

Implementação de Novos Agentes para Gerência em Redes de Computadores

Ana Cristina Benso da Silva * Carlos Becker Westphall †

CPGCC, Instituto de Informática, UFRGS
Caixa Postal 15064
91501-970 Porto Alegre - RS

Sumário

Este artigo apresenta o resultado da aplicação prática de alguns conceitos teóricos ministrados na disciplina de Gerência em Redes de Computadores do Curso de Pós-Graduação em Ciência da Computação da UFRGS. Os processos para gerência de redes envolvidos em uma associação específica assumem dois papéis possíveis: Gerente e/ou Agente. Neste sentido, seis novos agentes foram propostos com as suas respectivas funções de gerência e validados pela integração dos mesmos no ambiente de gerência da rede local do Instituto de Informática da UFRGS.

1 Introdução

A gerência em redes de computadores torna-se difícil porque as redes crescem em performance e complexidade, suportando um grande conjunto de serviços. Os sistemas de telecomunicações são muito complexos e serão cada vez mais no futuro. Eles poderão ser gerenciados somente se uma estrutura com bons princípios for seguida [WES 92].

Considerando que existem ferramentas insuficientes para solucionar os problemas e que, estas poucas, nem sempre são usadas nas organizações que utilizam redes de computadores, um aperfeiçoamento nos recursos envolvidos neste ambiente é necessário a fim de satisfazer, mais adequadamente, a gerência em redes de computadores.

Este trabalho é o resultado da aplicação prática de alguns conceitos

*Mestranda do CPGCC (Curso de Pós-Graduação em Ciência da Computação) da UFRGS, (benso@inf.ufrgs.br).

†Professor Responsável pela Disciplina de Gerência em Redes de Computadores do CPGCC da UFRGS, (westphal@inf.ufrgs.br).

ministrados na disciplina de Gerência em Redes de Computadores do Curso de Pós-Graduação da UFRGS, no primeiro semestre de 1992. A implementação de novos agentes para a gerência em redes de computadores foi o enfoque do trabalho. Utilizou-se para a validação prática o ambiente de gerência da rede local do Instituto de Informática da UFRGS, ou seja, o SunNet Manager.

Os agentes desenvolvidos e aqui apresentados propiciam novas funções ao ambiente de gerência. As fases de desenvolvimento do trabalho podem ser resumizadas em :

- definição do agente;
- implementação do agente;
- integração com o SunNet Manager, e
- testes.

A fase de definição e implementação dos agentes exigiram a familiarização com o software de gerência. O uso do software proporcionou as condições necessárias a identificação das novas funções a serem implementadas. A fase seguinte, de integração, exigiu o estudo da *API (Application Program Interface)* do SunNet Manager para a integração dos agentes. A interface entre o SunNet Manager e o agente é descrita pelo arquivo *.schema*, onde são estabelecidos os atributos do agente e os tipos de dados que compõem cada atributo, ou seja, os dados resultante da execução do agente com aquele atributo. A fase final se constitui dos testes realizados para o bom desempenho dos agentes integrados.

O trabalho foi desenvolvido em *workstations* SUNs, sistema operacional SunOS 4.1 compatível com o sistema Unix 4.3 BSD, compilador C do sistema SunOS, o ambiente de janelas OpenWindows 2.1 e o software de gerência de redes SunNet Manager.

A seção dois apresenta o ambiente de gerência em redes de computadores. A seção três mostra os novos agentes para gerência em redes. A seção quatro descreve a implementação dos novos agentes e suas respectivas funções de gerência. A seção cinco sugere alguns aperfeiçoamentos que podem ser realizados no trabalho desenvolvido.

2 O Ambiente de Gerência em Redes de Computadores

Esta seção visa abordar alguns conceitos necessários para o entendimento dos assuntos descritos ao longo deste trabalho, baseando-se nas proposições da ISO/OSI para gerência em redes de computadores.

As tarefas básicas da gerência em redes, simplificada, são obter informações da rede, tratar estas informações, possibilitando um diagnóstico, e encaminhar as soluções dos problemas. Para cumprir estes objetivos, *funções de gerência*

devem ser embutidas nos diversos componentes de uma rede, possibilitando descobrir, prever e reagir a problemas [WES 88].

Para resolver os problemas associados a gerência em redes a ISO através do OSI/NM propôs três modelos [WES 91]:

- O Modelo Organizacional estabelece a hierarquia entre sistemas de gerência em um domínio de gerência, dividindo o ambiente a ser gerenciado em vários domínios.
- O Modelo Informacional define os objetos de gerência, as relações e as operações sobre esses objetos. Uma MIB (Management Information Base) é necessária para armazenar os objetos gerenciados.
- O Modelo Funcional descreve as funcionalidades de gerência: gerência de falhas, gerência de configuração, gerência de performance, gerência de contabilidade e gerência de segurança.

