UNIVERSIDADE REGIONAL DE BLUMENAU

CENTRO DE CIÊNCIAS EXATAS E NATURAIS

CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

SISTEMA PARA MONITORAÇÃO DOS ACESSOS À INTERNET DE REDES CORPORATIVAS

ANDERSON RODRIGO RADTKE CARDOZO

BLUMENAU
2007

2007/2-03

ANDERSON RODRIGO RADTKE CARDOZO

SISTEMA PARA MONITORAÇÃO DOS ACESSOS À INTERNET DE REDES CORPORATIVAS

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
2007

2007/2-03

SISTEMA PARA MONITORAÇÃO DOS ACESSOS À INTERNET DE REDES CORPORATIVAS

Por

ANDERSON RODRIGO RADTKE CARDOZO

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

Presidente:	Prof. Francisco Adell Péricas– Orientador, FURB
Membro:	Prof. Sérgio Stringari – FURB
Membro:	Prof. Antonio Carlos Tavares – FURB

Blumenau, 27 de novembro de 2007

Dedico este trabalho ao meu irmão Andriei Ricardo Cardozo, que apesar de sua deficiência física e mental, provou que vale lutar pela vida mesmo estando limitado, e também a todos os que direta ou indiretamente ajudaram e colaboraram com os estudos e ensinamentos para a realização deste.

AGRADECIMENTOS

A Deus, criador supremo de todas as obras.

Aos meus pais, Wilson Cardozo e Roseli Maria Radtke Cardozo pela confiança, carinho e ensinamentos de toda uma vida.

A minha querida noiva Camila pelo apoio, paciência e confiança que depositou em mim em todas as horas dedicadas a este trabalho.

Ao meu filho Matheus, pelos sorrisos que me incentivaram nos momentos de dificuldades, angústia e desespero.

Ao meu avô Osmar Radtke, pela ajuda que ofereceu durante este período com os cuidados para o meu filho, proporcionando mais tempo para minha dedicação a este trabalho.

Aos meus amigos Leonardo Carbone, Fábio Schulter e Edmundo Nolas de Oliveira Junior, pelas ajudas, empurrões e cobranças.

A Elinton Oliveira Marçal e Everton Oliveira Marçal, diretores da Santa Catarina Informática, por terem ajudado financeiramente no meu curso e por disponibilizarem tempo e estrutura durante o desenvolvimento deste trabalho.

Ao meu orientador, Francisco Adell Péricas, por ter acreditado na conclusão deste trabalho sendo sempre prestativo e atencioso.

Não é o gênio que está um século à frente de seu tempo. É a humanidade que está cem anos atrás dele.

Robert Musil

RESUMO

Este trabalho apresenta um estudo sobre gerência de rede, resultando em duas ferramentas que auxiliam o gerenciamento do acesso à Internet para empresas que necessitam de controle dos recursos que disponibilizam para seus colaboradores. Uma destas ferramentas trabalha com o conceito de agentes que serão inicializados no momento em que o sistema operacional de cada computador cliente for carregado. Estes agentes verificam todo o acesso à Internet de cada estação identificando o domínio e registrando da forma mais adequada. Cada registro criado é armazenado no próprio cliente e será enviado para a ferramenta gerente em alguma determinada data e horário programados. Desta forma o tráfego em toda rede é poupado e torna o gerenciamento mais prático e rápido, podendo a ferramenta gerente solicitar os registros aos agentes a qualquer momento.

Palavras-chave: Gerência de redes. Tráfego. Placa de rede. Agente. Monitoração do conteúdo. Registro de acessos.

ABSTRACT

This work presents a study on network management, resulting in two tools that assist the management of access to the Internet for companies who need a control of resources that they make use for their collaborators. One of these tools works with the concept of agents who will initiate at the moment when the operational system of each client computer is loaded. These agents verify the access to the Internet of each station identifying the domain and registering of more adjusted form. Each register bred will be stored in the proper customer and will be sent for the controlling tool in some determined programmed day and schedule. Of this form, the traffic in all network is saved and becomes the fast management most practical, being able the controlling tool to request the registers to the agents at any moment.

Key-words: Nets manages. Traffic. Net board. Agent. Monitoring content. Register of accesses.

LISTA DE ILUSTRAÇÕES

LISTA DE SIGLAS

AFS – *Andrew File System*

BOOTP – *BOOTstrap Protocol*

DHCP – *Dynamic Host Configuration Protocol*

DNMP – *Delayed-NonMatching-to-Place*

DNS – *Domain Name System*

FID – *File IDentifier*

FTP – *File Transfer Protocol*

HTTP – *HyperText Transfer Protocol*

HTTPS – *HyperText Transfer Protocol Secure*

ID – *IDentifier*

IP – *Internet Protocol*

ISO – *International Organization for Standardization*

LAN – *Local Area Network*

NFS – *Network FileSystem*

NNTP – *Network News Transfer Protocol*

P2P – *Peer-to-Peer*

PID – *Process IDentifier*

SMB – *Session Message Block*

SNMP – *Simple Network Management Protocol*

SSH – *Secure SHell*

TCP – *Transmission Control Protocol*

TID – *Tree IDentifier*

UML – *Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO

Hoje em dia, com o crescimento das redes de computadores e com o vasto e diversificado conteúdo disponibilizado pela Internet, a concentração dos funcionários nas atividades que devem ser desempenhadas em suas respectivas funções nas empresas acaba ficando comprometida e, muitas vezes, uma solução acaba prejudicando muito mais do que solucionando, pois consome muito dos recursos que a empresa dispõe e gera novos problemas.

A facilidade do acesso à Internet sem restrições ou controle, muitas vezes leva o funcionário a responder ao correio eletrônico particular, conversar sobre assuntos que não tem relação com o seu trabalho por meio de mensagens instantâneas, verificar o seu perfil em portais de relacionamentos, jogar, ver fotos e vídeos inadequados em páginas que oferecem este tipo de conteúdo, entre outras coisas. Tudo isto interfere diretamente na produtividade e conseqüentemente reduz a eficiência da empresa, oferecendo também grandes riscos de contaminação da rede por programas nocivos como vírus e “cavalos de tróia”. O fato de consumir recursos utilizando a Internet de uma forma inadequada pode gerar custos adicionais para a empresa e atrapalhar outros setores e funcionários.

Existem diversas formas de monitorar, controlar e bloquear o acesso das estações da rede à Internet utilizando ferramentas que geralmente são instaladas em um servidor, e que por este motivo consomem muitos recursos, pois precisam tratar todas as requisições sozinhas já que centralizam neste servidor todas as atividades de processamento.

Baseado neste problema de gerenciar o acesso à Internet das estações de uma rede de uma forma não centralizada foi desenvolvido neste trabalho à utilização de agentes para distribuir o processamento e disponibilizar de uma forma clara, rápida e objetiva para o administrador da rede a identificação do domínio de todos os acessos à Internet realizados por todas as estações de trabalho que estão sendo monitoradas.

As informações estarão disponíveis ao administrador da rede de uma forma agendada ou requisitada. Cada agente registra em um arquivo no disco rígido da estação onde estiver ativo todos os acessos à Internet realizados por esta mesma estação. Uma ferramenta gerente vai solicitar estas informações aos agentes em algum determinado momento e com isto poderá apresentar as mesmas de uma forma centralizada para o administrador da rede.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho foi desenvolver uma ferramenta agente e uma ferramenta gerente para a monitoração dos acessos à Internet em um ambiente corporativo.

Os objetivos específicos do trabalho são:

a) disponibilizar nas estações da rede os agentes para registrar as informações dos acessos à Internet de cada estação, identificando estes domínios e registrando os mesmos da forma mais adequada em um arquivo no disco rígido;
b) disponibilizar uma ferramenta gerente para centralizar as informações colhidas dos agentes de cada estação da rede;
c) disponibilizar a qualquer momento por meio da ferramenta gerente a consulta dos acessos à Internet realizados pelas estações da rede. Isso deve ser possível através de um agendamento ou de uma requisição feita pelo administrador da rede.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está dividido em seis capítulos.

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.

O capítulo 3 apresenta o protocolo *Session Message Block* (SMB) com todas as suas propriedades e funcionalidades.

No capítulo 4 é apresenta a aplicação de captura de tráfego de rede Ntop.

O capítulo 5 descreve como foram desenvolvidas as ferramentas agente e gerente, apresentando os seus requisitos, a especificação através dos diagramas de casos de uso, diagramas de classe e diagramas de atividades. Também são apresentadas as suas funcionalidades e todo o esquema de funcionamento.

No capítulo 6 são apresentas as conclusões sobre o trabalho e as sugestões para extensões e trabalhos futuros.

