processo novo simplesmente introduzindo um PID novo no diálogo. A destruição do processo deve explicitamente ser indicada por "*Process Exit*", comando específico para esta finalidade. O servidor deve emitir um comando no processo de saída sempre que um processo do cliente é destruído. Isto permite que o servidor se livre de todos os recursos reservados por este processo e possa executar quaisquer atividades locais de gerência de processo que possa ser requerido.

Cada mensagem tem um formato comum. No quadro 1 pode-se ver um exemplo com a linguagem C.

Data type	Field	Value
BYTE	smb_fid[4];	contains 0xFF, 'SMB'
BYTE	Smb_com;	command code
BYTE	smb_rcls;	error code class
BYTE	smb_reh;	reserved (contains AH if DOS INT-24 ERR)
WORD	smb_err;	error code
BYTE	smb_res;	reserved
WORD	smb_res[7];	reserved
WORD	smb_tid;	tree id number
WORD	smb_pid;	caller's process id number
WORD	smb_uid;	user id number
WORD	smb_mid;	multiplex id number
BYTE	smb_wct;	count of parameter words
WORD	smb_vwv[];	variable number words of params
WORD	smb_bcc;	number of data bytes following
BYTE	smb_data[];	data bytes

Quadro 1 – Estrutura de mensagem SMB utilizando a linguagem C.

Para estabelecer uma conexão, deve-se saber qual a finalidade da mesma sendo que para cada finalidade existe uma forma diferente de estabelecer esta conexão:

a) compartilhamento de arquivos: as redes que usam compartilhamento de arquivos do protocolo conterão não somente sistemas multi-usuários com modelos baseados no servidor de proteção, mas os sistemas mono-usuário que não têm nenhum conceito dos UID ou das permissões. Uma vez que estas máquinas são conectadas à rede, estão em um ambiente multi-usuário e necessitam um método do controle de acesso. Primeiro, as máquinas desprotegidas necessitam permissão para fornecer a outras máquinas da rede que têm permissões; segundo, as máquinas desprotegidas necessitam controlar o acesso a seus arquivos por outro. Este protocolo define um mecanismo que habilita o software de rede à fornecer a

proteção onde falta o sistema operacional e suporte à proteção do servidor fornecido pelo sistema operacional. O mecanismo permite também que as máquinas com nenhum conceito do UID demonstrem a autorização de acesso às máquinas que têm um mecanismo de permissão. Finalmente, o protocolo de permissão está projetado de modo que possa ser omitido se ambas as máquinas compartilharem de um mecanismo comum da permissão. Este protocolo, chamado de *tree conection*, não especifica uma interface de usuário;

b) acesso a servidores desprotegidos: a requisição deve ser feita pelo nome da máquina fornecido pelo comando NET USE, e associá-la com o valor de índice retornado pelo servidor. Os pedidos subseqüentes que usam este índice devem incluir somente o caminho relativo à sub-árvore conectada enquanto o servidor trata a sub-árvore como o diretório de raiz. Quando a requisição tem um pedido de acesso ao arquivo para o servidor, localiza através de sua lista dos prefixos para essa máquina e o seleciona. Inclui então o índice associado com este prefixo em seu pedido junto com o restante do caminho. Ele oferece sempre um diretório e todas os arquivos debaixo desse diretório são afetados. Se um arquivo particular estiver dentro da escala de múltiplas ofertas, conectando-se a qualquer uma das escalas da oferta, se ganha o acesso ao arquivo com as permissões especificadas para a oferta nomeada no NET USE. O servidor não verificará para ver se há diretórios com as permissões mais restritivas;

c) acesso a servidores protegidos: os servidores com esquemas baseados na proteção de arquivos interpretarão a *tree connect* com o comando ligeiramente diferente dos sistemas com os esquemas orientados à proteção de arquivos. Eles interpretam o "nome" como um *username* melhor que um *pathname*. Quando este pedido é recebido, o *username* será validado e um TID representando a autenticidade do

servidor, que é retornada. Este TID deve ser incluído em todas as requisições feitas ao servidor. O sistema *permission-based* não necessita executar o comando NET SHARE;

d) comando de negociação: o cliente emite uma lista das primitivas com que pode comunicar-se. A resposta é uma seleção de uma daquelas primitivas (numeradas de 0 à n) ou -1 que indica que nenhum das primitivas são aceitáveis. A mensagem de negociação está ligada ao Circuito Virtual que deve ser enviada. Somente uma mensagem de negociação pode ser enviada: mensagens de negociação subseqüente serão rejeitadas com uma resposta de erro e nenhuma ação será tomada. O protocolo não impõe nenhuma estrutura particular às mensagens;

e) comando de atribuição de atributos no servidor: este comando é usado para determinar a capacidade total do servidor e o espaço livre restante. A distinção entre alocação unitária e blocos do disco permite o uso do protocolo com sistemas operacionais que alocam o espaço de disco nas unidades maiores do que o bloco físico do disco. As unidades de bloco/alocação usadas nesta resposta podem ser independentes do algoritimo físico ou lógico real de bloco/alocação usado internamente pelo servidor. Entretanto, devem refletir a quantidade de espaço no servidor;