Gerência de rede é uma aplicação distribuída que envolve as trocas de informações de gerência entre processos de gerência com a finalidade de monitorar e controlar os diversos recursos da rede. Os processos envolvidos em uma associação específica assumem dois papéis possíveis: Gerente e/ou Agente. O *gerente* é a parte da aplicação distribuída que gera operações de gerência e recebe notificações. O *agente* é parte da aplicação distribuída que gere os objetos gerenciados a ele associados (respondendo às operações solicitadas pelo gerente e emitindo notificações que refletem o comportamento dos objetos).

3 Novos Agentes para Gerência de Redes

Os novos agentes propostos executam funções de gerência com alguns aspectos simples e outros mais complexos. Os novos agentes auxiliam na gerência de redes possibilitando: monitoração de processos, distribuição de diretórios, monitoração de usuários, monitoração do tráfego entre sub-redes, descoberta de rotas ativas num domínio internet e monitoração do tráfego de datagramas.

3.1 Agente 1: Monitoração de Processos

O agente 1 realiza a monitoração dos processos. Assim o administrador da rede tem implementado no ambiente SunNet Manager as funções do comando *ps* [SUN 90], do sistema operacional SunOS, que mostra o *status* dos processos em execução.

3.2 Agente 2: Descobrir Rotas Ativas num Domínio Administrativo Internet

O agente 2 fornece informações sobre a topologia de uma rede pertencente a um domínio administrativo. A função do agente é descobrir os roteadores existentes

num domínio Internet através dos datagramas do protocolo ICMP (Internet Control Message Protocol).

3.3 Agente 3: Monitoração de Usuários

O agente 3 monitora o acesso dos usuários da rede. É possível com esse agente verificar os usuários *logados* local ou remotamente, bem como as máquinas livres.

3.4 Agente 4: Monitoração do Tráfego entre Sub-redes

O agente 4 analisa o tráfego entre as sub-redes Internet. A coleta de informações referente a troca de pacotes por máquinas de diferentes sub-redes permite traçar estatísticas que auxiliam a administração da rede na verificação de fluxos inadequados para o bom desempenho da rede.

3.5 Agente 5: Distribuição de Diretórios

Este agente implementa as funções do comando *mount* com a opção *-p* [SUN 90], do sistema operacional SunOS, que imprime a lista de sistemas de arquivos montados, permitindo ao administrador da rede observar como estão configurados os diretórios de uma estação cliente. O agente permite que se conheça a máquina servidora, o nome do diretório na máquina servidora, o nome do diretório local correspondente, as permissões de acesso a este diretório e o intervalo de dias entre cada *backup* do diretório.

3.6 Agente 6: Monitoração do Tráfego de Datagramas

O último é um agente de monitoração do tráfego de datagramas IP que passa pela estação que atua como roteador, permitindo a contabilização de uso dos recursos, uma vez que foi identificado quais são os equipamentos envolvidos na transmissão de cada datagrama. Este agente reúne informações nos moldes do agente *Traffic*, o qual é distribuído em conjunto com o pacote SunNet Manager, mas difere quanto a semântica das informações. As informações do agente 6 dizem respeito aos datagramas enviados a nível de enlace ao gateway para serem redirecionados para outros componentes de outras redes ao nível de rede, e o *Traffic* mantém informações sobre os datagramas que trafegam pela rede enviados ou não à estação que roda o agente. Uma segunda diferença está no fato do agente 6 não se restringir à interface Ethernet. Todos os pacotes que chegam, independentes do meio de transmissão, são analisados.

4 Implementação dos Novos Agentes para Gerência

As subseções abaixo detêm-se na descrição da implementação do núcleo do agente, abstraindo a interface com o SunNet Manager e o método de desenvolvimento de um agente para o ambiente de gerência [SUN 89].

4.1 Agente 1: Monitoração de Processos

O agente 1 é implementado executando o comando *ps*, do sistema operacional SunOS, com os parâmetros requisitados pelo usuário. Os parâmetros são mapeados através da seleção dos atributos do agente. Os atributos estão definidos no arquivo *agent1.schema*, e são os seguintes:

- **sumário:** define os parâmetros para que somente os processos do usuário sejam listados;
- **rodando:** define os parâmetros para fornecer informações restritas de saída dos processos que estão rodando nos estados: *runnable*, *page wait*, *inshort-term non-interruptible waits*;
- **extenso:** define os parâmetros para fornecer as informações conjugadas dos atributos anteriores, incluindo os processos não pertencentes ao usuário.

O tipo de dados retornados pelo agente é idêntico para todos os atributos, diferindo apenas na semântica da informação, conforme os parâmetros de execução.