2 GERÊNCIA DE REDES

Segundo Tanenbaum (1994, p. 3), muitas organizações já têm um número substancial de computadores em operação, muitas vezes situados em pontos distantes entre si. Por exemplo, uma empresa com muitas fábricas pode ter um computador em cada local para acompanhar o inventário, monitorar a produtividade e emitir a folha de pagamento local. Inicialmente, cada um desses computadores talvez tenha trabalhado de modo isolado mas, em um certo momento, a gerência pode ter decidido conectá-los a fim de poder extrair e correlacionar informações acerca de toda empresa.

Ainda segundo Tanenbaum (1994, p. 4), colocada de forma ligeiramente mais geral, a questão aqui é o compartilhamento de recursos, e o objetivo é fazer com que todos os programas, dados e equipamentos estejam disponíveis para qualquer um na rede, independentemente da localização física do recurso e do usuário. Em outras palavras, o simples fato de um usuário por acaso estar a quilômetros de distância dos seus dados não deveria impedi-lo de usar os dados como se fossem locais. O balanceamento da carga é um outro aspecto do compartilhamento de recursos. Esse objetivo pode ser resumido dizendo-se que é uma tentativa de acabar com a tirania da geografia.

Péricas (2003, p. 121-122) afirma que o gerenciamento de redes é dividido em gerência de falhas, gerência de configuração, gerência de contabilização, gerência de desempenho e gerência de segurança.

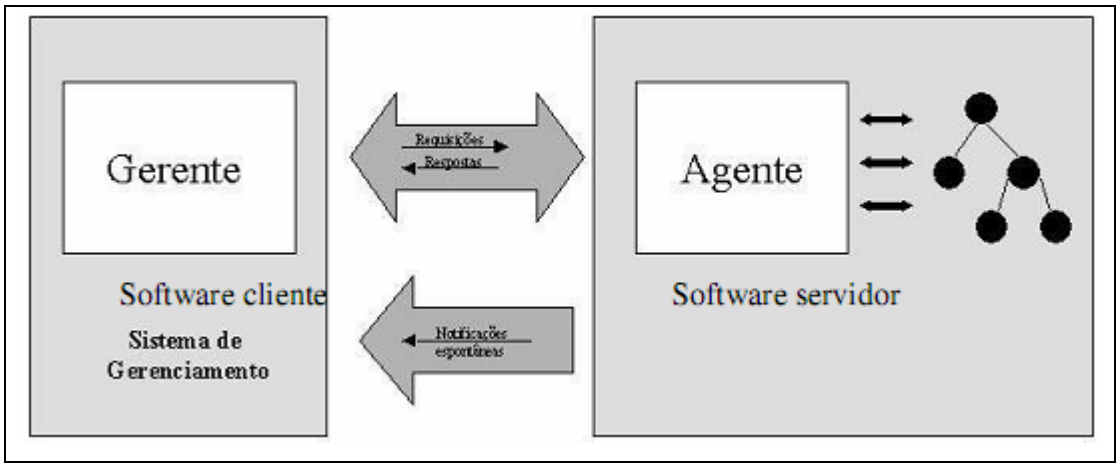
Conforme Péricas (2003, p. 122), gerência de segurança permite prevenir e detectar o uso impróprio ou não autorizado de recursos de uma rede, assim como administrar a sua segurança. Outra preocupação da segurança volta-se para as pessoas que tentam ter acesso a serviços remotos, para aos quais elas não estão autorizadas a usar. Uma rede de computadores é essencialmente vulnerável a acessos não autorizados, pois permite que qualquer equipamento analise o conteúdo de mensagens que estejam trafegando pela rede mesmo que não sejam direcionadas ao equipamento. Além disso, qualquer equipamento conectado a uma rede pode criar, alterar ou extrair mensagens mascarando a sua operação através de endereços fictícios.

Oliveira (2005, p. 14) afirma que o objetivo de Gerência de Redes é monitorar e controlar os elementos da rede, assegurando a qualidade do serviço. Para realizar esta tarefa, os gerentes de rede são geralmente auxiliados por um sistema de gerência de redes que por sua vez pode ser definido como um conjunto de ferramentas integradas para a monitoração e

controle. Este sistema pode apresentar uma interface única, com informações sobre a rede, e pode oferecer também um pacote de comandos que são utilizados para executar quase todas as tarefas de gerência de rede.

Conforme Péricas (2003, p. 125), a infra-estrutura de gerenciamento especificada pela *International Organization for Standardization* (ISO) define que gerente é uma aplicação de gerência (software cliente) que faz as requisições de operações, recebe notificações, enquanto que o agente (software servidor) é que recebe e processa estas operações e envia as respostas e emite as notificações. Na Figura 1 pode-se ver um exemplo de comunicação entre as aplicações gerente e agente.

Conforme Lopes, Sauvé e Nicolletti (2003, p. 4), 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.



Fonte: Péricas (2003, p. 125).

Figura 1 – Comunicação entre entidades de gerência

Segundo Oliveira (2005, p. 14-15), os elementos gerenciados possuem um programa especial chamado agente e este 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 de Internet, servidores de correio eletrônico 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.

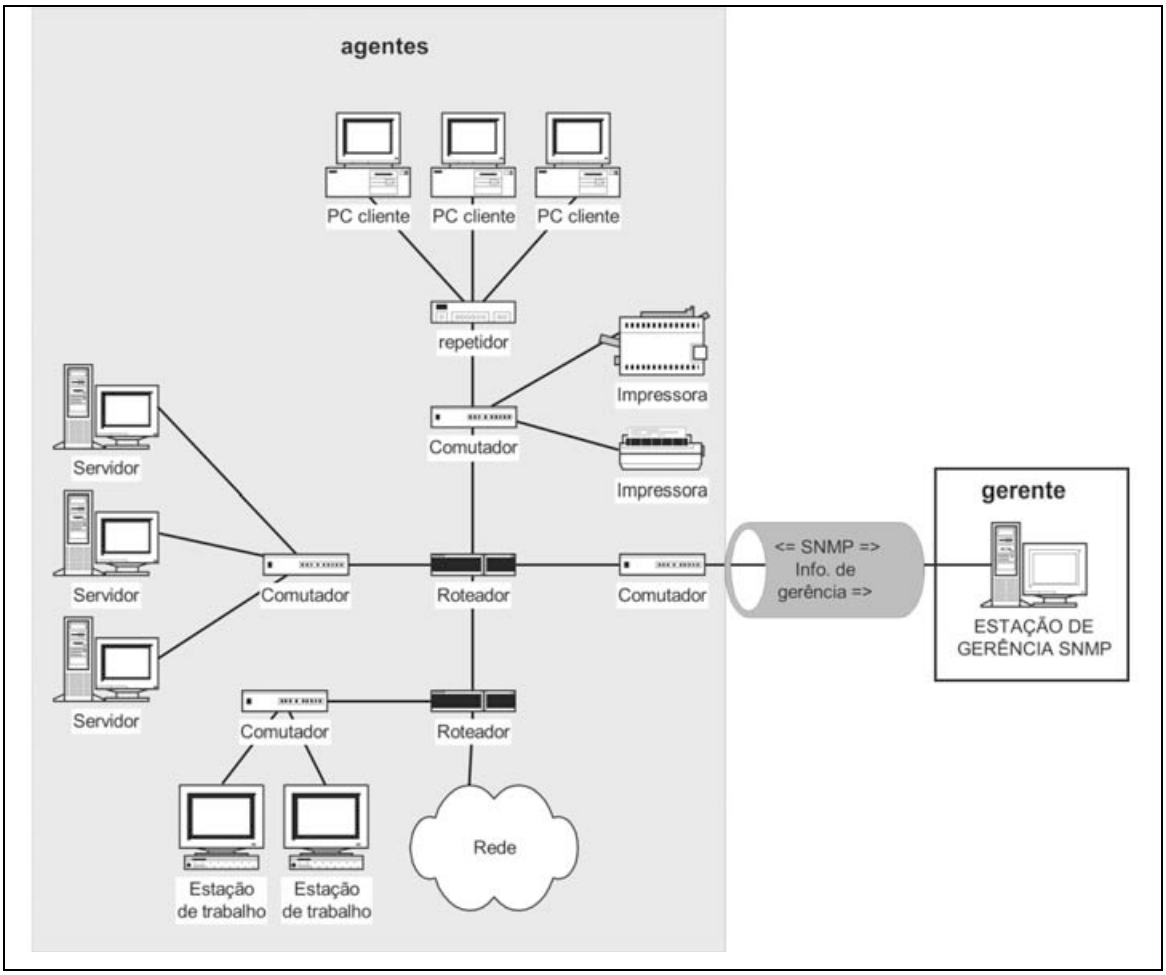
Lopes, Sauvé e Nicolletti (2003, p. 4-5) afirmam que o software presente na estação de gerência que conversa diretamente com os agentes nos elementos gerenciados é chamado de gerente. A estação de gerência pode obter informação de gerência presente nos elementos gerenciados através de uma sondagem regular dos agentes ou até mesmo recebendo informações enviadas 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.