f) comando de checagem do caminho: mensagem de checagem do caminho é usada para verificar se um caminho existe e é um diretório. Nenhum erro é retornado se o caminho existir e a requisição tiver o acesso a ele. Os servidores não têm um conceito de "*working directory*", o cliente deve sempre fornecer os caminhos completos (relativo ao TID);

g) comando de conexão com a TID: o caminho/usuário deve ser especificado da raiz da rede. O campo da TID na requisição da mensagem é ignorado pelo servidor. O

tamanho máximo transmitido na resposta indica o tamanho máximo da mensagem que o servidor aceita. O cliente não deve gerar mensagens, nem esperar receber as respostas maiores do que esta. Isto deve ser constante no servidor. Uma *Tree Conect* deve ser emitida para todos os *subtrees* alcançados, mesmo se contém uma senha nula (MICROSOFT CORPORATION, INTEL CORPORATION, 1988).

4 DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo será visto em detalhes as propriedades e funcionalidades do protótipo de *software* que foi implementado. São apresentados os requisitos do software bem como toda a sua especificação através da linguagem de modelagem UML (FOWLER, 2000) e do projeto orientado a objetos. São também apresentados os detalhes da sua implementação e testes realizados numa rede corporativa.

4.1 REQUISITOS DO *SOFTWARE*

Os requisitos do protótipo de *software* são separados em funcionais e não-funcionais. Os requisitos funcionais são:

a) armazenar na própria estação gerenciada o inventário;
b) visualizar na própria estação gerenciada o inventário, clicando-se no ícone do agente;
c) atualizar as informações do inventário na estação gerenciada toda vez que o agente for iniciado;
d) executar o Agente na inicialização da estação gerenciada;
e) deixar o Agente visível na barra de tarefas do Windows num ícone junto ao relógio da barra de tarefas, como *IconTray;*
f) visualizar no Gerenciador todas as máquinas por nome e/ou IP, num menu do tipo *tree-view* onde clicando-se no nome da máquina apareça o inventário dela;
g) solicitar a um determinado agente a atualização daquela estação gerenciada.

Os requisitos não-funcionais para o *software* são:

a) ter uma interface amigável tanto no Agente como no Gerenciador para que seja

fácil o entendimento das informações apresentadas;

b) rodar no sistema operacional Windows 2000 e XP.

4.2 ESPECIFICAÇÃO

Para a especificação do protótipo *software*, foi utilizado a *Unified Modeling Linguage* (UML), utilizando o digrama de caso de uso, de classe, e de atividades. Para a criação destes diagramas foi utilizada a ferramenta *Power Designer*.

4.2.1 Diagrama de casos de uso

Na Figura 4 vê-se o diagrama de casos de uso do *software*, que especifica como o agente e gerenciador interagem para consultar o inventário das estações e repassar ao administrador da rede.

