Um Esquema para o Gerenciamento do Tráfego de Aplicações em Redes TCP-IP

Carlos Kelner Silveira * *Edmundo R. M. Madeira* †

Departamento de Ciência da Computação
IMECC - UNICAMP
CP 6065 - CEP: 13081 970 - Campinas - SP
E-mail: {ckelner,edmundo}@dcc.unicamp.br

Resumo

Este artigo apresenta um esquema para o gerenciamento do tráfego de aplicações entre grupos de computadores em redes TCP-IP. Foi desenvolvido um agente de gerenciamento, seguindo o modelo baseado no protocolo SNMP, que coleta os dados (número de bytes e datagramas por aplicação) e responde às requisições de gerentes SNMP. Alguns dados reais foram obtidos, usando-se o software de gerenciamento SunNet Manager e o agente desenvolvido, com o intuito de validar o esquema e verificar as vantagens de tal tipo de gerenciamento.

Abstract

This paper presents a scheme for the management of application traffic among groups of computers in TCP-IP networks. A management agent was developed, based on the SNMP management model, which collects data (bytes and datagrams per application) and replies requests from SNMP managers. Using the software SunNet Manager and the developed agent, some real data were obtained to validate the scheme and verify the advantages of this kind of management.

1 Introdução

As redes de computadores estão sendo cada vez mais utilizadas e a produtividade atingida com sua utilização, seja para ambientes eletrônicos de trabalho em grupo ou para compartilhamento de recursos, tornou-as indispensáveis para organizações eficientes.

Essa necessidade incentivou o surgimento de pesquisas na área de gerenciamento de redes, que originou dois modelos básicos: um baseado no CMIP (Common Management Information Protocol) proposto pela ISO (International Organization for Standardization) para ambiente OSI (Open Systems Interconnection) e outro baseado no SNMP (Simple Network Management Protocol) [CFSD90, Ros91], usado na Internet. Pela vasta quantidade de produtos que seguem o modelo baseado no SNMP, pode-se considerá-lo um padrão *de facto*.

*Mestrando do Departamanto de Ciência da Computação - Unicamp.
†Professor do Departamento de Ciência da Computação - Unicamp.

Como o SNMP originalmente não fornece meios de gerenciamento de protocolos a nível de aplicação [CM94b, RW94], e sentindo-se necessidade desse tipo de informação, este trabalho propõe um esquema para gerenciar o tráfego de aplicações entre grupos de computadores em redes TCP-IP.

O objetivo do esquema proposto é monitorar o tráfego entre grupos de computadores em redes TCP-IP. Estes grupos podem estar em qualquer lugar da Internet, desde que se observem algumas restrições definidas adiante. Além disso, o tráfego pode ser classificado pela aplicação que o gerou (correio eletrônico, transferência de arquivos, entre outros) [Com91, LR93].

Nossa intenção não é medir o tráfego instantâneo, mas o tráfego acumulado em intervalos, cuja duração é configurável. O agente de gerenciamento mantém informações sobre o tráfego acumulado no intervalo imediatamente anterior ao atual. Essas informações permanecem invariáveis até o final do intervalo, quando são atualizadas com o tráfego relativo ao intervalo recém terminado. Cabe ao gerente fazer requisições periódicas, com período igual à duração do intervalo configurado no agente, para ler as informações do agente e guardá-las para posterior consulta. Fizemos a opção do gerente guardar os dados dos vários intervalos, pois simplifica a implementação do agente e fornece ao usuário, que usa o aplicativo gerente, um maior controle sobre o processo de monitoração. Pode-se observar que, pelo fato dos dados no agente permanecerem invariáveis durante todo o intervalo, as requisições do gerente podem ser iniciadas em qualquer instante do primeiro intervalo de medição, não exigindo uma sincronização sofisticada.

As vantagens propiciadas por este gerenciamento são a identificação do tráfego dos diversos segmentos de uma dada rede, ou redes, possibilitando a localização de "gargalos". Essa localização pode ser bastante útil para a relocação de servidores diversos (de arquivos, de impressão, entre outros) e para a decisão de formar sub-redes.

Na seção 2, descreve-se o esquema proposto, na seção 3, a implementação de um protótipo e alguns resultados obtidos e na última seção, a conclusão.

2 Esquema de Gerenciamento de Aplicações

2.1 Introdução

De acordo com a camada da arquitetura TCP-IP que se observa, temos acesso a informações de diferentes tipos [Com91]:

- **Camada de Interface (enlace)**: O cabeçalho de cada quadro tem os endereços de origem e destino de nível físico, além de outras informações também dependentes do meio físico.
- **Camada de Rede**: No cabeçalho IP temos os endereços de origem e destino, cada um deles referindo-se a uma máquina na rede a nível global, entre outras informações.
- **Camada de Transporte**: Seja o protocolo de transporte UDP ou TCP, temos as portas de origem e destino que identificam um par de processos se comunicando, representando o uso de uma aplicação na rede, além de outras informações.

Tendo acesso aos dados da camada IP, podemos ter controle sobre o tráfego entre grupos de computadores quaisquer na Internet. Se fizermos o mesmo com relação à

camada de transporte, teremos também a informação relativa à aplicação que gerou tal tráfego.

Sendo assim, um monitor que observe as camadas de rede e de transporte pode contabilizar o tráfego entre quaisquer grupos de computadores na Internet, estejam eles na mesma rede física ou não, e discriminar o tráfego de acordo com a aplicação utilizada.

2.2 Domínios Gerenciáveis

Definimos um domínio como um par de conjuntos de elementos de rede entre os quais se quer monitorar o tráfego.