4.2 Agente 2: Descobrendo Rotas Ativas num Domínio Administrativo Internet

Através do utilitário *ping* [SUN 90], que envia pacotes ICMP [POS 81a] para *hosts* da rede, é possível descobrir todos os roteadores existentes num domínio Internet. O utilitário *ping* utiliza o datagrama *echo_request* do protocolo ICMP para solicitar um ICMP *echo_response* de um host específico ou de um gateway. A recepção de um *echo_response* sinaliza que o host ou gateway está em funcionamento. Também, é possível descobrir a(s) rota(s) disponível(is) para se alcançar um host ou gateway. O utilitário *ping* implementado na SUN deve receber "*-s -lRv 64 1*" como parâmetros para executar este serviço. O parâmetro "*-s*" requisita a exibição da rota na saída padrão.

Os parâmetros "*-lRv*" requisitam que a opção *record route* do protocolo IP seja setada, isto é, requisitam o armazenamento da rota do pacote dentro do cabeçalho IP; o valor "*64*" informa o tamanho desejado do datagrama em bytes; o valor "*1*" requisita apenas a recepção de um *echo_response*. Por tanto, para se descobrir a rota até a sub-rede 143.54.7.0, por exemplo, usa-se :

```
% ping -s -lRv 143.54.7.0 64 1
```

resultando em :

```
PING 143.54.7.0: 64 data bytes.
```

```
72 bytes from canopus (143.54.8.7): icmp_seq=0, time=5 ms.
```

```
IP options: <loose source route> minuano-gw (143.54.11.2),  
<record route> rigel-gw (143.54.11.3), minuano-gw (143.54.11.2),  
rigel (143.54.8.6), canopus (143.54.8.7), 0.0.0.0(Current),  
0.0.0.0, 0.0.0.0.
```

```
---143.54.7.0 PING Statistics---
```

```
1 packets transmitted, 1 packets received, 0% packet loss,  
round-trip (ms) min/avg/max = 5/5/5.
```

Para se descobrir a rota deve-se seguir o seguinte algoritmo:

1. Definição das estruturas de dados usadas

- Tabela de roteadores

```
string[20] nome description "Nome do Roteador"
```

```
string[16] ipif1 description "Endereco IP da interface 1"
```

```
string[16] ipif2 description "Endereco IP da interface 2"
```

2. Obter a lista de estações visitadas

- Ação: Deve-se inverter a *record route*

- Resultado: "canopus (143.54.8.7), rigel (143.54.8.6), minuano-gw (143.54.11.2), rigel-gw (143.54.11.3)"

3. Enquanto houver estações na lista

- Lê estação (por exemplo, a primeira estação é a "canopus (143.54.8.7)")

- Endereco de rede é igual ao da última estação lida ?

sim

– tr = penúltima estação lida

– Se tr.ipif2 não contiver nenhum endereço

* tr.ipif2 = endereço IP da interface ethernet 2

(Se tr.nome não é da estação de onde partiu o ping então tr.ipif2, ele deve ser interpretado como o endereço IP do roteador para o segmento de rede onde se localiza tr.nome)

* Se é a última estação da rota

- tr.ipif2 = endereço IP da rede a ser alcançada (por exemplo, 143.54.7.0)

- tr = antepenúltima estação lida

- tr.ipif2 = endereço IP da interface ethernet 2

* Continue em 3

- Marca estação como visitada

- tr.nome = nome da estação

- tr.ipif1 = endereço IP da interface ethernet 1

4. Continua em 3

No final, obter-se-ia a Tabela de Roteadores demonstrada abaixo:

NOME	IPIF1	IPIF2
canopus	143.54.8.7	143.54.8.6
rigel	143.54.8.6	143.54.11.3
minuano-gw	143.54.11.2	143.54.7.0

Tabela 1: Tabela de Roteadores

A linha 0 da tabela não descreve um roteador, mas sim informações da estação de onde foi disparado o *ping*. A coluna *nome* indica o nome da estação; a coluna *ipif1* informa o endereço IP da estação; e a coluna *ipif2* informa o endereço IP da estação roteadora do segmento onde se localiza *ipif1*. As demais linhas descrevem os roteadores. O roteador *rigel* interliga duas redes IP: a 143.54.8.0 e a 143.54.11.0. O roteador *minuano-gw* interliga duas redes IP: a 143.54.11.0 e a 143.54.7.0. Pelo exemplo percebe-se que não foi obtido o endereço IP da segunda interface do roteador *minuano-gw*. Isto significa que não há nenhuma sub-rede Internet alcançável a partir deste roteador e desta rede (*ipif2*), para o domínio administrativo em questão.