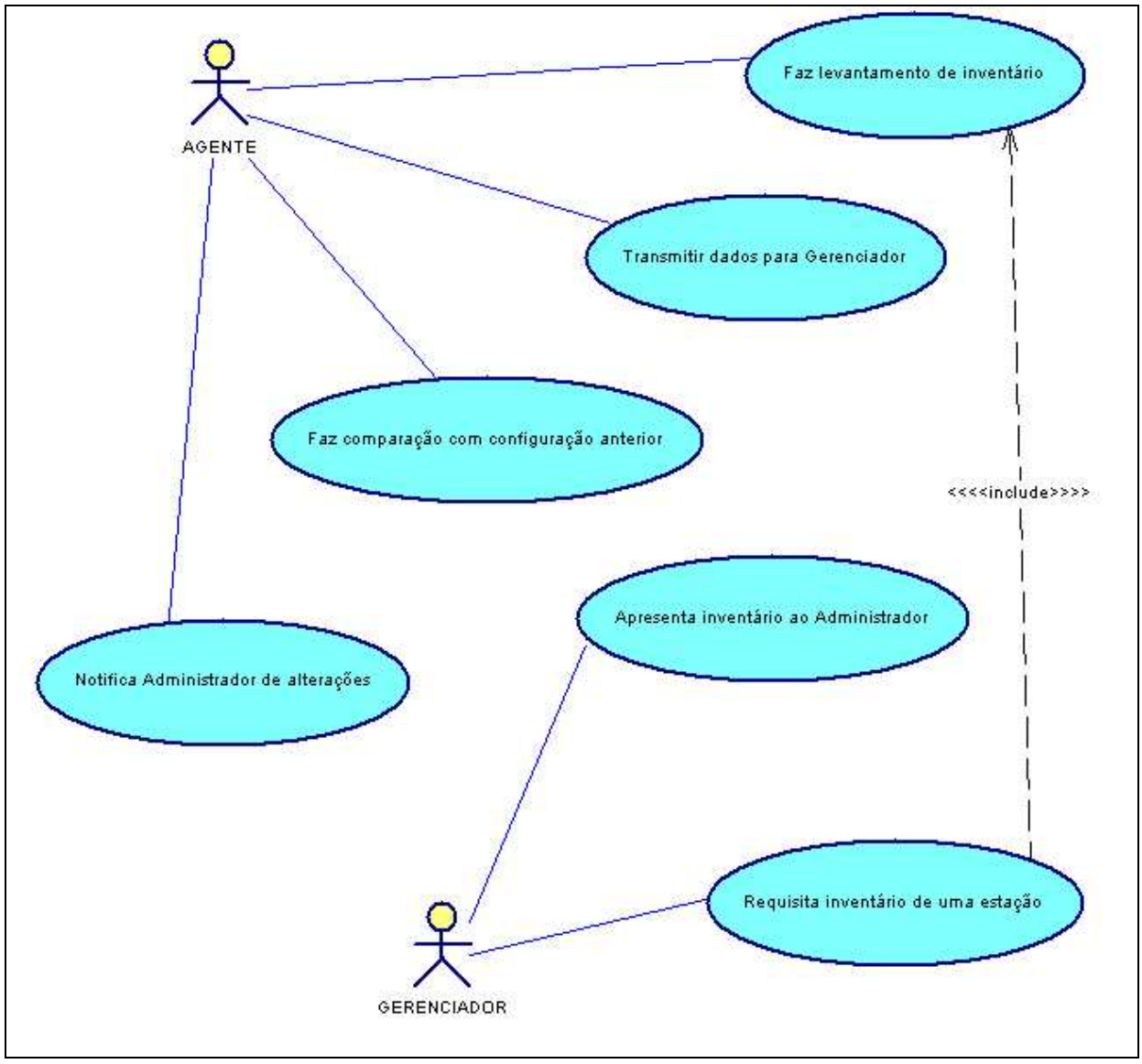


Figura 4 – Diagrama de casos de uso

Cada um destes 6 casos de uso tem a seguinte função:

a) notifica administrador de alterações**:** este caso de uso serve para avisar o administrador de rede caso exista alguma alteração em uma das estações da rede que está sendo gerenciada;

b) faz comparação com configuração anterior**:** este caso de uso verifica o inventário atual com o inventário armazenado da última vez que foi feito pelo agente;

c) faz levantamento de inventário**:** neste caso de uso é feito o levantamento das informações de inventário e armazenado para posterior utilização;

d) transmitir dados para gerenciador**:** este caso de uso faz a transmissão dos dados

para o gerenciador através da rede;

e) apresenta inventário ao administrador**:** neste caso de uso, é apresentado o inventário de cada estação da rede;

f) requisita inventário de uma estação**:** este caso de uso faz uma solicitação para o agente na estação de rede para que seja feito um levantamento de inventário naquele momento.

4.2.2 Diagrama de Classe

O diagrama de classes, descreve como foram divididas as classes na implementação do protótipo, apresentando em detalhes todas as propriedades de cada classe, conforme figura 5.

A classe *TInfHardware* é a principal classe do sistema, sendo instanciada tanto no Agente quanto no Gerente. Esta classe é especializada para cada um dos tipos de informações de hardware que compõe o gerenciador de patrimônio. Todas as classes especializadas da *TInfHardware* são instanciadas junto com a mesma, e caso alguma informação não seja encontrada, de uma determinada classe, o atributo “descrição”, que é comum em todas as classes especializadas, é setado com “Informação não encontrada”.