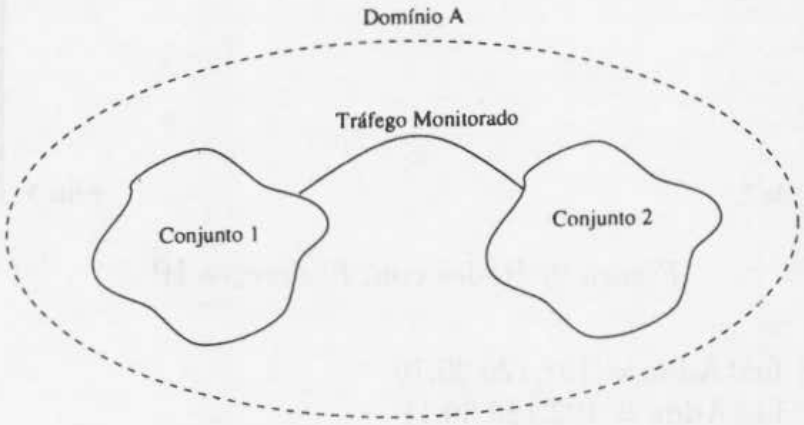

Figura 1: Estrutura de um Domínio

Na figura 1 vemos dois conjuntos de elementos de rede formando um domínio. Os conjuntos de domínios gerenciáveis seguem a seguinte regra de formação:

Seja um conjunto definido por dois endereços IP, **firstAddr** e **lastAddr**. Pertencem ao conjunto todos os elementos de rede com endereços IP, **ipAddr**, tal que:

$$\textbf{firstAddr} =< \textbf{ipAddr} =< \textbf{lastAddr}$$

Definindo-se conjuntos desta forma, pode-se monitorar o tráfego tanto entre dois elementos de rede como entre duas redes ou sub-redes. Convém relembrar que um endereço TCP-IP [Com91, LR93] é formado por um par (netid, hostid), portanto, podemos sempre englobar todos os elementos de uma rede em conjuntos que formam domínios gerenciáveis.

Podemos ver na figura 2, três redes interligadas estando os elementos de rede com os respectivos endereços IP. Vejamos alguns possíveis domínios gerenciáveis para este caso:

1. Domínio formado por dois elementos de rede de uma mesma rede física, por exemplo, domínio = $(A1);(B1)$. Neste caso, temos:

   Conjunto 1: firstAddr = 192.120.30.10
   lastAddr = 192.120.30.10
   Conjunto 2: firstAddr = 192.120.30.12
   lastAddr = 192.120.30.12

2. Domínio formado por grupos de elementos de rede em redes físicas distintas, por exemplo, domínio = $(A1, C1);(A3, B3)$. Neste caso, temos:

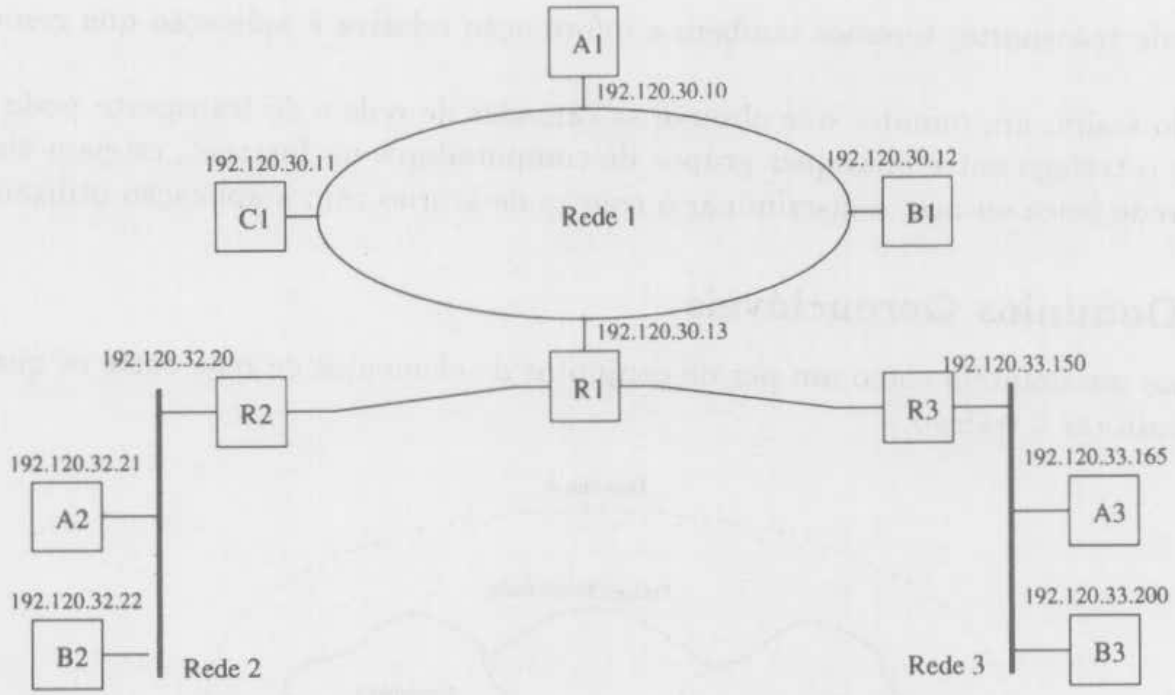

Figura 2: Redes com Endereços IP

Conjunto 1: firstAddr = 192.120.30.10
lastAddr = 192.120.30.11
Conjunto 2: firstAddr = 192.120.33.165
lastAddr = 192.120.33.200

2.3 Base de Informações de Gerenciamento (MIB)

Vamos definir a estrutura de dados que dá suporte ao modelo de gerenciamento de aplicações [MR90]. Como vimos, a base de dados de gerenciamento é o que, de fato, define as informações que serão gerenciadas e, conseqüentemente, o escopo das operações que poderão ser realizadas. Somente através de leituras e escritas o controle e a monitoração do elemento de rede são realizados, sendo a MIB (Management Information Base) assim, de importância primordial no modelo.

Sendo o nosso modelo privado, vamos usar um nó filho do **enterprise** [Ros91] como raiz da sub-árvore, e vamos rotulá-lo de **apTraffic**.

Precisamos de uma estrutura de dados que armazene os domínios que estão sendo gerenciados e, para cada domínio, as aplicações cujo tráfego se deseja medir.