4.3 Agente 3: Monitoração de Usuários

O agente 3 é implementado utilizando o comando *rusers* [SUN 90] do sistema operacional SunOS. No arquivo *agente3.schema* estão definidos quatro atributos que definem a abrangência de execução do utilitário. Os atributos são:

- **estacao:** executa *rusers* apenas na estação solicitada;
- **global:** executa *rusers* em todas as estações da rede;
- **local:** executa *rusers* apenas nas estações da sub-rede da estação solicitada;
- **servidoras:** executa *rusers* apenas nas estações servidoras de rede.

O tipo de dados retornados para estes atributos são os mesmos. É retornado o nome da estação e o *username* do usuário *logado*. Pode-se ter mais de um usuário listado para uma mesma máquina, indicando que tais usuários estão remotamente *logados*.

4.4 Agente 4: Monitoração do Tráfego entre Sub-redes

Uma vez que era necessário coletar informações de 5 sub-redes diferentes e que essas informações deveriam ser direcionadas rapidamente para um único hospedeiro, optou-se pela criação de um pequeno programa chamado Coletor. Cada sub-rede possui um Coletor instalado em uma de suas máquinas. Esse programa fica inativo até receber uma chamada do agente 4, quando coleta informações sobre um determinado número de pacotes que estão transitando, naquele momento, na sub-rede em que ele se encontra. Tão logo tenha sido feita a análise dos pacotes coletados, o Coletor envia para o agente 4 estatísticas concernentes ao tráfego analisado. Tal arquitetura pode ser melhor compreendida a partir da observação da figura 1.

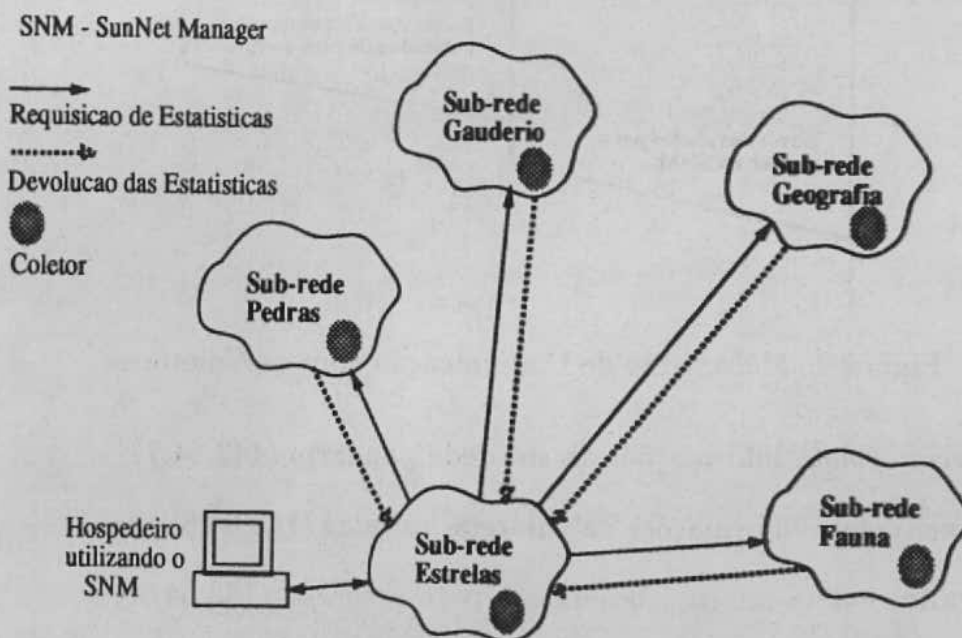


Figura 1: Arquitetura Operacional do Agente 4.

A coleta de dados é realizada através de um utilitário do SunOS denominado *etherfind* [SUN 90]. A comunicação entre o agente 4 e os coletores foi implementada através do uso de RPC (Chamadas de Procedimentos Remotos) [SUN 89a]. Esse mecanismo permite que o Coletor fique inativo até que a requisição de uma amostra de pacotes seja efetuada. Após o envio das estatísticas coletadas para o agente, o Coletor volta ao seu estado inativo, como é elucidado na figura 2.