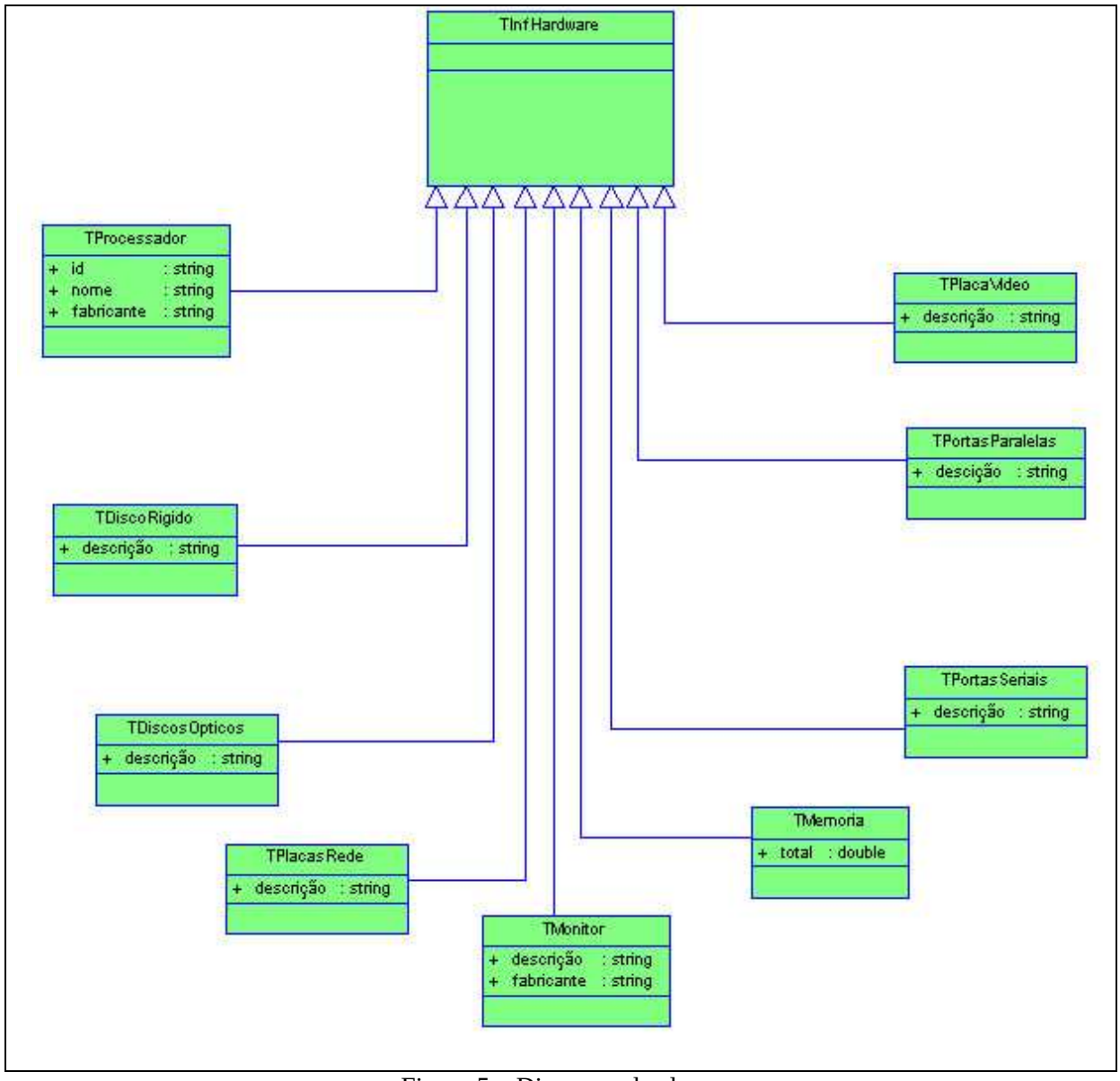


Figura 5 – Diagrama de classe

4.2.3 Diagrama de Atividades

O diagrama de atividades mostra como são os procedimentos do Agente para que as informações cheguem ao administrador de rede, que poder ser visto na figura 6. O caso de uso “Faz Levantamento de Inventário” é um processo disparado automaticamente a cada 30 minutos ou solicitado pelo administrador de redes a qualquer momento.