Vamos usar duas variáveis para informações do estado geral do agente:

- **apMode**: indica se o agente está em monitoração (1) ou não (2).
- **apPeriod**: indica a duração, em segundos, do intervalo entre atualizações da MIB.

Usaremos duas tabelas para armazenar informações sobre domínio e tráfego. A primeira, **apDomTable**, armazena os domínios a serem monitorados, tendo a seguinte estrutura:

- **apDomIndice**: inteiro identificando um domínio em particular.

- **apFirstAddr1**: menor endereço IP do primeiro conjunto do domínio.
- **apLastAddr1**: maior endereço IP do primeiro conjunto do domínio.
- **apFirstAddr2**: menor endereço IP do segundo conjunto do domínio.
- **apLastAddr2**: maior endereço IP do segundo conjunto do domínio.
- **apDomStatus**: estado corrente da linha da tabela: válida (1) e inválida (2).

A segunda tabela armazena os dados relativos ao tráfego acumulado, discriminado por aplicações, e deve ser atualizada na freqüência determinada pela variável **apPeriod**. Chamamos esta tabela de **apDataTable**. As suas linhas têm as seguintes colunas:

- **apDataIndice**: identifica o domínio a monitorar, relacionando-se com a tabela **apDomTable** através do **apDomIndice**.
- **apDataProt**: código do protocolo de transporte usado pela aplicação a ser monitorada.
- **apDataAplic**: código da porta usada pela aplicação a ser monitorada.
- **apDataBytesEnv**: número de bytes enviados do primeiro conjunto ao segundo conjunto do domínio para uma dada aplicação.
- **apDataDatEnv**: número de datagramas enviados do primeiro conjunto ao segundo conjunto do domínio para uma dada aplicação.
- **apDataBytesRec**: número de bytes recebidos pelo primeiro conjunto do segundo conjunto do domínio para uma dada aplicação.
- **apDataDatRec**: número de datagramas recebidos pelo primeiro conjunto do segundo conjunto do domínio para uma dada aplicação.
- **apDataStatus**: estado corrente da linha da tabela: válida (1) e inválida (2).

A figura 3 mostra a estrutura da base de dados proposta para o gerenciamento do tráfego de aplicações. Como exemplo, vamos mostrar como seria a base de dados de um agente que estivesse gerenciando o tráfego da rede da figura 2 para os dois domínios lá descritos, sendo que em cada domínio estaríamos interessados em medir o tráfego devido às seguintes aplicações:

- **Domínio 1:** FTP (protocolo de transporte TCP - 6 e porta 21) e TELNET (protocolo de transporte TCP - 6 e porta 23).
- **Domínio 2:** SMTP (protocolo de transporte TCP - 6 e porta 25).

Supondo que a medição não tivesse sido iniciada e que o agente estivesse programado para acumular dados a cada 10 minutos, isto é, 600 segundos, teríamos os seguintes dados na MIB:

apMode = 2;
apPeriod = 600;