O SunNet Manager está estruturado de modo que, a interface com o agente é descrita num arquivo *.schema*, e a comunicação com esse agente também ocorre por RPC. Isso permite que se tenha uma independência dos dados coletados em relação ao meio como os mesmos serão visualizados no SunNet Manager. As informações que o agente 4 pode fornecer ao SunNet Manager estão agrupadas em 6 tabelas. Cada uma dessas possui um conjunto específico de informações. No caso do agente 4 todas as tabelas possuem os mesmos campos. As tabelas existentes são as seguintes:

- pedras: coleta informações da sub-rede *pedras* (143.54.6);

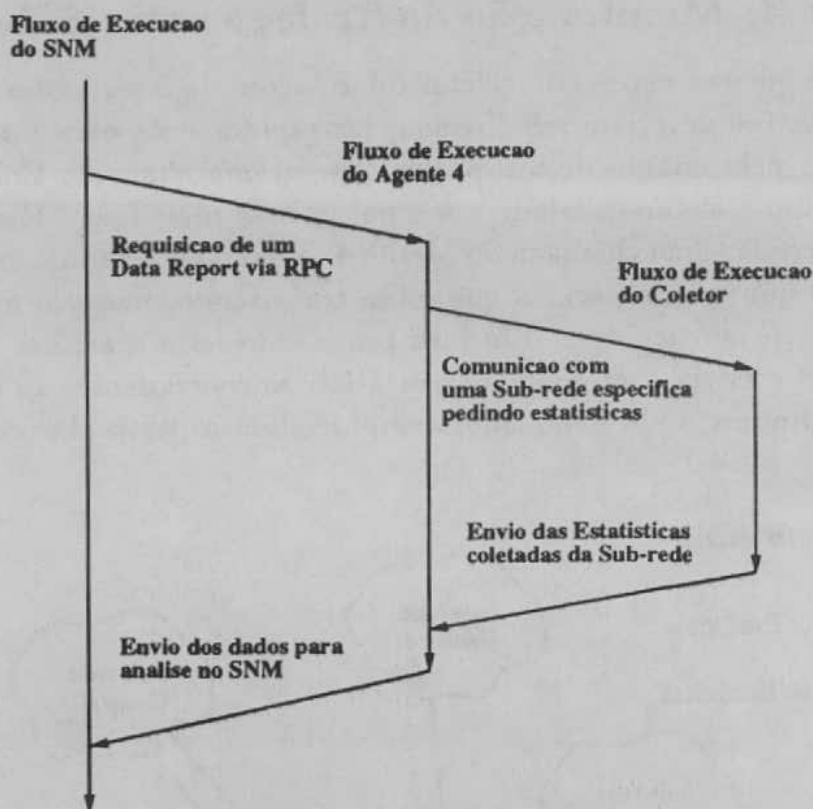


Figura 2: Mecanismo de Comunicação com os Coletores.

- **gauderio:** coleta informações da sub-rede *gauderio* (143.54.7);
- **estrelas:** coleta informações da sub-rede *estrelas* (143.54.8);
- **geografia:** coleta informações da sub-rede *geografia* (143.54.9);
- **fauna:** coleta informações da sub-rede *fauna* (143.54.10);
- **all_sub_nets:** coleta informações de todas as sub-redes.

Cada Coletor efetua somente a análise dos pacotes que são originados na sua sub-rede, evitando-se, dessa maneira, a contagem duplicada de pacotes, transmitidos de uma rede à outra, por diferentes Coletores. Os pacotes analisados pelos Coletores, fornecem dados para as seguintes estatísticas:

- **pkts:** número de pacotes analisados;
- **mid_size:** tamanho médio dos pacotes (em bytes);
- **udp:** quantidade de pacotes com protocolo udp;
- **tcp:** quantidade de pacotes com protocolo tcp;
- **other_proto:** quantidade de pacotes com outros protocolos;
- **pedras:** quantidade de pacotes destinados à sub-rede *pedras*;

- **gauderio**: quantidade de pacotes destinados à sub-rede *gauderio*;
- **estrelas**: quantidade de pacotes destinados à sub-rede *estrelas*;
- **geografia**: quantidade de pacotes destinados à sub-rede *geografia*;
- **fauna**: quantidade de pacotes destinados à sub-rede *fauna*;
- **others**: quantidade de pacotes destinados a outras sub-redes(externas).

O gráfico exibido na figura 3 se refere a monitoração da sub-rede *estrelas*. Observa-se em determinado momento um elevado número de pacotes destinados à sub-rede *pedras*. Esse comportamento foi induzido para ilustrar como se apresentaria um tráfego anormal, ou seja, um excessivo número de pacotes destinados a outra sub-rede.