Conforme Oliveira (2005, p. 15), 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.

Ainda segundo Oliveira (2005, p. 15), o gerente e os agentes trocam informação de gerência usando um protocolo de gerência. O protocolo inclui operações de monitoramento como a leitura de informação de gerência e operações de controle como a alteração de informação de gerência presente no elemento gerenciado.

Segundo Lopes, Sauvé e Nicolletti (2003, p. 5-6), 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.

Na Figura 2 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 implementado pelo software gerenciador que no caso desta figura foi o *Simple Network Management Protocol* (SNMP).



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

Figura 2 – Elementos de uma arquitetura geral de solução de gerência

Oliveira (2005, p. 18), afirma que 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.

Conforme Lopes, Sauvé e Nicolletti (2003, p. 6), com os analisadores de protocolo 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.

Segundo Péricas (2003, p. 125), uma aplicação de gerência pode fazer ao mesmo tempo o papel de gerente para uma aplicação e o papel de agente para uma outra aplicação gerente.

2.1 O PAPEL DO GERENTE DE REDES

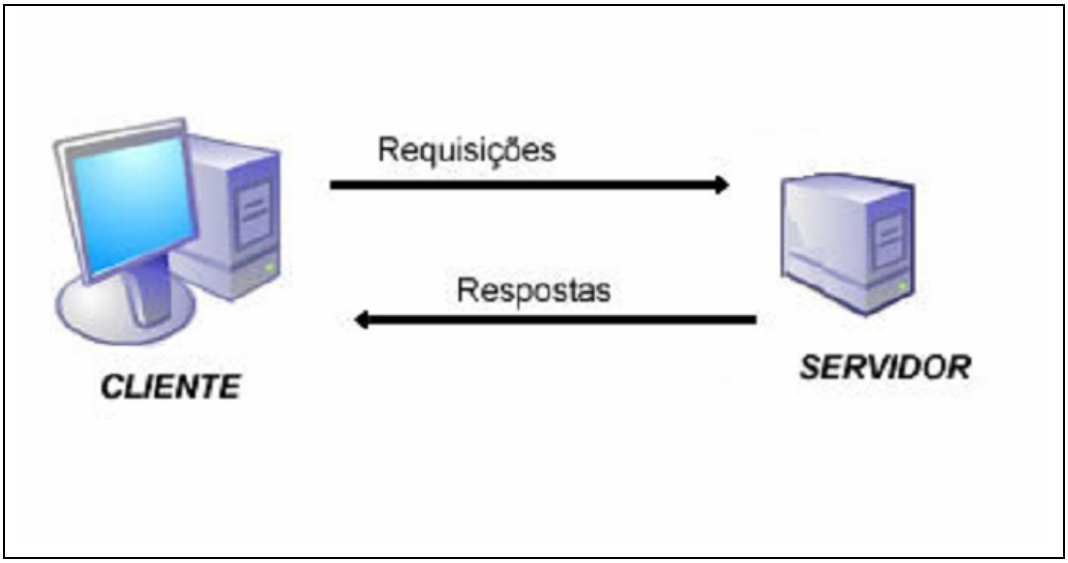
Lopes, Sauvé e Nicolletti (2003, p. 7) afirmam que o gerente da equipe de gerência de rede não é, necessariamente, um técnico em redes. O gerente tem um certo conhecimento em redes, mas não no nível do suporte técnico. Dentre as atividades deste gerente encontra-se: avaliar o desempenho da sua equipe 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.

Segundo Kurose e Ross (2005, p.572), um administrador de rede pode monitorar padrões de tráfego entre fontes e destinos e notar, por exemplo, que, comutando servidores segmentados de *Local Area Network* (LAN), o total de tráfego que passa por várias LANs poderia ser reduzido de maneira significativa. Imagine a felicidade geral (especialmente da alta administração) com um desempenho melhor sem o custo de novos equipamentos. De modo similar, monitorando a utilização de um enlace, um administrador de rede pode determinar que um segmento de LAN ou o enlace com o mundo externo está sobrecarregado e que um enlace de maior largura de banda deve ser providenciado (ainda que com um custo mais alto). Ele também poderia ser alertado automaticamente quando o nível de congestionamento de um enlace ultrapassasse determinado limite, para providenciar um enlace de maior largura de banda antes que o congestionamento se tornasse sério.

3 PROTOCOLO SMB

O *Session Message Block* (SMB) é um protocolo da IBM para compartilhar arquivos, impressoras, portas seriais e para comunicação entre computadores. O SMB pode ser usado com o protocolo *Transmission Control Protocol* (TCP) na Internet ou com outros protocolos de rede. Uma vez estabelecida uma conexão, os clientes podem enviar comandos SMB para o servidor para acessar pastas compartilhadas, ler e escrever em arquivos entre outras funcionalidades (WOOD, 2000).

Na Figura 3 pode-se observar como o SMB trabalha.



Fonte: Oliveira (2005, p. 30).

Figura 3 – Forma de trabalho do protocolo SMB

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 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 que deve ser utilizada em todos os próximos acessos ao servidor (SHARPE, 2002).

Oliveira (2005, p. 31-32) afirma que 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. 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 *IDentifiers* (IDs) conhecida como *Tree IDentifier* (TID). Uma 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 *Process IDentifier* (PID). Um PID identifica excepcionalmente um processo do cliente dentro de um ambiente;
d) *file environment*: é representado por um *file handle* denominado *File IDentifier* (FID). Um FID identifica um arquivo aberto que é único dentro de um ambiente.

Ainda segundo Oliveira (2005, p. 32), 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.

Conforme Oliveira (2005, p. 32), os sistemas podem usar este protocolo para obter ou fornecer serviços de arquivos remotos 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.

Ainda conforme Oliveira (2005, p. 32), 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 o 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, relacionando em cada coluna o tipo do dado, o campo e o valor.

```
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
```

Fonte: Oliveira (2005, p. 33).

Quadro 1 – Estrutura de mensagem SMB utilizando a linguagem C

Oliveira (2005, p. 33-36) afirma que 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 usam compartilhamento de arquivos do protocolo e conterão não somente sistemas multi-usuários com modelos baseados no servidor de proteção, mas os sistemas mono-usuário que não têm nenhum conceito dos UIDs ou das permissões. Uma vez que estas máquinas são conectadas à rede, estão em um ambiente multi-usuário e necessitam de um método do controle de aceso. Primeiro, as máquinas desprotegidas necessitam controlar os acessos aos seus arquivos por outro. Este protocolo define um mecanismo que habilita o software de rede a fornecer a proteção onde falta o sistema operacional e suporte à proteção do servidor fornecido pelo sistema operacional;
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 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 todos 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 retomada. 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 nenhuma 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 e as mensagens de negociação subseqüentes 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 algoritmo 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 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 connect* deve ser emitida para todas as *subtrees* alcançadas, mesmo se contém uma senha nula.

4 APLICAÇÃO NTOP

Ntop é uma aplicação de tráfego de rede que mostra a utilização da rede, similar ao que o popular comando `top` do Unix faz. Ntop é baseado na biblioteca `libpcap` e foi escrito de uma forma portável com o objetivo de rodar em qualquer plataforma Unix e Win32. Usuários Ntop podem utilizar o navegador da Internet para buscar informações do tráfego e da situação da rede (NTOP.ORG, 2006).

Segundo Zanoni (2007), o Ntop é um programa que monitora passivamente uma rede, coletando dados sobre os protocolos e sobre os *hosts* da rede. Características e funcionalidades:

a) analisa os pacotes que trafegam na rede;
b) lista e ordena o tráfego de rede de acordo com vários protocolos;
c) exibe estatísticas de tráfego;
d) armazena estatísticas de forma permanentemente em bancos de dados;
e) identifica passivamente várias informações sobre os *hosts* da rede, incluindo o sistema operacional executado e endereço de e-mail do usuário da estação;
f) exibe a distribuição do tráfego *Internet Protocol* (IP) entre vários protocolos da camada de aplicação;
g) decodifica vários protocolos da camada de aplicação, inclusive os encontrados em softwares do tipo *Peer-to-Peer* (P2P);
h) atua como coletor de fluxos gerados por roteadores e *switches* através da tecnologia NETFLOW;
i) possui um *webserver* integrado que permite consultas às informações através de um navegador.

Ainda segundo Zanoni (2007), após instalar e configurar o Ntop, existem vários parâmetros definidos por padrão que podem ser alterados conforme relacionado no Quadro 2.