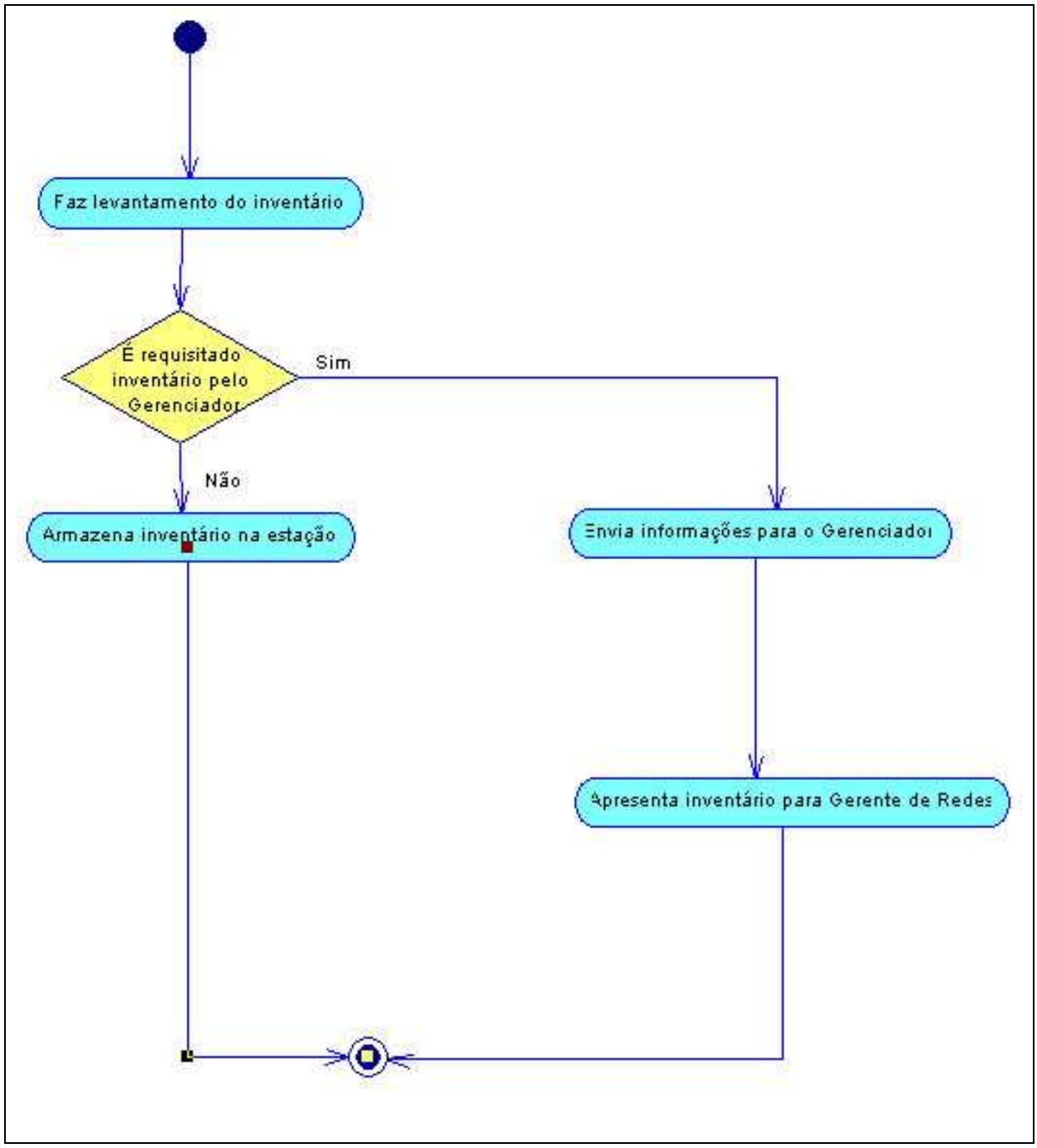


Figura 6 – Diagrama de atividades do Agente

4.3 IMPLEMENTAÇÃO

Neste capítulo é apresentado como foi implementado o protótipo de acordo com a sua especificação descrita na sessão anterior.

4.3.1 Técnicas e Ferramentas Utilizadas

Para a implementação do protótipo foi utilizada a linguagem *Object Pascal DELPHI 7* da Borland (CANTU, 2003). O Delphi apresenta muitas facilidades na manipulação do Registro do Windows, além de ser totalmente compatível com o sistema operacional Windows XP. Para a implementação da leitura do Registro do Windows no Agente, foi criado um componente (uInformacoes.pas) para ler as informações do Registro através de suas chaves específicas que guardam as informações desejadas para fazer o levantamento de patrimônio do computador. As informações lidas por este componente são armazenadas temporariamente em suas propriedades que em seguida, são armazenadas no próprio Registro do Windows numa chave específica de sua propriedade criada para armazenar persistentemente as informações colhidas. Estas informações armazenadas no Registro do Windows pelo componente, servem mais tarde para fazer a comparação entre as informações que estão sendo lidas e as informações lidas anteriormente, permitindo com que se possa efetuar uma comparação entre as duas e afirmar se houve ou não mudanças nos componentes instalados no computador. No quadro 2 pode ser visto uma parte do método LeRegistro, que faz toda a leitura das informações sobre os componentes instalados no computador, onde são lidas as informações sobre os discos instalados.

```
           procedure AchaDiscos(prChave : String);
          var
             wContadorChaves : Integer;
             wSubKeys        : TStrings;
             wValor          : String;
          begin
             if Self.FRegistro.KeyExists(prChave) then
                begin
                   Self.FRegistro.OpenKey(prChave,False);
                   wSubKeys := TStringList.Create;
                   Self.FRegistro.GetKeyNames(wSubKeys);
                   if wSubKeys.Count <> 0 then
                      begin
                         for wContadorChaves := 0 to wSubKeys.Count - 1 do
                            AchaDiscos(prChave + '\' + wSubKeys.Strings[wContadorChaves]);
                      end
                   else
                      begin
                         if Self.FRegistro.ReadString('Type') = cCDRom then
                            begin
                               wValor := Self.FRegistro.ReadString('Identifier');
                               if length(TrimLeft(TrimRight(wValor))) = 0 then
                                  wValor := 'Valor não encontrado.';

                               Self.FDiscoOptico.Add;


TDiscosOpticos(Self.FDiscoOptico.Items[Self.FDiscoOptico.Count-1]).Descricao := wValor
                            end
                         else
                            if Self.FRegistro.ReadString('Type') = cHDD then
                               begin
                                  wValor := Self.FRegistro.ReadString('Identifier');
                                  if length(TrimLeft(TrimRight(wValor))) = 0 then
                                     wValor := 'Valor não encontrado.';

                                  Self.FDiscosRigidos.Add;

TDiscosRigido(Self.FDiscosRigidos.Items[Self.FDiscosRigidos.Count-1]).Descricao            :=
Self.FRegistro.ReadString('Identifier');
                               end;
                      end;
                end;
          end;
```

Quadro 2 – Procedimento que lê as informações do Registro sobre discos no computador.

No Gerente, o componente utilizado no Agente é utilizado para receber as informações e montar estas informações no mesmo formato em que o Agente as tratou, ou seja, o Gerenciador enxerga as informações dos componentes da estação do mesmo modo que foi definido pelo Agente.

Para a transmissão das informações pela rede, as informações tiveram que ser quebradas em partes porque não foi possível transmitir todas as informações de cada estação de uma única só vez, então, como pode ser observador no quadro 3, foi enviado em partes esta “mensagem” para que o Gerente recebesse as informações de inventário completas e as

apresentasse em sua *interface.*

```
procedure TMailSlot.EnviarMensagem(Destino: String; Mensagem: TStrings);
var
   wContador,
   wTamanho : Integer;
begin
   if length(TrimLeft(TrimRight(Destino))) = 0 then
      exit;

   Self.FMicroDestino := Destino;
   wTamanho := length(Mensagem.Text);

   if wTamanho <= 255 then
      begin
         with Self.FMensagemEnviar do
            begin
               Clear;
               AddStrings(Mensagem);
               Insert(0,Self.FMicroOrigem);
               Insert(1,Self.FUsuario);
            end;
         Self.Enviar;
      end
   else
      begin
         with Self.FMensagemEnviar do
            begin
               Clear;
               Add('#INICIO#');
               Insert(0,Self.FMicroOrigem);
               Insert(1,Self.FUsuario);
               Self.Enviar;

               for wContador := 0 to Mensagem.Count - 1 do
                  begin
                     Clear;
                     Add(Mensagem.Strings[wContador]);
                     Insert(0,Self.FMicroOrigem);
                     Insert(1,Self.FUsuario);
                     Self.Enviar;
                  end;
               Clear;
               Add('#FIM#');
               Insert(0,Self.FMicroOrigem);
               Insert(1,Self.FUsuario);
               Self.Enviar;
            end;
      end;
end;
```