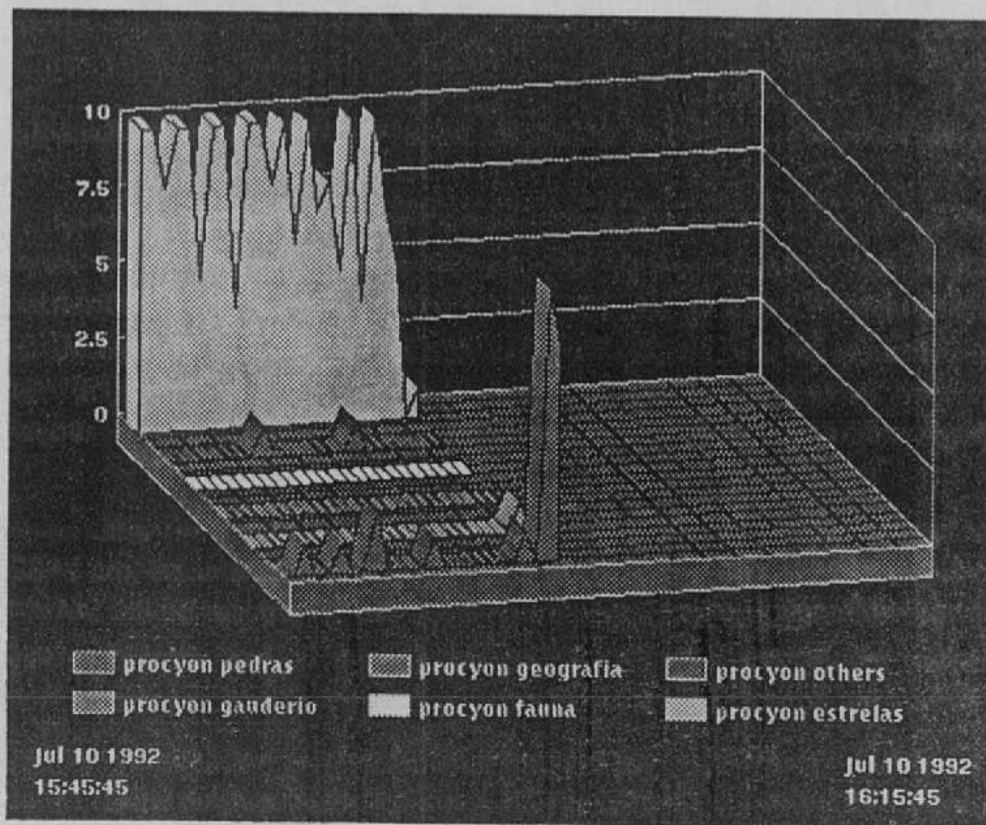


Figura 3: Monitoração do Tráfego na Sub-rede Pedras.

4.5 Agente 5: Distribuição de Diretórios

Os atributos do agente estão definidos no arquivo *.schema*. Os atributos são os argumentos retornados pelo agente. O atributo definido para o agente 5 é *tabela_diretorio*. Este atributo tem os seguintes tipos de dados:

```
string[20] maquina description "estação"  
string[60] dir_remoto description "diretório remoto"
```

```
string[60] dir_local description "diretório local"  
string[16] acesso description "permissões de acesso"  
int backup description "intervalo de tempo entre backups"
```

Os atributos são buscados no arquivo */etc/fstab*. Este arquivo guarda os diretórios correntes montados em uma determinada estação. É organizado em linhas, com a seguinte estrutura:

```
<nome do servidor>:<diretório no servidor><diretório local><tipo de sistema de arquivo><parâmetro do sistema de arquivos><tempo de backup><número de tentativas de leitura>
```

Poder-se-ia obter tais informações pelo comando *mount -p*, entretanto isto não traria nenhum ganho para o agente. De qualquer forma, os dados retornados teriam de ser colocados em um buffer e, posteriormente, teriam de ser organizados. Além disso a execução do comando *mount* não é rápida.

4.6 Agente 6: Monitoração do Tráfego de Datagramas

O agente 6 realiza sua tarefa capturando todos os datagramas IP [COM 91] que chegam ao roteador. De cada datagrama, retira-se e usa-se o seu cabeçalho para a atualização de estatísticas, as quais são enviadas ao gerente nos intervalos por este determinado. O cabeçalho de datagramas IP carrega consigo uma série de informações interessantes, considerando-se a tarefa de acompanhamento de tráfego. Algumas das informações disponíveis são [POS 81]:

- Endereço IP dos equipamentos origem e destino do datagrama;
- Protocolo ao qual pertence os dados contidos no datagrama após ao cabeçalho;
- Tempo restante para que o datagrama seja descartado pelo roteador, ou *Time-to-Live*;
- Comprimento do datagrama e do cabeçalho.

A partir destas informações, podem ser coletadas estatísticas as quais podem servir para avaliar o desempenho de tráfego. A partir dos endereços contidos no cabeçalho, por exemplo, podem ser geradas tabelas. A primeira tabela, *Datagramas de*, traz em cada uma de suas entradas uma tupla do tipo (*host origem, número de datagramas enviados*). Esta tabela expressa o número de datagramas originados por determinado computador os quais passam pelo roteador, onde o agente 6 está ativo. Pode ser gerada também uma tabela análoga à primeira, *Datagramas para*, a qual demonstra quantos datagramas são enviados para cada computador através do roteador. Ainda com base nos endereços contidos nos cabeçalhos, uma terceira tabela pode ser obtida: a tabela *Entre*. Esta tabela informa o número de datagramas trocados entre dois determinados computadores. Uma tabela discriminando protocolos e datagramas enviados também pode ser mantida pelo agente. Além de tabelas,

podem ser geradas estatísticas, como médias e desvio-padrão, sobre as informações de comprimento e *Time-to-Live*.