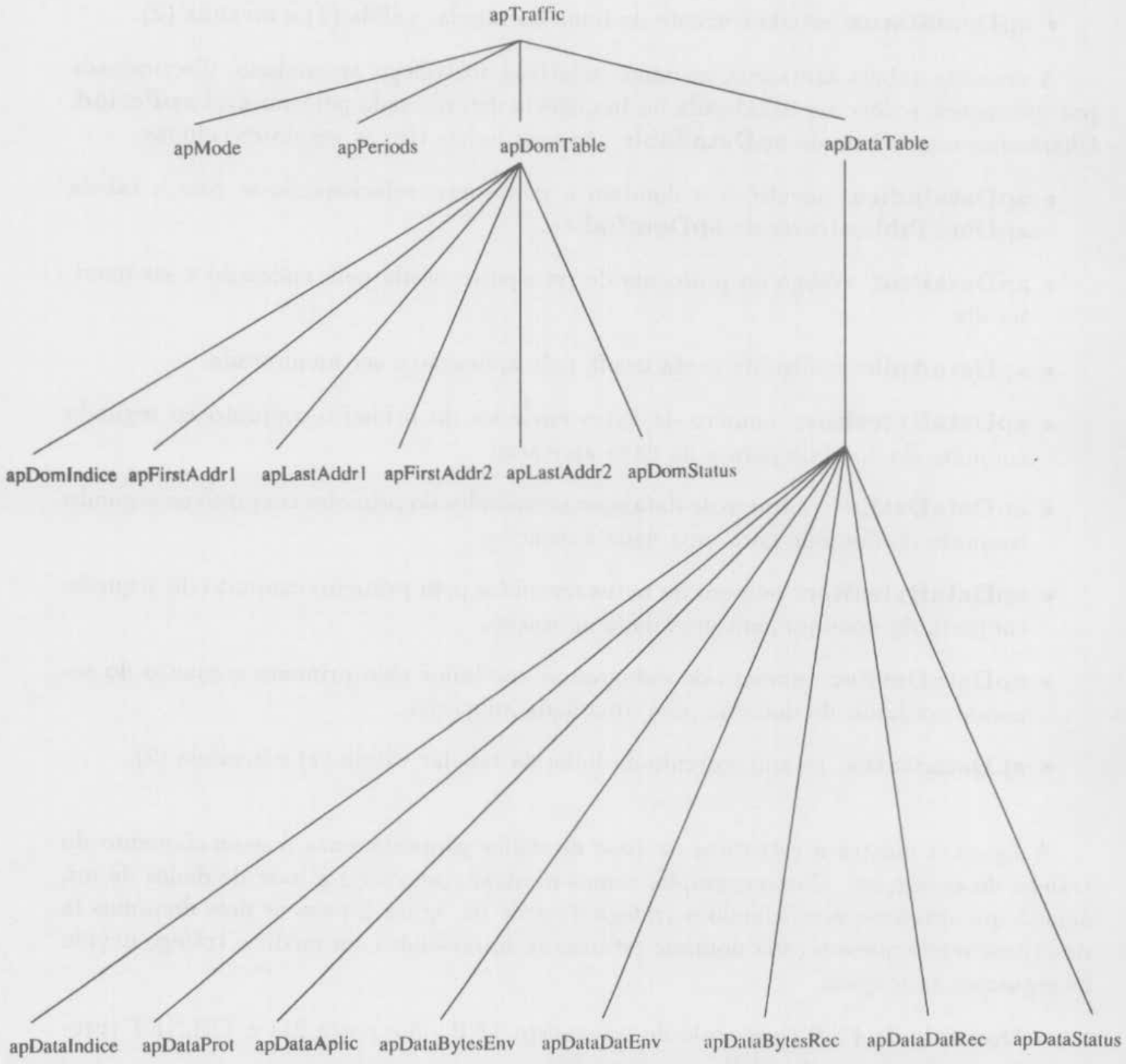

Figura 3: Sub-árvore da MIB apTraffic

DomIndice	FirstAddr1	LastAddr1	FirstAddr2	LastAddr2	DomStatus
1	192.120.30.10	192.120.30.10	192.120.30.12	192.120.30.12	1
2	192.120.30.10	192.120.30.11	192.120.33.165	192.120.33.200	1

apDomTable

Indice	Prot	Aplic	BytesEnv	DatEnv	BytesRec	DatRec	Status
1	6	21	0	0	0	0	1
1	6	23	0	0	0	0	1
1	127	127	0	0	0	0	1
2	6	25	0	0	0	0	1
2	127	127	0	0	0	0	1

apDataTable

Obs: as linhas cujos valores do *apDataProt* e *apDataAplic* são ambos iguais a 127 são usadas para acumular o tráfego devido às outras aplicações somadas.

Como mencionado, a MIB é atualizada com a periodicidade dada por **apPeriod**. Durante cada período, o tráfego é acumulado em variáveis temporárias, e no final do período o acumulado é escrito na variável da MIB (pelo agente, como veremos adiante). Sendo assim, a MIB contém sempre valores relativos ao tráfego acumulado no período anterior, somente mudando de valor ao fim de cada período. Supondo um cenário hipotético, vamos acompanhar o processo de acumulação de dados para o domínio 1, protocolo de transporte 6 e protocolo de aplicação 23. Na figura 4, cada seta representa um datagrama recebido ou enviado e o número corresponde ao tamanho em bytes. Note-se que na figura 4, admite-se que o valor de **apPeriod** é 600, equivalente a um intervalo de 10 minutos.

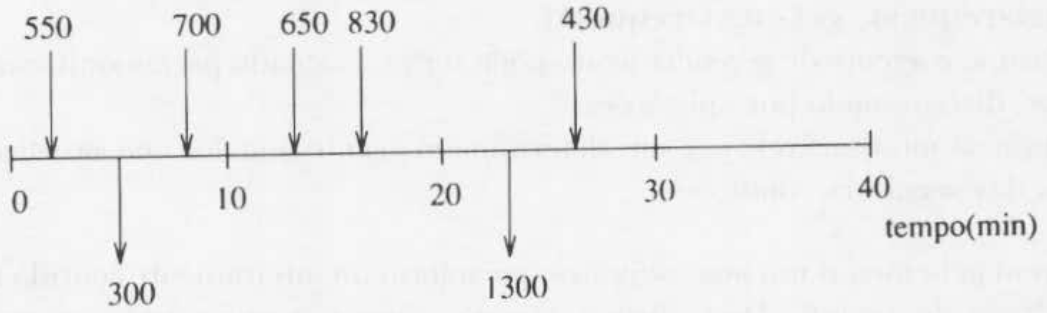

Figura 4: Tráfego entre grupos de um domínio

- **No início da medição:**

Indice	Prot	Aplic	BytesEnv	DatEnv	BytesRec	DatRec	Status
1	6	23	0	0	0	0	1

- **Depois da atualização feita aos 10 minutos:**

Indice	Prot	Aplic	BytesEnv	DatEnv	BytesRec	DatRec	Status
1	6	23	300	1	1250	2	1

- **Depois da atualização feita aos 20 minutos:**

Indice	Prot	Aplic	BytesEnv	DatEnv	BytesRec	DatRec	Status
1	6	23	0	0	1180	2	1

- **Depois da atualização feita aos 30 minutos:**

Indice	Prot	Aplic	BytesEnv	DatEnv	BytesRec	DatRec	Status
1	6	23	1300	1	130	1	1

2.4 Agente de Gerenciamento

O agente de gerenciamento tem como objetivos a coleta de dados (preenchimento da MIB) e o atendimento das requisições do(s) gerente(s) autorizado(s).

O agente realiza a coleta de dados analisando todos os pacotes que circulam pela rede a qual está conectado através de uma interface de rede em modo promíscuo.

O agente examina os datagramas, compara os endereços IP de origem e destino com os endereços dos domínios da tabela **apDomTable** e as portas de origem e destino com as aplicações classificadas na tabela **apDataTable** e acumula o tráfego correspondente em variáveis temporárias. Além disso, com a periodicidade de **apPeriod** e no final dos intervalos, o agente escreve na MIB o tráfego acumulado desde a atualização anterior.

Quando o pacote é uma requisição SNMP, o agente decodifica a mensagem, que chega no formato ASN.1, realiza a tarefa requerida e devolve uma mensagem de resposta. As tarefas requeridas podem ser: escrever um valor na MIB (**set-request**) ou ler um valor da MIB (**get-request, get-next-request**).

Desta forma, o agente de gerenciamento pode ser programado para monitorar o tráfego de domínios, discriminado por aplicações.

São domínios monitoráveis por um determinado agente aqueles que satisfizerem pelo menos uma das seguintes condições:

- Tiverem pelo menos um dos conjuntos que o formam inteiramente contido na mesma rede física do agente. Desta forma, como o agente examina todos os pacotes que circulam na rede, terá acesso a todos os pacotes enviados ou recebidos pelo dito conjunto.

- A rede à qual está ligado o agente seja passagem obrigatória para o tráfego entre os conjuntos que formam o domínio, assim, o agente examinará todos os pacotes que circularem entre os conjuntos.