Quadro 3 – Procedimento que transmite as informações entre Agente e Gerente

4.3.2 Operacionalidade da implementação

Conforme foi visto na especificação, e em toda a teoria estudada para o desenvolvimento, o protótipo é executado em duas partes distintas, o Agente e o Gerente. O Agente é o programa que é instanciado em todas as estações, que por sua vez, imediatamente após instanciado, faz uma varredura nas chaves do Registro do Windows para colher as informações de inventário e em seguida atualiza os dados já armazenados também no Registro

do Windows sobre os componentes daquele computador, e caso encontre alguma diferença em qualquer um dos componentes encontrados e o que já estava armazenado da varredura anterior, comunica o Gerente no mesmo momento, este esquema é apresentado na Figura 7. O Agente, além de colher as informações do computador e notificar o Gerente quando detectar alguma mudança nos componentes instalados, também é responsável por apresentar ao próprio operador da estação os dados de inventário da estação.

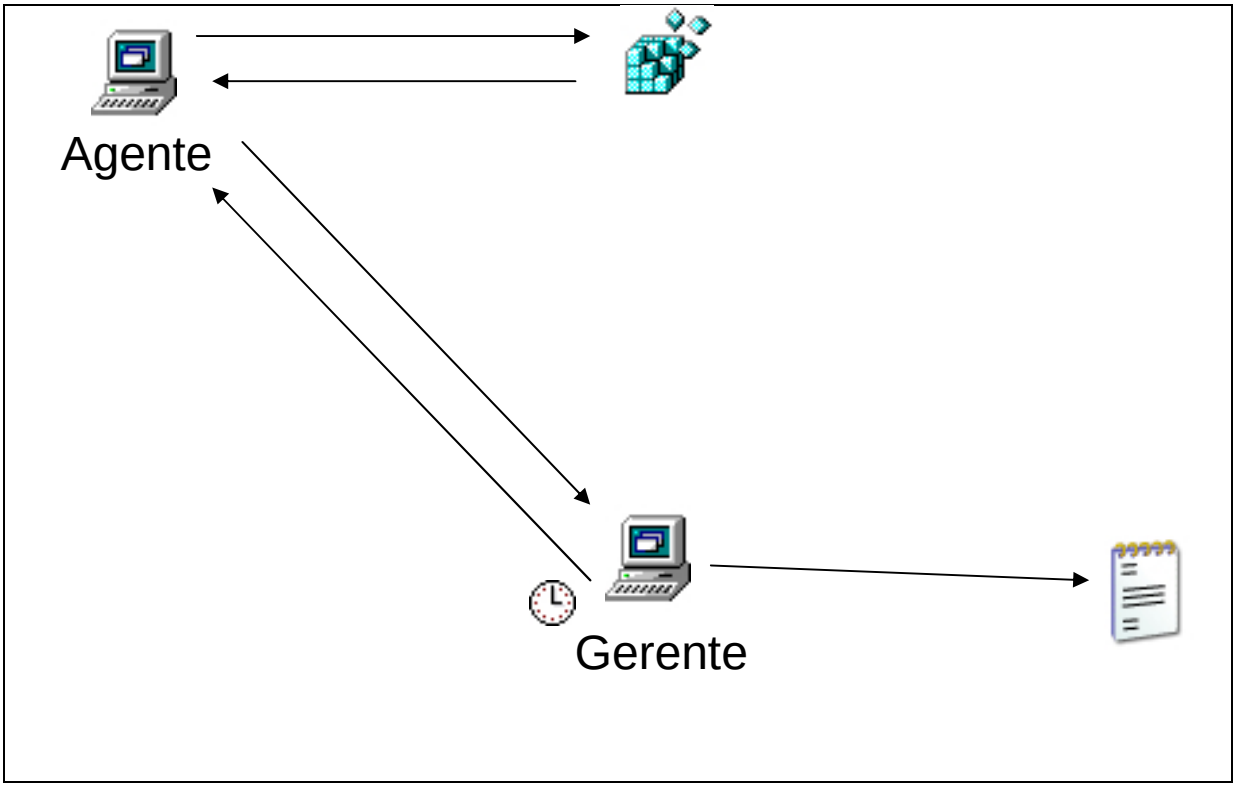


Figura 7 - Esquema de funcionamento entre agente e gerente

O Agente fica instanciado apresentando um ícone junto a barra de tarefas do Windows e clicando neste ícone o usuário local pode visualizar as informações de *hardware* do seu computador, conforme pode ser visto na Figura 8.