A estrutura do agente se fundamenta em dois módulos: o processo encarregado de receber os datagramas IP, ou capturador, e o processo encarregado da manutenção das estatísticas e da interface com o gerente, ou interfaceador. A necessidade de atribuir a tarefa de recepção dos datagramas a um processo dedicado a esta tarefa vem da natureza dinâmica esperada do roteador onde o agente 6 está em execução, ou seja, a manipulação de um grande volume de datagramas recebidos em uma velocidade alta nos momentos em que o tráfego atinge seu pico. A figura 4 ilustra a estrutura do agente.

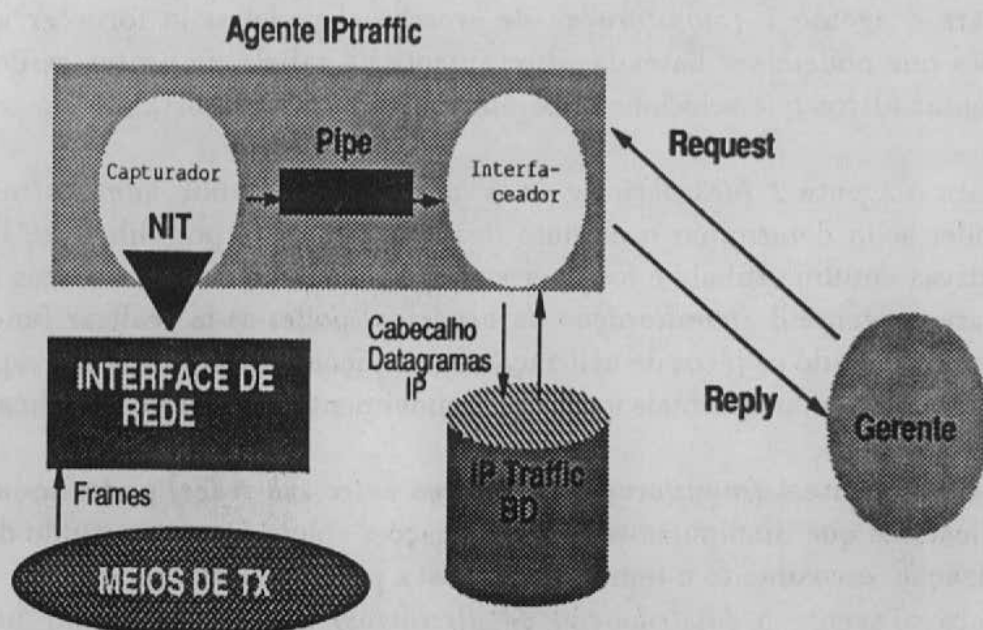


Figura 4: A Estrutura do Agente 6.

A partir do instante em que recebe uma requisição do gerente, o agente 6 inicia sua atividade. Através da chamada de sistema *fork()*, cria o processo capturador, cuja única tarefa é receber os datagramas e passar seus cabeçalhos para o interfaceador. O módulo do agente que faz a captura de datagramas se utiliza de um mecanismo provido pelo sistema operacional SunOS chamado *Network Interface Tap (NIT)* [SUN 90]. O *NIT* é uma facilidade composta de diversos componentes e *drivers*. Estes elementos fornecem mecanismos para o desenvolvimento de aplicações que requerem acesso à rede ao nível de enlace.

Algumas das possibilidades providas pelo *NIT* são o armazenamento dos *frames* recebidos em *buffers*, para redução do *overhead* acarretado pela repetição de chamadas de sistema *read()*, assim como a filtragem pelo próprio *NIT* do tipo de *frame* a ser recebido (datagrama IP, por exemplo). Há ainda a possibilidade de agregar ao *frame* recebido uma marca de tempo e o número de *frames* perdidos devido a problemas de controle de fluxo (indicativo de um tempo muito longo para o tratamento das informações recebidas pela aplicação que utiliza *NIT*), assim como a opção de colocar a interface de rede, no caso de interfaces *ethernet*,

em modo promíscuo, capturando todos os *frames* que transitam pelo barramento. Particularmente, no que diz respeito ao agente 6, o modo promíscuo é ignorado.

Para cada datagrama recebido pelo processo capturador, o cabeçalho do datagrama IP e a marca de tempo da sua chegada são enviados através do mecanismo de *pipe* [STE 90] ao interfaceador, que é o módulo responsável pela manutenção de estatísticas e pela interface com o gerente. A seguir, o capturador volta ao estado de recepção de novo datagrama.