Na figura 5, são exemplos de domínios monitoráveis pelo agente de gerenciamento localizado na rede 2:

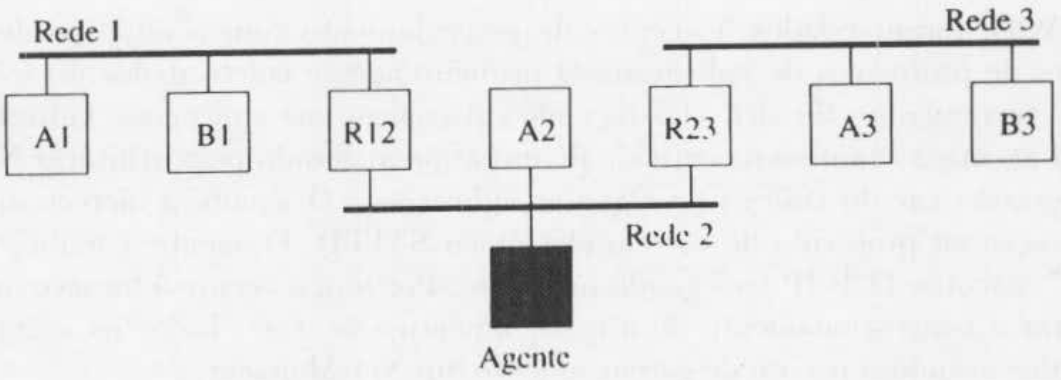

Figura 5: Rede com um Agente de Gerenciamento

- (A2)-(R23);
- (A1, B1)-(A3, B3);
- (R12, A2)-(A3, B3);

Não são monitoráveis:

- (A1)-(B1);
- (B1, R12, A2)-(A1, B1);
- (A1)-(R12);

Obs: podemos observar que no último caso de domínio não monitorável, apesar do elemento de rede R12 pertencer à rede 2, comunica-se com o elemento de rede A1 sem utilizar a interface com a rede 2, portanto o agente de gerenciamento não tem acesso ao tráfego gerado entre eles.

2.5 Gerente de Rede

O gerente tem como objetivo programar agentes a ele submetidos para que estes monitorem o tráfego entre domínios discriminado por aplicações, ler a MIB do agente periodicamente para recolher os dados sobre o tráfego do intervalo anterior e apresentar os dados para o usuário. O gerente comunica-se com o agente através das primitivas SNMP **get-request**, **get-next-request** e **set-request**.

Enviando requisições para escrever na tabela **apDomTable**, classificam-se os domínios a serem monitorados, na tabela **apDataTable**, as aplicações que se quer medir para cada domínio. Através da escrita nas variáveis **apMode** e **apPeriod** controla-se o estado do agente e a duração do intervalo entre atualizações da MIB.

Utilizou-se um software gerente que realiza as requisições necessárias e tem ferramentas para exibição de gráficos.

2.6 Trabalhos Relacionados

Existem alguns trabalhos na área de gerenciamento de aplicações, contudo, com enfoques diferentes do esquema proposto.

Em [RW94] foram criados 5 agentes de gerenciamento com o objetivo de fornecer informações de protocolos de aplicação. O primeiro agente coleta dados do tráfego global de um barramento *ethernet*, classificando tal tráfego nas aplicações **Telnet**, **FTP** e **SMTP**. O agente 2 examina o conteúdo de um arquivo gerado pelo utilitário **Netwatch** e fornece estatísticas do tráfego de algumas aplicações. O agente 3 oferece um serviço de confirmação ao protocolo de correio eletrônico **SMTP**. O agente 4 exibe, de forma "amigável", pacotes TCP-IP trafegando pela rede. Por fim, o agente 5 fornece meios para se monitorar o congestionamento de um barramento *ethernet*. Todos os agentes foram desenvolvidos usando o pacote de gerenciamento SunNet Manager.

Há em [Kil94b] um trabalho definindo uma MIB para o gerenciamento de aplicações de uma maneira geral. Os mesmos autores, em [Kil91a], falam sobre o gerenciamento de correio eletrônico. Define-se uma porção de uma MIB compatível com os protocolos de gerenciamento da Internet (SNMP e SNMPv2), que permite a monitoração de Agentes de Transferência de Mensagens (MTA) presentes nos roteadores.

Em [TdS94] apresenta-se um modelo de gerenciamento do protocolo **MHS X.400** (correio eletrônico). Foram analisados os requisitos para a gerencia do correio eletrônico e realizada uma modelagem, incluindo-se a definição de uma MIB, que cumprisse tais requisitos. Na ocasião da apresentação do trabalho, um protótipo estava sendo implementado.

Em [Cic94] tem-se o desenvolvimento de um agente SNMP para plataformas DOS, tendo sido implementada toda a MIB II, a MIB RMON e alguns objetos adicionais que possibilitam o gerenciamento de protocolos de aplicação como o SNMP, por exemplo.

Em [Car94], [CM91a] e [CM91b] apresenta-se um modelo de gerenciamento do protocolo FTP baseado em domínios. Foi criada uma MIB sub-dividida em quatro grupos que poderiam ser utilizados de acordo com as necessidades de gerenciamento. As informações disponíveis são, entre outras, número de conexões ftp abertas, arquivos mais transferidos. A implementação do modelo foi realizada utilizando o ISODE, sendo construído um sub-agente comunicando-se com agentes SNMP usando o protocolo SMUX.

Os trabalhos [Car91], [CM91a] e [CM91b] foram os precursores do esquema apresentado. Deles extraíram-se as idéias de gerenciar aplicações e coletar dados direto da rede, lá apresentado como sugestão. O agente 1 de [RW91] apresenta algumas semelhanças com o esquema de gerenciamento de tráfego por aplicações aqui exposto, não permitindo, no entanto, a definição de domínios de gerenciamento nem a livre escolha da aplicação a gerenciar. [Kil94a] e [TdS91] apresentam esquemas para o gerenciamento de uma determinada aplicação, no caso o correio eletrônico, com um nível maior de detalhes. [Cic94] mostra o gerenciamento de aplicações como validação do seu trabalho, não sendo o tópico principal.