Parâmetro	**Descrição**
-A	Define ou altera a senha do usuário administrador.
-a <arquivo>	Habilita *logs* no servidor *web*: por padrão, o Ntop não gera *logs* das requisições que seu servidor *web* recebe. Para habilitar, usa-se esta opção acompanhada pelo nome do arquivo onde serão armazenados os *logs*.
-b	Desabilita decodificadores de protocolos: os decodificadores de protocolos examinam e coletam informações sobre vários tipos de protocolos das pilhas NetBIOS, Netware e TCP/IP.
-d	Inicia o Ntop em modo deamon (*background*): este parâmetro é sempre incluído pelo *script* de inicialização.
-i <nome>	Nome das interfaces que serão monitoradas.
-K	Habilita o modo depuração: útil para diagnosticar problemas do serviço.
-M	Não une o tráfego das interfaces de rede: por padrão o Ntop une os dados coletados de todas as interfaces em um único conjunto de contadores. Em uma rede pequena local isto é interessante, pois gera uma imagem melhor da rede como um todo.
-m	Redes que serão consideradas locais.
-n	Não resolve endereços de nomes.
-P <caminho>	Caminho do diretório que contém o banco de dados do programa.
-p <arquivo>	Substitui os protocolos que o Ntop analisa por padrão pelos contidos no arquivo.
-u <usuário>	Usuário que executará o processo Ntop: por padrão é "ntop".
-W <porta>	Porta do servidor *web* (HTTPS): por padrão o servidor *web* não responde HTTPS, é necessário especificar a porta para habilitar o suporte. Um endereço pode ser especificado também no formato "endereço:porta".
-w <porta>	Porta do servidor web (HTTP): por padrão o webserver escuta na porta 3000. Um endereço pode ser especificado também no formato "endereço:porta".

Fonte: Zanoni (2007).

Quadro 2 – Lista de parâmetros mais úteis do Ntop

Zanoni (2007) afirma que por padrão o Ntop monitora apenas um conjunto reduzido de protocolos, listados no Quadro 3. Este conjunto de protocolos pode ser substituído pelo administrador através do parâmetro –p, que recebe o nome de um arquivo contendo os protocolos a monitorar como argumento.

Protocolo	**Portas**
FTP	20 21
HTTP	80 443 3128
DNS	53
Telnet	23 513
NBios-IP	137 138 139
Mail	25 109 110
DHCP/BOOTP	67 68
DNMP	161 162
NNTP	119
NFS/AFS	2049 7000-7009
X11	6000-6010
SSH	22
Kazaa	1214
WinMX	6699 7730
eDonkey	4661-4665
Bit Torrent	6881-6999 6969
Messenger	1863 5000 5001

Fonte: Zanoni (2007).

Quadro 3 – Protocolos padrões monitorados pelo Ntop

O Ntop é distribuído sob uma licença pública geral. Os formatos disponíveis do Ntop são o formato fonte, para compilar virtualmente em toda plataforma Unix ou Windows e o formato de aplicação binária, para instalação. A versão binária para Windows possui uma limitação de captura de mil pacotes. Esta limitação pode ser retirada caso recompile o formato fonte. O formato de aplicação binária para Unix não possui limitação (NTOP.ORG, 2006).

5 DESENVOLVIMENTO DO TRABALHO

Neste capítulo serão vistas, em detalhes, as propriedades e funcionalidades das ferramentas agente e gerente que foram implementadas. São apresentados os requisitos e toda a sua especificação através da linguagem de modelagem *Unified Modeling Language* (UML) e do projeto orientado a objetos. São também apresentados os detalhes de suas implementações e testes realizados em uma rede corporativa.

5.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Os requisitos das ferramentas agente e gerente estão separados em funcionais e não-funcionais.

Os requisitos funcionais são:

a) iniciar o aplicativo agente dentro da estação da rede toda vez que esta for inicializada;
b) registrar da forma mais adequada em um arquivo, os acessos à Internet feitos na estação da rede;
c) permitir que o administrador da ferramenta gerente agende um determinado dia da semana e horário para automaticamente solicitar os dados dos arquivos com os registros dos acessos à Internet de cada estação;
d) possibilitar que o administrador da rede solicite o recolhimento dos dados dos arquivos com os registros dos acessos à Internet de uma estação específica ou de todas a qualquer momento;
e) permitir a consulta das informações dos acessos à Internet de cada estação.

Os requisitos não-funcionais são:

a) serem implementados utilizando o ambiente de programação Delphi 7 da Borland;
b) serem compatíveis com os sistemas operacionais Windows 98, 98SE, 2000, ME e XP.

5.2 ESPECIFICAÇÃO

Para especificação das ferramentas agente e gerente, foi utilizado a UML, usando os diagramas de casos de uso, de classes e de atividades. Para criação destes diagramas foi utilizado o software Enterprise Architect da Sparx Systems.

5.2.1 Diagrama de casos de uso

Na Figura 4 vê-se o diagrama de casos de uso das ferramentas, que especifica como o agente e gerente interagem para compartilhar o arquivo de registros dos acessos à Internet das estações e apresentar ao administrador da rede.

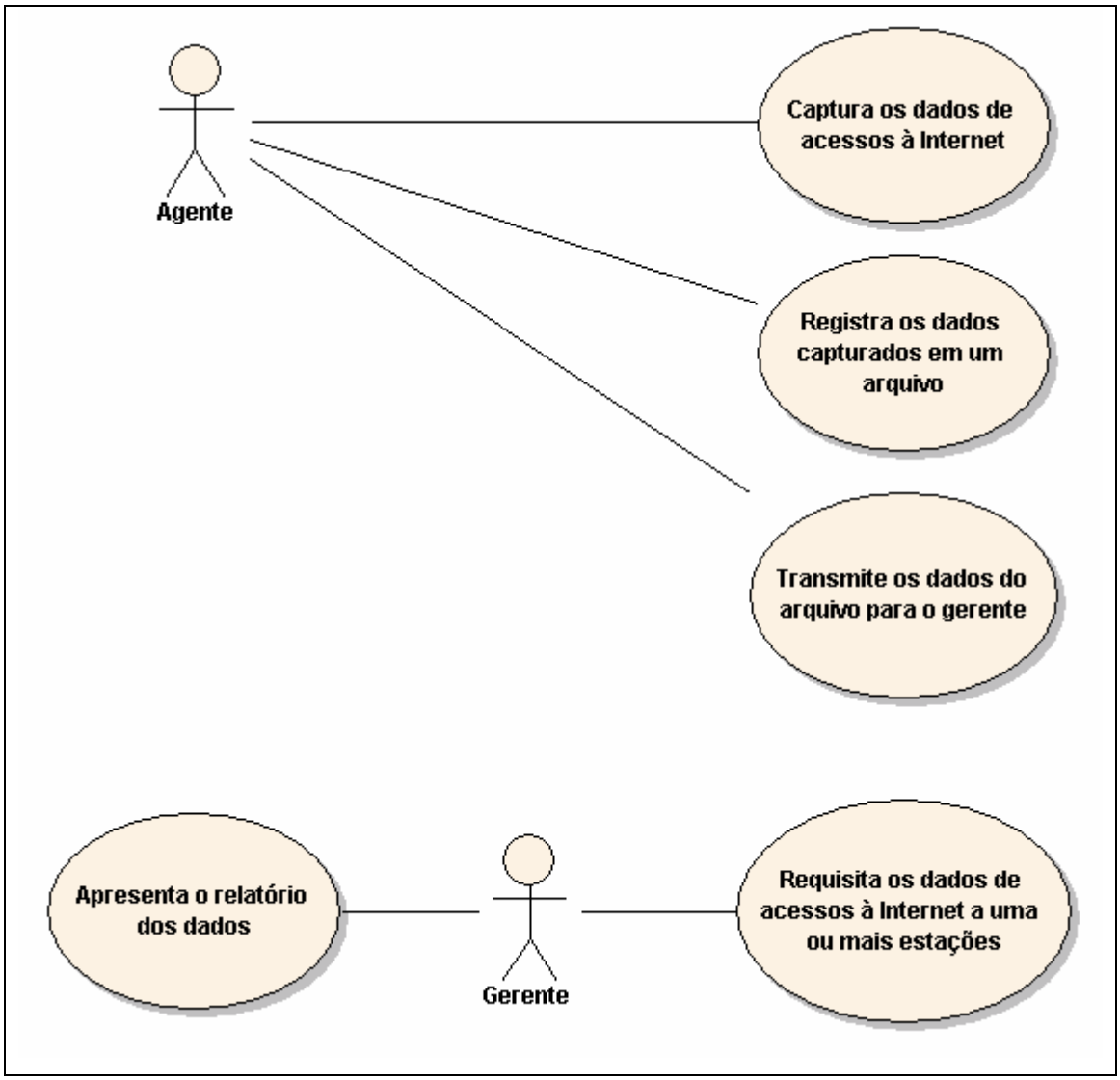


Figura 4 – Diagrama de casos de uso

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

c) captura os dados de acessos à Internet: neste caso de uso o agente captura os dados fornecidos pelo aplicativo Ntop que está monitorando os acessos à Internet realizados pela estação de rede;
d) registra os dados capturados em um arquivo: este caso de uso serve para que o agente possa registrar em um arquivo os acessos à Internet feitos na estação de rede capturados no caso de uso anterior;
e) transmite os dados do arquivo para o gerente: neste caso de uso o agente envia os dados do arquivo de acessos à Internet encontrado na estação para o gerente quando for requisitado pelo mesmo;
f) requisita os dados de acessos à Internet a uma ou mais estações: este caso de uso serve para o gerente solicitar a coleta dos dados de acessos à Internet de todas estações em algum determinado dia da semana e horário agendado pelo administrador ou de uma estação específica;
g) apresenta o relatório dos dados: neste caso de uso o administrador pode consultar as informações dos acessos à Internet de cada estação monitorada.