5 Aperfeiçoamentos

Para o agente 1 (*monitoração de processos*) poder-se-ia fornecer algumas informações que podem ser baseadas diretamente na tabela de processos do kernel e implementar filtros que selecionem as informações mais importantes do comando *ps*.

Para o agente 2 (*descobrir rotas ativas num domínio administrativo Internet*) poder-se-ia determinar o número de estações ativas por sub-rede, listar as estações ativas em uma tabela e fornecer suporte a mais de duas interfaces IP.

Para o agente 3 (*monitoração de usuários*) poder-se-ia realizar funções estatísticas relacionando os picos de utilização das estações nos horários correspondentes, distribuindo de maneira mais uniforme o movimento de usuários e aplicações ao longo da rede.

Para o agente 4 (*monitoração do tráfego entre sub-redes*) poder-se-ia desenvolver aplicativos que manipulassem as informações coletadas, permitindo determinar a utilização, escoamento e tempo de resposta para cada sub-rede.

Para o agente 5 (*distribuição de diretórios*) poder-se-ia definir um novo grupo de objetos para a gerência dos servidores de disco da rede, permitindo descobrir quais diretórios são oferecidos e em qual máquina. Possibilitar a montagem dinâmica de diretórios permitiria ao administrador da rede modificar mais facilmente a configuração de uma máquina, cliente ou servidora.

Para o agente 6 (*monitoração do tráfego de datagramas*) poder-se-ia desenvolver aplicativos que possibilitassem a apresentação dos resultados relacionados a avaliação de desempenho dos recursos em questão.

6 Conclusão

Observa-se que o principal objetivo foi atingido, ou seja, a aplicação prática de alguns conceitos ministrados em aula, possibilitando a realização de um trabalho de pesquisa com boa qualidade técnica e originalidade.

A experiência de implementação dos novos agentes foi bem elucidativa. As duas fases mais importantes foram a definição dos agentes e a integração dos mesmos no ambiente de gerência da rede local, deixando o novo software disponível ao uso para os administradores da rede.

O desenvolvimento destes agentes foi o passo inicial para a realização de trabalhos futuros neste contexto, inclusive alguns possíveis aperfeiçoamentos foram citados. Pretende-se também com a exploração das RPCs atribuir um caráter mais distribuído ao ambiente de gerência em redes de computadores.

7 Agradecimentos

Gostariamos de agradecer a valiosa contribuição daqueles que participaram do desenvolvimento das idéias aqui descritas, ou seja, os mestrandos: Carlos Eduardo Sampaio de Souza, Cleber Weissheimer, Daniel Francisco Sachet, Dinamerico Schwingel, Ewerton Longoni Madruga, Katia Barbosa Saikoski, Luis Fernando Nunes Fernandes, Marcelo Augusto Rauh Schmitt, Marcelo Spohn, Rodrigo Stumpf Trindade, Rosane Beatriz Oliveira Severo Torres e Valter Roesler.

Referências

- [COM 91] COMER, D. E. **Internetworking with TCP/IP: Principles, Protocols, and Architecture**. Volume 1. Seg. Edição. Prentice-Hall, Englewood Cliffs, NJ, 1991.
- [POS 81] POSTEL, J.B. **Internet Protocol**. Request for Comments 791, DDN Network Information Center, SRI International, Setembro, 1981, 45pp.
- [POS 81a] POSTEL, J.B. **"Internet Control Message Protocol - DARPA Internet Program Specification"**. Request for Comments 792, 1981.
- [SUN 89] Sun Microsystem Inc. **SunNet Manager Tutorial - How Write an Agent**, 1989.
- [SUN 89a] Sun Microsystem Inc. **Network Programming Guide**, 1989.
- [STE 90] STEVENS, W.R. **UNIX Network Programming**. Prentice-Hall Inc. Englewood Cliffs, NJ, 1990.
- [SUN 90] Sun Microsystems Inc. **SunOS Reference Manual**. Vol I, 1990.
- [WES 88] WESTPHALL, C. B. **Proposição de Funções em Gerência de Comunicação de Dados**. *Dissertação de Mestrado*, Porto Alegre, UFRGS-CPGCC, Maio 05, 1988.
- [WES 91] WESTPHALL, C. B. **Conception et développement de l'architecture d'administration d'un réseau métropolitain**. *Thèse de Doctorat nouveau régime*. Université Paul Sabatier. Toulouse, 16 Juillet 1991.

- [WES 92] WESTPHALL, C. B. & ASSUOL, S. **Management Architecture for Networks of the Future**. IFIP/IEEE International Workshop on Distributed Systems: Operations & Management. October 12-13, 1992. Munich, Germany.