Figura 8 – Tela apresentada ao usuário da estação de rede pelo Agente.

Para a administração do patrimônio de rede, é utilizado o Gerente, que acumula o inventário de todas as estações de rede e as apresenta para o Gerente de Redes. A apresentação das informações é feita dividida por computadores, ou seja, é apresentado o inventário de cada estação da rede em separado, e para cada estação, são apresentados os seus componentes. Tudo isto é feito através de um menu *Tree-View* o que torna simples a sua operação e visualização, como pode ser visto na Figura 9. O Gerente não armazena persistentemente as informações porque já estão armazenas em cada uma das estações. Com isso o Gerenciador cada vez que é executado, solicita aos Agentes todas as informações.

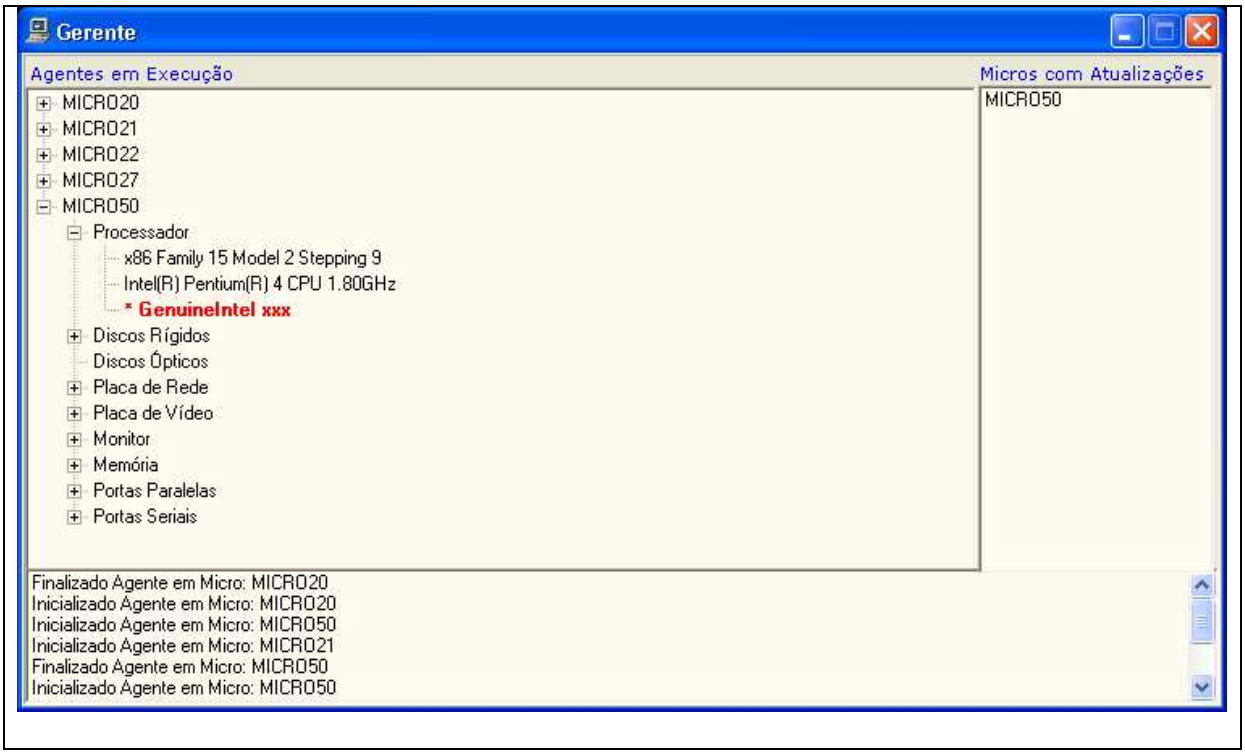


Figura 9 – Tela do Gerenciador, apresentando o inventário das estações.

Quando há alguma alteração no inventário de alguma estação da rede, é indicado pelo Gerente apresentando a alteração em destaque. Além da apresentação da alteração, é gerado um *log* onde são gravadas todas as alterações. Este *log* é mantido pelo gerenciador para consultas posteriores ou no caso de desligamento da rede.

Além do *log*, no momento em que o gerenciador recebe uma alteração de uma das estações de rede, o nome da estação alterada aparece numa lista separada da lista do inventário para que o Gerenciador de Rede tome alguma providência referente aquela alteração.

O Gerente também, num intervalo de 30 minutos, solicita a cada Agente da rede o dados de inventário atualizados, para que as informações centralizadas no Gerente, fiquem sempre atualizdas.

Para fazer o gerenciamento de inventário de rede com o protótipo implementado neste trabalho, basta distribuir o Agente nas estações, configurá-lo com o nome da máquina Gerente

e colher as informações através do Gerenciador, não dependendo assim de nenhuma aplicação externa instalada na rede. Isto torna o protótipo viável para gerenciar qualquer tipo de rede interna de qualquer tamanho, bastando apenas que ela seja uma rede Windows.