5.2.2 Diagrama de classes

O diagrama de classes descreve como serão divididas as mesmas na implementação, apresentando em detalhes todos os atributos, métodos e relacionamentos de cada, conforme a Figura 5.

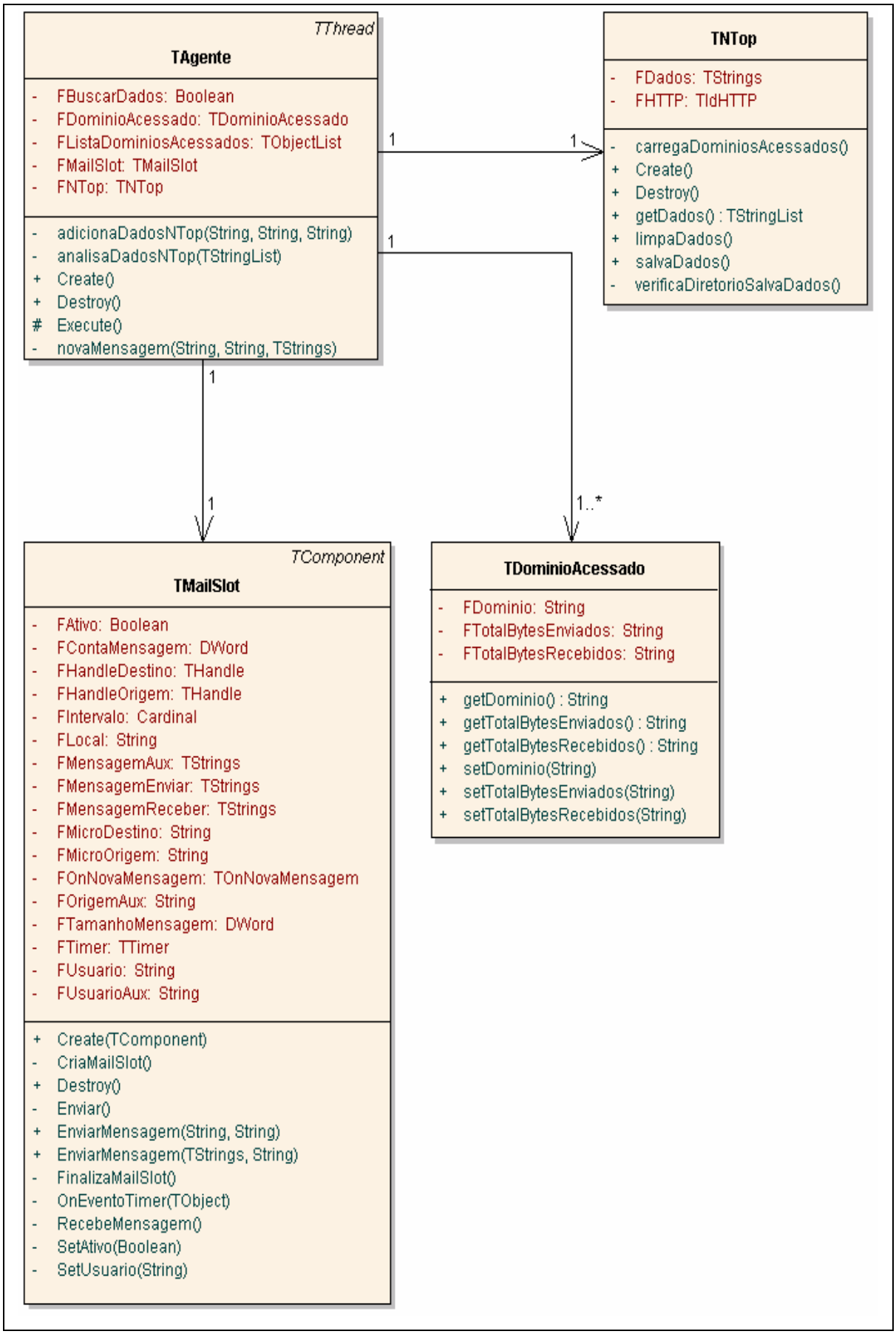


Figura 5 – Diagrama de classes

Cada uma destas quatro classes tem a seguinte função:

a) `TAgente`: esta é a principal classe da ferramenta agente sendo ela uma *thead* responsável por instanciar cada uma das demais classes;
b) `TNTop`: esta classe serve para coletar os dados de acessos à Internet da aplicação Ntop;
c) `TMailSlot`: esta classe é um componente da classe `TAgente` e tem o objetivo de enviar e receber dados, ou seja, estabelece a comunicação entre a ferramenta agente e a ferramenta gerente utilizando o protocolo SMB;
d) `TDominioAcessado`: esta classe serve para armazenar os dados coletados pela classe `TNTop`.

Abordando o relacionamento entre estas quatro classes, a classe `TAgente` instancia a classe `TNTop` que periodicamente coleta os dados de acessos à Internet. Para cada domínio acessado a classe `TAgente` instancia a classe `TDominioAcessado` que armazena estas informações. Em algum determinado momento a classe `TAgente` recebe da classe `TMailSlot` a informação da requisição pela ferramenta gerente dos dados de acessos à Internet e envia estes dados para que a classe `TMailSlot` repasse para ferramenta gerente.

5.2.3 Diagrama de atividades

O diagrama de atividades mostra como são os procedimentos da ferramenta agente para que os dados de acessos à Internet realizados pelas estações cheguem à ferramenta gerente, que pode ser visto na Figura 6.