3 Implementação

Vamos então observar os aspectos da implementação do modelo de gerenciamento de aplicações proposto.

O agente de gerenciamento, cuja função é de coletar os dados da MIB e comunicar-se através do protocolo de gerenciamento SNMP, foi desenvolvido usando-se um IBM-PC 486, com sistema operacional DOS, conectado a uma rede padrão Ethernet através de uma placa de rede IBM compatível com o padrão NE2000. O software gerente utilizado foi o SunNet Manager, rodando em estações SUN.

3.1 Base de Informações de Gerenciamento

Foi criada uma estrutura de dados para armazenar a MIB no agente. Cada objeto tem os seguintes campos:

- **name:** nome da variável SNMP (string);
- **id:** identificador de objeto (vetor de inteiros);
- **id_len:** comprimento do identificador de objeto (inteiro);
- **syntax:** código da sintaxe associada ao objeto (byte);
- **access:** regra de acesso ao objeto; pode ser somente leitura - RO, somente escrita - WO ou leitura e escrita - RW (string);
- **num_value:** conteúdo numérico da variável da MIB, se for o caso (inteiro longo);
- **char_value:** conteúdo caractere da variável da MIB, se for o caso (string);
- **acumulator:** acumula valores temporários do objeto.

A MIB é construída criando-se uma lista de tais registros. Quando se trata de uma variável da MIB numérica, o campo **num_value** é utilizado, caso contrário, o campo **char_value** é utilizado.

Algumas funções que manipulam a lista de objetos, ou seja, a MIB, foram implementadas, entre elas: mib-fill (adiciona um objeto à MIB), mib-delete (extrái um objeto da MIB), mib-point (aponta para um objeto da MIB). A lista é manipulada de forma que esteja sempre ordenada pelo campo **id**.

3.2 Agente de Gerenciamento

O software que implementa o agente é constituído das seguintes partes (ver figura 6):

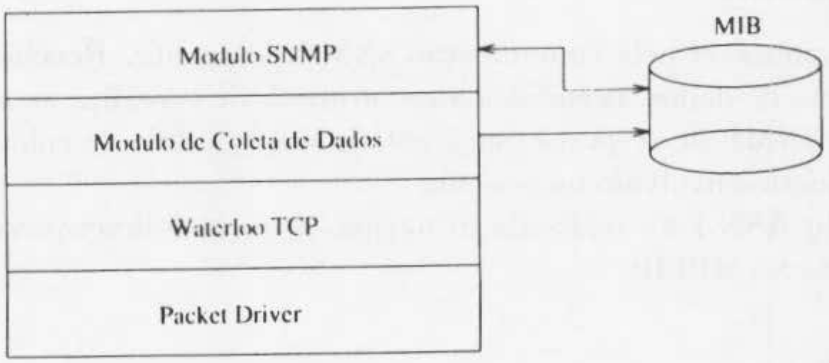

Figura 6: Agente de Gerenciamento

Vejamos, em detalhe, cada um dos módulos.

3.2.1 Packet Driver

O packet driver [Rom89] serve de interface entre o hardware de rede e o sistema operacional do micro e fica residente em memória. É de programação simples e permite acesso aos pacotes de rede no nível de enlace.

3.2.2 Waterloo TCP

O conjunto de programas Waterloo TCP [Eng91] é uma implementação completa do TCP-IP para DOS. Nós utilizamos apenas alguns módulos do Waterloo TCP: os que lidam com o enfileiramento de pacotes e acesso às rotinas do packet driver.

3.2.3 Coleta de Dados

Módulo desenvolvido no projeto, cuja função é receber *frames* tipo Ethernet do módulo Waterloo TCP (wattcp), fazer as atualizações necessárias na MIB referentes à aplicações que utilizem o protocolo de rede IP e passar para o módulo SNMP os pacotes correspondentes. Além disso, recebe os pacotes SNMP de resposta, monta os cabeçalhos UDP, IP e Ethernet e envia pela rede usando o wattcp.

Os passos referentes à coleta de dados são:

Para cada pacote enviado na rede, armazenam-se os endereços IP de origem e destino, as portas de origem e destino e o tamanho do datagrama. Com esses dados em mãos, verifica-se se o datagrama corresponde ao tráfego entre algum (ou alguns) dos domínios e incrementam-se as variáveis temporárias (**acumulator**) correspondentes às aplicações representadas pela porta de origem ou destino, conforme o datagrama seja enviado pelo servidor ou pelo cliente, respectivamente.

Quando se trata de um datagrama enviado por um gerente SNMP para o agente em questão, este módulo passa o PDU SNMP para o módulo SNMP e espera pelo PDU SNMP de resposta. Quando o módulo SNMP envia ao módulo de Coleta de Dados o PDU SNMP de resposta, este monta os cabeçalhos UDP, IP e Ethernet e envia para a rede utilizando o módulo wattcp.

Com a periodicidade dada pela variável **apPeriod**, o módulo de coleta de dados copia o campo **acumulator** para o campo **num_value**, e faz **acumulator** igual a zero. Note-se que toda requisição SNMP feita ao agente irá interagir com o campo **num_value**, no caso de variáveis da MIB numéricas.

3.2.4 Módulo SNMP

Este módulo é responsável pela comunicação SNMP do agente. Recebe o pacote SNMP do módulo de coleta de dados, decodifica a estrutura ASN.1, realiza as ações requisitadas e monta o pacote SNMP de resposta que é entregue ao módulo de coleta de dados. Este módulo também foi desenvolvido no projeto.

A decodificação ASN.1 foi realizada utilizando-se rotinas desenvolvidas no projeto e algumas funções da SNMPLIB.

3.3 Gerente

Como já mencionado, o software gerente utilizado foi o **SunNet Manager** versão 2.2 [Sun93a, Sun93b], executado em estações SUN.