4.4 RESULTADOS E DISCUSSÃO

Vendo o funcionamento do protótipo, observou-se que todas as informações são apresentadas no mesmo momento em que os dados são levantados, ou seja, além de ter o inventário de todas as estações da rede na tela do gerenciador, pode-se também observar quais estações estão sendo gerenciadas naquele momento. Caso o uso do Agente se torne obrigatório nas estações, é possível verificar se uma determinada estação está ligada ou não e saber quando ela foi desliga, porque quando isto acontece, aquela estação sai da lista do gerenciador.

Todas as alterações de inventário das estações são facilmente visualizadas na tela do gerenciador, informando claramente as informações de cada componente instalado no computador.

Além de ver as estações em tempo real, pode-se fazer uma análise mais detalhada das alterações verificando o *log* gerado pelo gerenciador. Neste *log* ficam registradas as alterações e inicialização e finalização de cada estação da rede.

Comparando este protótipo ao do trabalho do Jacobowski (2004), pode-se observar que este adapta-se a qualquer tipo de rede, porque não depende de aplicações de terceiros instaladas nas estações, que no caso dele, necessita-se o *Windows Management Instrumentation* (WMI). Foi observado também que o protótipo dele envia e-mail ao Administrador de Rede, funcionalidade que não foi incorporada neste trabalho porque todas as informações são apresentadas, no próprio computador do Administrador da Rede.

5 CONCLUSÕES

Com os estudos e implementações que foram feitos neste trabalho, concluí que é bem simples gerenciar o inventário das estações de uma rede, porque existem diversos protocolos que auxiliam neste gerenciamento, bem como extrair as informações do computador e apresentá-las de maneira legível.

O Registro do Windows é um centro de informações muito importante para o computador, só que não é nada amigável, porque as informações ficam dispostas em posições diferentes dentro dele variando até mesmo de computador para computador com o mesmo sistema operacional. Informações muito básicas são de fácil entendimento, ao contrário de outras que ficam praticamente incompreensíveis.

Entretanto, conseguir extrair todas as informações desejadas do Registro para este trabalho. Pode-se considerar o Registro uma grande fonte de informação sobre o computador em relação a *hardware* e *software*, só que ele não segue uma seqüência lógica para guardar as informações, ou seja, parece que não é para entender mesmo.

Já a transmissão de dados utilizando o protocolo SMB atendeu perfeitamente todos os requisitos, porque o tempo de resposta é bastante satisfatório, bem como a confiabilidade de transmissão das informações, apesar de que foi necessário quebrar a mensagem para poder transmitir todas as informações necessárias entre o Agente e o Gerente.

Todos os requisitos foram supridos com o protótipo, que visa ter de modo simples o inventário das estações de uma rede, apresentando os componentes de cada estação de maneira que qualquer usuário possa entender tais informações.

Uma consulta aprimorada do *log* poderia ter sido implementada, mas o mais importante é que o *log* é gerado com todas as informações necessárias para uma análise futura. Esta consulta aprimorada do *log* pode ser feita como extensão deste trabalho.

5.1 EXTENSÕES

Este trabalho fez todo um estudo e implementou um protótipo de *software* que fez o inventário de *hardware* das estações de uma rede. Como extensão pode ser implementado e estudado como fazer o levantamento de *software* das estações da rede.

Além do inventário de *software* sugere-se também aprimorar o protótipo na parte gerencial, ou seja, criação de relatório baseado no *log*, envio de e-mail para o administrador de rede, comunicação através de mensagens entre Gerente e Agente e uma forma de ter certeza que o Agente nas estações não possa ser fechado sem o consentimento do Administrador de Redes.

REFERÊNCIAS BIBLIOGRÁFICAS

CANTU, Marco. **Dominando o Delphi 7:** a bíblia. Tradução Kátia Aparecida Roque. São Paulo: Pearson, 2003.

FOWLER, Martin; SCOTT, Kendall. **UML Essencial:** um breve guia para a linguagem padrão de modelagem de objetos. 2. ed. Tradução Vera Pezerico e Christian Thomas Price. Porto Alegre: 2000.

INFOWESTER [2004]. Disponível em http://www.infowester.com/tutregistrowin.php. Acesso em: 14 feb. 2005.

JACOBOWSKI, Rodrigo. **Protótipo de ferramenta para monitoriação de computadores utilizando o padrão de gerência WMI da Microsoft e a plataforma de desenvolvimento .NET**. 2004. 47 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

JAVVIN [2005]. Disponível em: http://www.javvin.com/protocolCMIP.html. Acesso em: 23 abr. 2005.

LINKTIONAY.COM [2001]. Disponível em: http://www.linktionary.com/c/cmip.html. Acesso em: 23 abr. 2005.

LOPES, Raquel V.; SAUVÉ, Jacques P.; Nicolletti, Pedro S. **Melhores práticas para Gerência de Redes de Computadores**. Rio de Janeiro: Campus, 2003.

MICROSOFT CORPORATION, INTEL CORPORATION, File Sharing Protocol: help. Version 2.0, 1988. Documento eletrônico disponibilizado no site SAMBA, Just What is SMB?.

SAMBA, **Just what is SMB?** [2002]. Disponível em: http://samba.anu.edu.au/cifs/docs/what-is-smb.html. Acesso em: 10 set. 2004.

SOUZA, Lindeberg Barros de. **Redes de computadores**: Dados, Voz e Imagem. São Paulo: Érica, 1999.