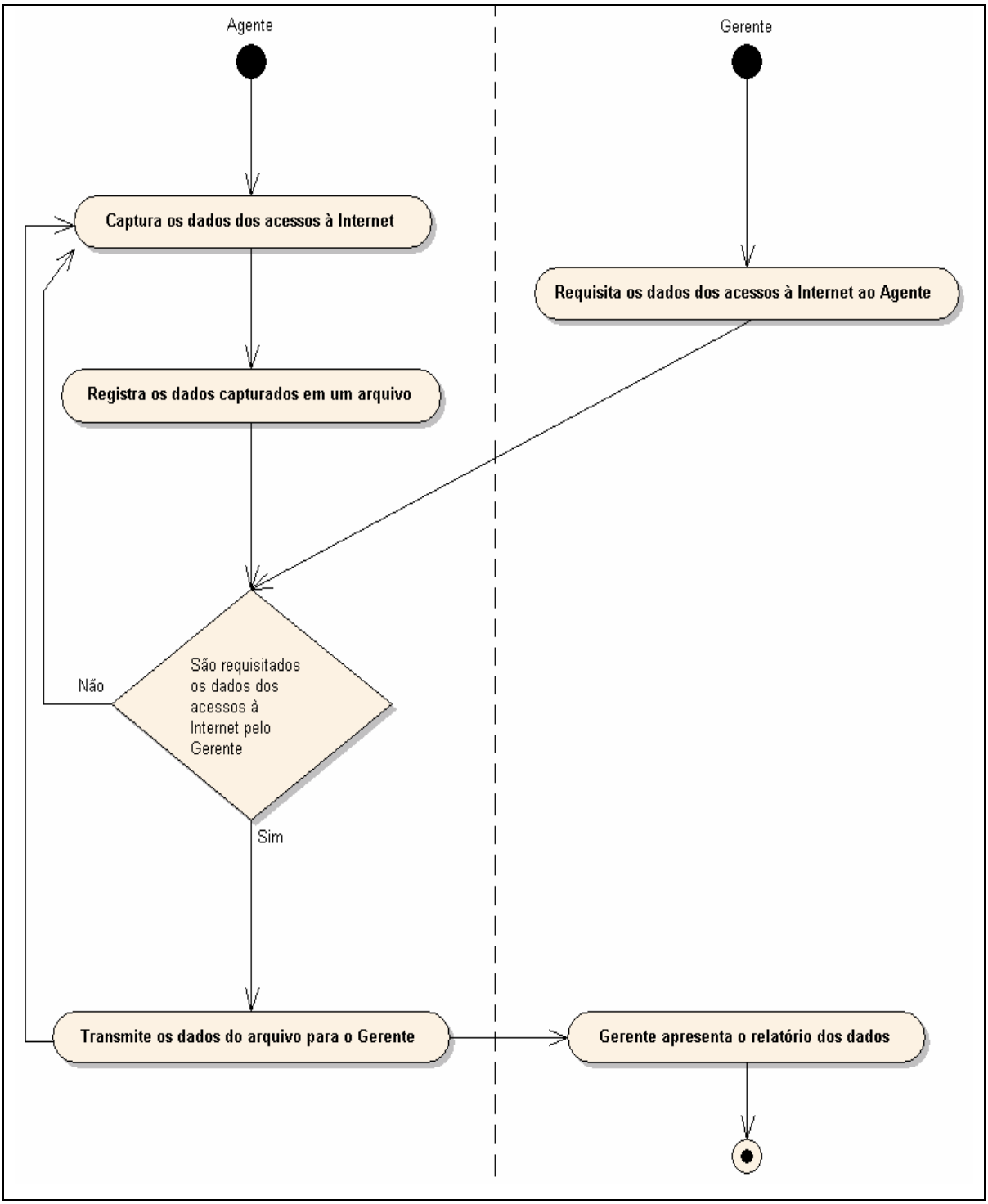


Figura 6 – Diagrama de atividades

5.3 IMPLEMENTAÇÃO

Esta seção contém o detalhamento sobre a implementação das ferramentas agente e gerente. O tópico inicial identifica as técnicas e ferramentas utilizadas. O tópico seguinte apresenta a operacionalidade da implementação.

5.3.1 Técnicas e ferramentas utilizadas

Para implementação das ferramentas foi utilizado o ambiente de programação Delphi 7 da Borland. O Delphi apresenta muitas facilidades, além de ser totalmente compatível com todas as versões do sistema operacional Windows.

Foi utilizada a aplicação Ntop, e quando a mesma é iniciada, é criado um servidor local e todos os dados monitorados pela mesma podem ser acessados através de qualquer navegador da Internet.

O sistema de arquivos do aplicativo Ntop para apresentação dos dados é formado por páginas em HTML, além de armazenar determinadas informações em um banco de dados protegido. Para este trabalho não foi necessário utilizar o banco de dados do referido aplicativo pois apenas os dados contidos nos arquivos HTML já supriram a necessidade.

A captura dos dados monitorados pelo aplicativo Ntop é realizada pela ferramenta agente que guarda todo conteúdo da página em HTML gerada pelo Ntop responsável somente pelos domínios acessados. Após isto o agente procura por um conjunto de identificadores que se refere às páginas acessadas, repetindo este processo até que localize todos identificadores e conseqüentemente encontre todos os domínios e demais dados necessários dos acessos à Internet da estação. Este processo é repetido periodicamente e também no momento que a ferramenta gerente faz solicitações. Após o processo citado anteriormente, o agente armazena em um arquivo todas estas informações para posteriormente enviar este conteúdo para a ferramenta gerente quando a mesma solicitar.

No Quadro 4 pode ser visto o procedimento `analisaDadosNTop` da classe `TAgente` que é responsável na ferramenta agente pelas ações descritas no parágrafo anterior.

```
procedure TAgente.analisaDadosNTop(Dados: TStringList);
var
   wStrAux, wDominio, wBytesRec, wBytesEnv : String;
   wPos, wCont : Integer;
begin
   try
      try
         for wCont := 147 to Dados.Count - 1 do
            begin
               wStrAux := Dados.Strings[wCont];
               wPos := Pos(cDominio,wStrAux);
               if wPos <> 0 then
                  begin
                     // Localiza Domínio
                     wDominio := copy(wStrAux,wPos,length(wStrAux));
                     wDominio := copy(wDominio,Pos(cIgual,wDominio)+1,
                                      length(wDominio));
                     wDominio := copy(wDominio,0,Pos(cFechaTag,
                                      wDominio)-1);
                     // Localiza Bytes Enviados
                     wPos      := Pos(cALIGNRIGHT,wStrAux);
                     wBytesEnv := copy(wStrAux,wPos,length(wStrAux));
                     wBytesEnv := copy(wBytesEnv,Pos(cFechaTag,
                                       wBytesEnv)+1,length(wBytesEnv));
                     wBytesEnv := copy(wBytesEnv,0,Pos(cAbreTag,
                                       wBytesEnv)-1);
                     wPos      := Pos(cEComercial,wBytesEnv);
                     if wPos <> 0 then
                        wBytesEnv := copy(wBytesEnv,0,wPos-1);
                        if Pos('.',wBytesEnv) <> 0 then
                           wBytesEnv := wBytesEnv + ' KBytes'
                        else
                           wBytesEnv := wBytesEnv + ' Bytes';
                     // Localiza Bytes Recebidos
                     wPos      := Pos(cPercentual,wStrAux) + 1;
                     wBytesRec := copy(wStrAux,wPos,length(wStrAux));
                     wPos      := Pos(cALIGNRIGHT,wBytesRec);
                     wBytesRec := copy(wBytesRec,wPos,length
                                       (wBytesRec));
                     wBytesRec := copy(wBytesRec,Pos(cFechaTag,
                                       wBytesRec)+1,length(wBytesRec));
                     wBytesRec := copy(wBytesRec,0,Pos(cAbreTag,
                                       wBytesRec)-1);
                     wPos      := Pos(cEComercial,wBytesRec);
                     if wPos <> 0 then
                        wBytesRec := copy(wBytesRec,0,wPos-1);
                     if Pos('.',wBytesRec) <> 0 then
                        wBytesRec := wBytesRec + ' KBytes'
                     else
                        wBytesRec := wBytesRec + ' Bytes';
                     Self.adicionaDadosNTop(wDominio,wBytesEnv,
                                            wBytesRec);
                  end;
            end;
         finally
      end;
   end;
end;
```

Quadro 4 – Procedimento que captura informações do Ntop

O protocolo SMB foi utilizado para estabelecer a comunicação entre a ferramenta agente e a ferramenta gerente e este é implicitamente apresentado no Quadro 5 com o procedimento `CriaMailSlot` da classe `TMailSlot` através do comando `CreateMailSlot`. Este comando possui como função criar uma caixa de correio, conhecido como *mailbox*, para o recebimento ou envio das mensagens. O comando `CreateMailSlot` é nativo de uma biblioteca do Microsoft Windows e de uso exclusivo para o protocolo SMB.

```
procedure TMailSlot.CriaMailSlot;
begin
   Self.FLocal := '\\.\mailslot\' + cMailBox;
   Self.FHandleOrigem := CreateMailslot(PChar(Self.FLocal),0,0,nil);

   if Self.FHandleOrigem = INVALID_HANDLE_VALUE then
      Self.SetAtivo(False);
end;
```

Quadro 5 – Procedimento que cria uma caixa de correio do SMB

Outro comando também nativo desta mesma biblioteca do Microsoft Windows e de uso exclusivo para o protocolo SMB é o `CreateFile` que tem como função enviar a mensagem para alguma caixa de correio determinada, conforme visto no Quadro 6 no procedimento `Enviar` da classe `TMailSlot`.

```
procedure TMailSlot.Enviar;
var
   wBytes : DWord;
   wDestino : String;
begin
   wDestino := '\\' + Self.FMicroDestino + '\mailslot\' + cMailBox;
   Self.FhandleDestino := CreateFile(PChar(wDestino),GENERIC_WRITE,
                                     FILE_SHARE_READ,nil,CREATE_ALWAYS,
                                     FILE_ATTRIBUTE_NORMAL,0);
   try
      if Self.FHandleDestino = INVALID_HANDLE_VALUE then
         exit
      else
         WriteFile(Self.FHandleDestino,
                   Pointer(Self.FMensagemEnviar.Text)^,
                           length(Self.FMensagemEnviar.Text),
                           wBytes,nil);
   finally
      CloseHandle(Self.FHandleDestino);
   end;
end;
```

Quadro 6 – Procedimento que envia mensagem para uma caixa de correio do SMB

Um último comando utilizado na implementação da biblioteca do Microsoft Windows citada e também de uso exclusivo para o protocolo SMB é o `ReadFile` que é responsável pelo recebimento da mensagem.