O SunNet Manager compreende uma série de ferramentas que podem ser usadas para gerenciar redes. O software engloba um aplicação gerente e vários agentes de gerenciamento. A comunicação entre o gerente e os agentes é através do RPC (Remote Procedure Call). Um dos agentes fornecidos pela SUN é o **snmp-mibII**, que pode ser usado como agente *proxy* para a comunicação com a agentes que seguem o protocolo SNMP (ver figura 7).

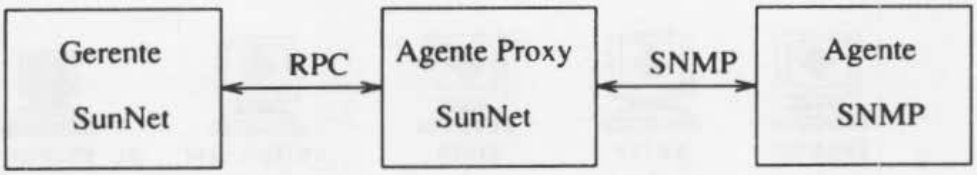

Figura 7: Agente *Proxy* do SunNet Manager

Desta forma, estabelecemos a comunicação entre a aplicação de gerenciamento e o agente desenvolvido para PC.

Como o modelo proposto para o gerenciamento do tráfego de aplicações apresenta uma extensão da MIB SNMP, para que o SunNet Manager tivesse conhecimento das variáveis a serem gerenciadas, a nova MIB, descrita no formato ASN.1, foi compilada através de uma ferramenta fornecida no pacote, o **mib2schema**, gerando um arquivo que indica ao SunNet Manager quais as novas variáveis e suas sintaxes.

Isto feito, podem ser usadas todas as facilidades de requisições e gráficos presentes no SunNet Manager.

Para monitorar o tráfego de aplicações usando o SunNet Manager, seguiram-se os seguintes passos:

1. Definem-se os domínios a serem monitorados e escrevem-se os valores a eles correspondentes na tabela **apDomTable** do agente;
2. Definem-se as aplicações a monitorar para cada domínio e escrevem-se os valores na tabela **apDataTable** do agente;
3. Define-se a duração, em segundos, dos intervalos de monitoração e escreve-se na variável **apPeriod** do agente;
4. Define-se o envio de requisições periódicas, periodicidade igual à definida na variável **apPeriods**, que lêem os valores da tabela **apDataTable** e guardam em um arquivo de *log*. O número de intervalos a serem monitorados também deve ser definido;
5. Terminado o processo de monitoração, tem-se um arquivo de *log* contendo dados relativos ao tráfego de aplicações dos domínios classificados na tabela **apDomTable**. As ferramentas *browser* e *grapher* do SunNet Manager são utilizadas para a visualização do arquivo de *log* e para traçar os gráficos.

3.4 Resultados

Foram realizadas algumas medidas do tráfego de aplicações na rede do Departamento de Ciência da Computação (DCC) da Unicamp. A topologia da rede está apresentada na figura 8.

A máquina Jaguari é utilizada como servidora de disco. A máquina Tiete é a servidora de correio eletrônico e a roteadora de saída do DCC. Uma informação útil para o gerenciamento desta rede é relacionada à quantidade do tráfego do *backbone*, que é a sub-rede **143.106.7.0**, devido a correio eletrônico para fora do departamento e quanto é devido a serviço de disco da Jaguari.

O agente de gerenciamento foi colocado no *backbone* na rede do DCC. O software foi executado no PC cujo endereço Internet é 143.106.7.20.

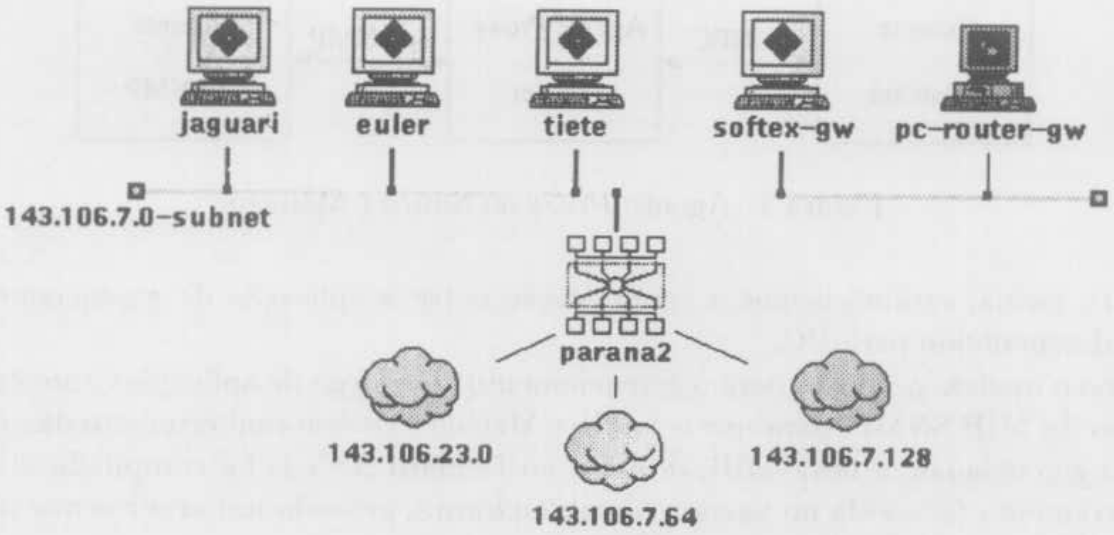

Figura 8: Rede do DCC - Unicamp

Para o caso da Jaguari, definimos um domínio como sendo formado, por um lado, pelas sub-redes roteadas pela máquina Paraná (**143.106.7.64, 143.106.7.128 e 143.106.23.0**) e por outro pela máquina Jaguari. Configuramos a duração do intervalo de monitoração para 1200 segundos, ou seja, 20 minutos, e efetuamos medições das 9:00h às 12:00, totalizando, portanto, 9 intervalos de 20 minutos. Além disso, monitoramos especificamente a aplicação RPC (porta 111) sob UDP (protocolo 17), já que sabe-se que o maior usuário do RPC sob UDP é o NFS (Network File System).