Para transmissão das informações pela rede, estas tiveram que ser quebradas em partes quando a seqüência era superior a duzentos e cinqüenta e cinco bytes, então, como pode ser observado no Quadro 7, o procedimento `EnviarMensagem` da classe `TMailSlot` é responsável

pelo tratamento para o envio segmentado dos dados da ferramenta agente para a ferramenta gerente.

```
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);
                     Insert(2,'#CONTINUA#');
                     Self.Enviar;
                  end;

               Clear;
               Add('#FIM#');
               Insert(0,Self.FMicroOrigem);
               Insert(1,Self.FUsuario);
               Self.Enviar;
            end;
      end;

end;
```

Quadro 7 – Procedimento que trata e segmenta informações

5.3.2 Operacionalidade da implementação

Primeiramente é importante destacar que o software é executado em duas partes distintas, o agente e o gerente.

É necessária a instalação e configuração do aplicativo Ntop na estação, fazendo com que o mesmo inicie automaticamente quando o sistema operacional desta estação for carregado. O mesmo procedimento deve ser efetuado com relação à ferramenta agente, ou seja, ela deve ser instalada na estação, fazendo também com que a mesma inicie automaticamente quando o sistema operacional desta estação for carregado.

O agente é a ferramenta que é instanciada em todas as estações, que por sua vez, imediatamente após iniciada, apresenta uma janela pedindo para informar o nome da estação de gerência, ou seja, a estação que possui a ferramenta gerente instalada, conforme a Figura 7. Isto ocorre apenas na primeira execução do agente ou quando o arquivo que armazena esta informação não for encontrado. Após isto a ferramenta agente passa a capturar as informações monitoradas pela aplicação Ntop registrando estas informações em um arquivo na própria estação.

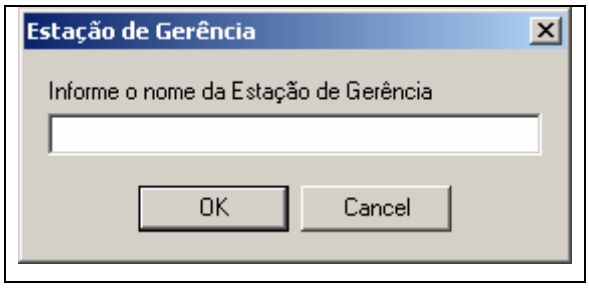


Figura 7 – Janela de solicitação da estação de gerência pelo agente

Para a administração é utilizada a ferramenta gerente, que acumula as informações dos acessos à Internet de todas as estações da rede e as apresenta para o administrador. A apresentação das informações é feita dividida por estações, ou seja, são apresentados os domínios acessados de cada estação da rede em separado. Tudo isto é feito através de um menu *Tree-View* tornando simples a sua operação e visualização, como pode ser observado na Figura 8.

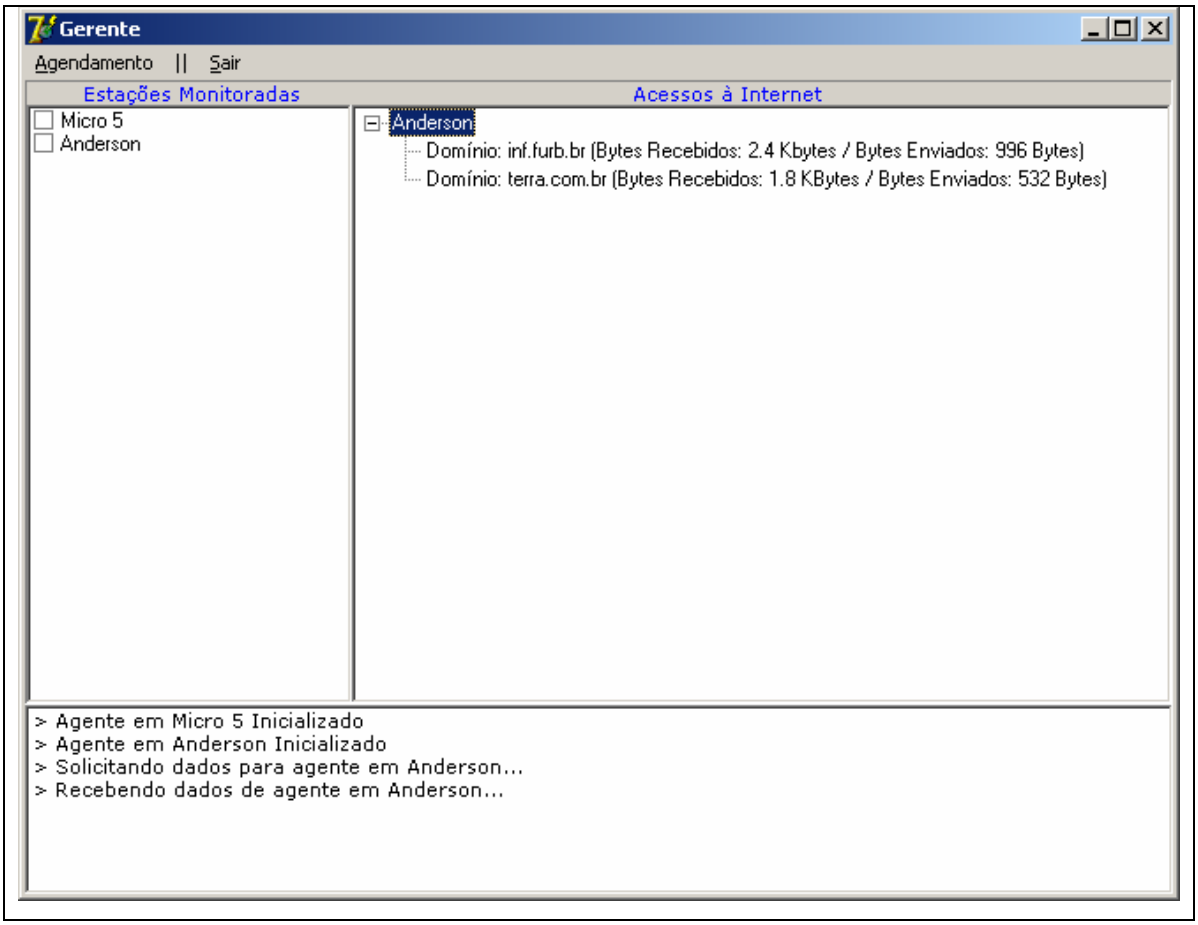


Figura 8 – Tela do gerente

Além de apresentar as estações monitoradas e os acessos à Internet com os domínios e os respectivos bytes recebidos e enviados, é gerado um registro no rodapé da tela onde são apresentadas todas as situações dos agentes.

Existe um botão chamado Agendamento que permite ao administrador informar o dia da semana e horário para a solicitação dos dados aos agentes e também existe um botão chamado sair para finalizar a aplicação gerente.

A Figura 9 apresenta a utilização do botão direito do *mouse* sobre alguma determinada parte da área das estações monitoradas para que um menu ofereça as opções de solicitar dados de todas estações ou solicitar dados das estações selecionadas.

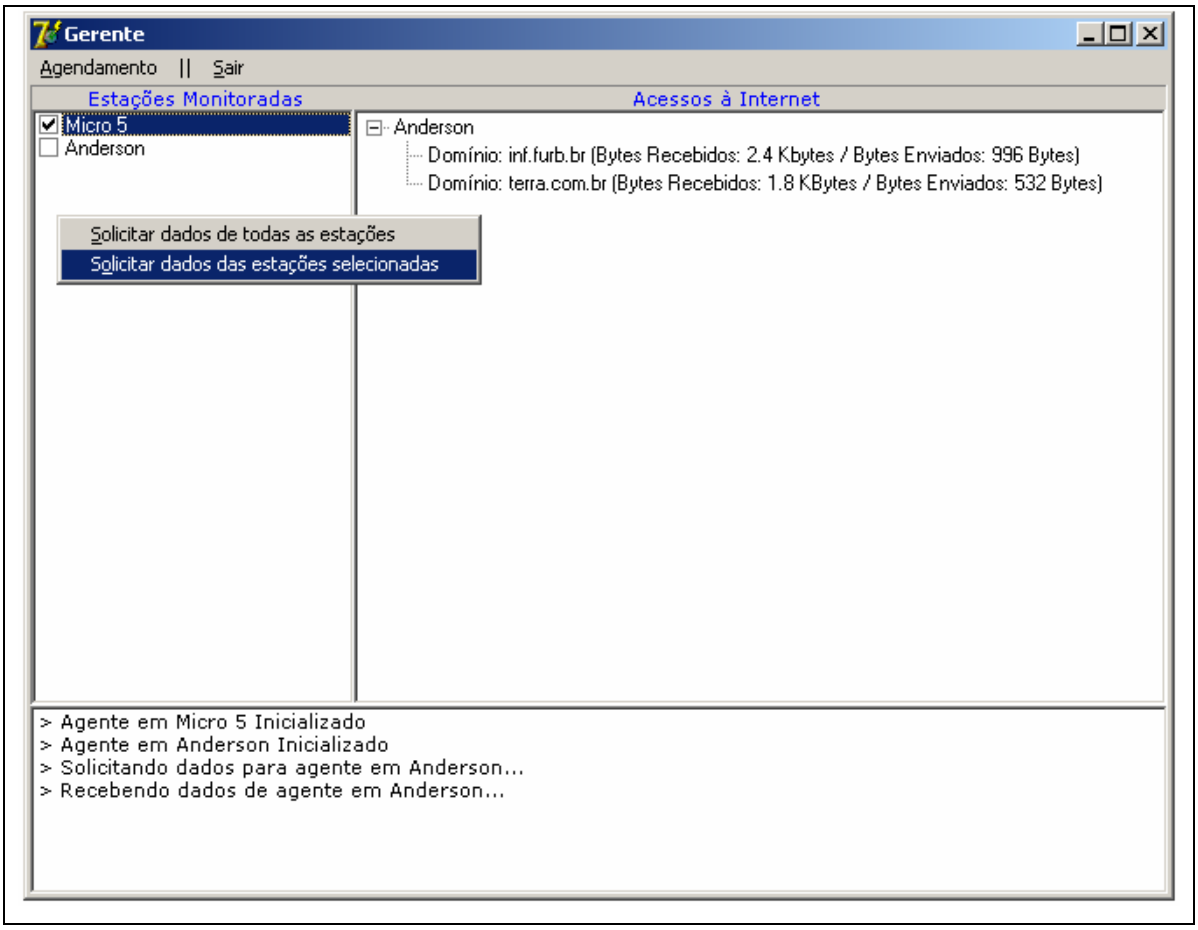


Figura 9 – Tela do gerente com o menu de solicitação

5.4 RESULTADOS E DISCUSSÃO

Observando o funcionamento das ferramentas agente e gerente, todas as informações são apresentadas no gerente no mesmo momento em que os dados são solicitados aos agentes, ou seja, além de ter os acessos à Internet representados pelo domínio, bytes recebidos e bytes enviados de todas as estações da rede na tela pode-se também acompanhar 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 desligada, porque quando isto ocorre, aquela estação sai da lista do gerente.

Além de ver as estações em tempo real, pode-se fazer uma análise mais detalhada verificando o registro gerado pelo gerente no rodapé, neste local são apresentadas as inicializações e finalizações dos agentes em cada estação, como também as solicitações e

recebimentos dos dados de acessos à Internet.

Comparando este software ao ISA Server (MICROSOFT CORPORATION, 2006), pode-se observar que este não tem seu processamento centralizado em um servidor, porém, depende de outro software para ser utilizado.

Agora comparando com o Squid Web Proxy Cache (SQUID DEVELOPMENT PROJECTS, 2006), além da vantagem de este não ter o processamento centralizado, também tem a vantagem de não ser exclusivo para a plataforma Linux mas continua tendo a dependência de outro software.

Já comparando ao Web-Fi Server (TERRASOFT SOFTWARE, 2006), também não tem seu processamento centralizado e não requer o Microsoft Internet Explorer como navegador padrão na rede, mas depende de outro software para ser utilizado.

Por último, comparando com o INCA (INSITE SOLUÇÕES INTERNET, 1995), o mais próximo pois descentraliza o processamento, tem a vantagem de não ser exclusivo para as plataformas Unix, Linux ou Solaris mas continua tendo a desvantagem de depender de outro software.

Vários testes foram realizados em um ambiente corporativo com vinte e nove estações e todos foram muito satisfatórios, mostrando ser rápido e preciso nas informações obtidas das estações e na usabilidade da ferramenta gerente.

6 CONCLUSÕES

Com os estudos e implementações que foram feitos neste trabalho, conclui-se que é bem simples gerenciar o acesso à Internet das estações de uma rede de forma descentralizada, porque existem diversos protocolos que auxiliam neste gerenciamento.

A segurança com relação a manter os agentes sem o risco dos usuários das estações finalizarem os mesmos não pode ser garantida.

Como os acessos à Internet são armazenados na própria estação em um arquivo, não foi possível garantir a segurança de que os usuários encontrem e modifiquem o conteúdo deste arquivo.

O Ntop foi uma ótima escolha para monitorar os acessos à Internet porém a versão para Windows do mesmo possui uma limitação de mil pacotes e isto prejudica quando é necessário realizar testes mais longos pois implica em ter que desinstalar e instalar novamente o referido aplicativo.

A captura dos dados monitorados pelo Ntop foi bastante simples, já que o mesmo oferece os arquivos em HTML e sendo assim bastou rastrear determinados identificadores para encontrar as informações desejadas.

Apresentar os dados capturados com o agente do Ntop de maneira legível foi fácil e rápido pois o ambiente de programação Delphi possui vários recursos que ajudam neste aspecto.

A transmissão de dados utilizando o protocolo SMB atendeu perfeitamente todos os requisitos. O tempo de resposta é bastante satisfatório, bem como a confiabilidade de transmissão das informações, apesar de que foi necessário quebrar os dados para poder transmitir todas as informações necessárias entre a ferramenta agente e a ferramenta gerente.

Todos os requisitos foram supridos com o sistema, que visa apresentar de modo simples e rápido os acessos à Internet realizados pelas estações de uma rede.

Os objetivos do trabalho foram alcançados, sendo implementado no sistema uma ferramenta agente e uma ferramenta gerente que possibilitam monitorar os acessos à Internet podendo estas informações serem requisitadas a qualquer momento para todas estações ou para determinas estações selecionadas.

6.1 EXTENSÕES

Como extensão deste trabalho, sugere-se criar uma segurança por meio de criptografia para os arquivos que guardam os acessos à Internet de cada estação ou persistir as mesmas em uma base de dados segura.

Também seria interessante apresentar na ferramenta gerente outros dados além das páginas acessadas, bytes recebidos e bytes enviados como por exemplo, data e horário de acesso da página e o tempo decorrido na utilização da página.

Aprimorar a parte gerencial, ou seja, permitir a impressão de relatórios dos acessos à Internet de uma determinada estação ou de todas estações também seria interessante.

Como o código fonte do Ntop está disponível, outra extensão seria a incorporação deste no agente, retirando também a limitação existente de mil pacotes da versão binária para Windows.

Uma última sugestão de extensão seria garantir que o agente não possa ser finalizado nas estações sem o consentimento do administrador.

REFERÊNCIAS BIBLIOGRÁFICAS

INSITE SOLUÇÕES INTERNET. **INCA.** Santana da Paraíba, 1995. Disponível em: <http://www.insite.com.br/inca>. Acesso em: 31 out. 2007.

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a internet**: uma abordagem top-down. São Paulo: Person Addison Wesley, 2005. 634 p.

LOPES, R. V.; SAUVÉ, J. P.; NICOLLETTI, P. S. **Melhores práticas para gerência de redes de computadores**. Rio de Janeiro: Campus, 2003. 373 p.

MICROSOFT CORPORATION. **What is ISA Server 2006?** [S.l.], 2006. Disponível em: <http://www.microsoft.com/isaserver/default.mspx>. Acesso em: 31 out. 2007.

NTOP.ORG. [S.l.], 2006. Disponível em: <www.ntop.org>. Acesso em: 31 ago. 2007.

OLIVEIRA, E. N. **Protótipo de software para gerência de patrimônio dos equipamentos de uma rede utilizando Session Message Block**. 2005. 53 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau. Disponível em: <http://www.inf.furb.br/~pericas/orientacoes/gerenciasmb2005.pdf>. Acesso em: 15 ago. 2007.

PÉRICAS, F. A. **Redes de computadores**: conceitos e a arquitetura internet. Blumenau: EdiFURB, 2003. 158 p.

SHARPE, R. **Just what is SMB?** [S.l.], 2002. Disponível em: <http://samba.anu.edu.au/cifs/docs/what-is-smb.html>. Acesso em: 31 ago. 2007.

SQUID DEVELOPMENT PROJECTS. **Squid Web Proxy Cache.** [S.l.], 2006. Disponível em: <www.squid-cache.org>. Acesso em: 31 out. 2007.

TANENBAUM, A. S. **Redes de computadores**. Rio de Janeiro: Campus, 1994. 786 p.

TERRASOFT SOFTWARE. **Web-Fi Server.** [S.l.], 2006. Disponível em: <http://www.terrasoft.com.br/web-fi-server>. Acesso em: 31 out. 2007.

WOOD, D. **SMB HowTo.** [S.l.], 2000. Disponível em: <http://www.tldp.org/howto/smb-howto.html>. Acesso em: 31 ago. 2007.

ZANONI, G. **Monitorando redes usando ntop**. Vitória, 2007. Disponível em: <http://www.imasters.com.br/artigo/6498/redes/monitorando_redes_utilizando_ntop>. Acesso em: 06 set. 2007.