A figura 9 apresenta um gráfico mostrando o tráfego, em número de bytes, das três redes acima mencionadas para a máquina Jaguari devido a RPC (bytes enviados e recebidos).

Para o caso da Tiete, definimos um domínio como sendo formado, por um lado, pelas três redes mencionadas, e por outro, pela máquina Tiete. Configuramos a duração do intervalo para 600 segundos, ou seja, 10 minutos, e efetuamos medições das 13:00h às 14:00h, totalizando, portanto, 6 intervalos de 10 minutos. Além disso, monitoramos especificamente a aplicação SMTP (porta 25) sob TCP (protocolo 6).

Na figura 10, o gráfico mostra o tráfego do domínio descrito, devido ao correio eletrônico (SMTP) para bytes enviados e bytes recebidos.

Algumas medidas foram realizadas com o intuito de caracterizar o tráfego interno ao departamento e entre o departamento e o exterior. A aplicação com maior tráfego devido a comunicação entre máquinas do departamento foi, com grande diferença para as demais, o serviço de disco (NFS, porta 111, transporte 17). Foram monitoradas também a transferência de arquivos (FTP, porta 21, transporte 6) e correio eletrônico (SMTP, porta 25, transporte 6) (ver figura 11).

Para o caso do tráfego entre máquinas do DCC e máquinas não pertencentes ao DCC, as aplicações mais utilizadoras da rede foram: *rlogin* (portas 23 e 513, transporte 6), www (porta 80, transporte 6) e serviço de disco (porta 111, transporte 17) (ver figura 12).

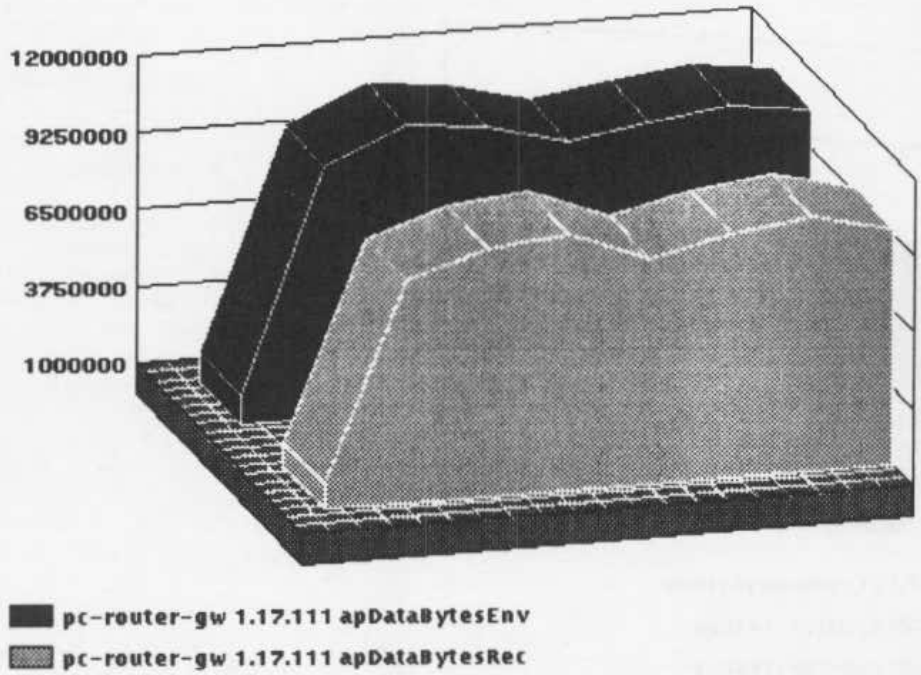

Figura 9: Tráfego de RPC para a máquina Jaguari

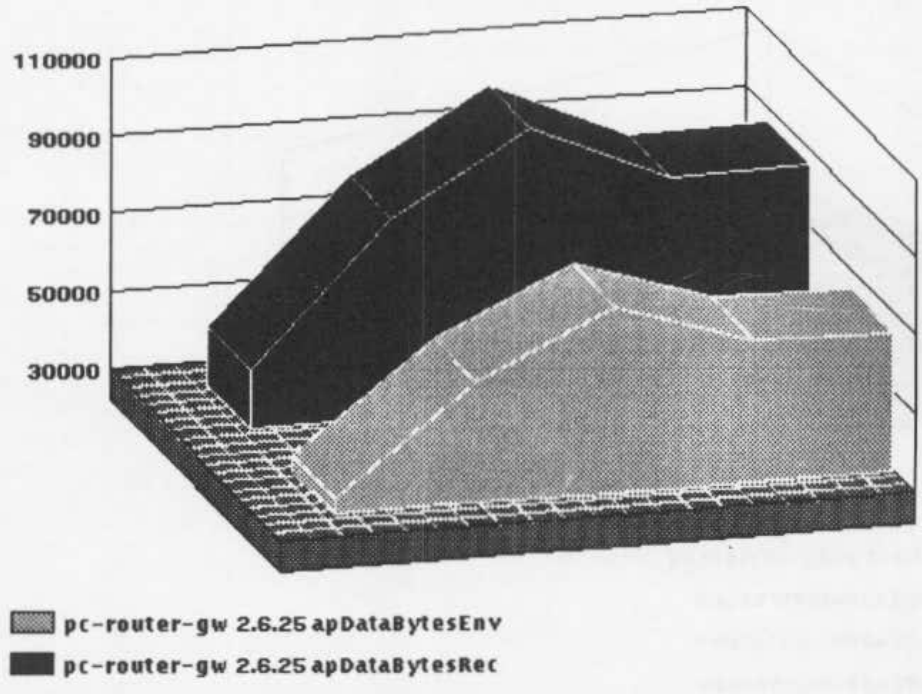

Figura 10: Tráfego de SMTP para a máquina Tiete

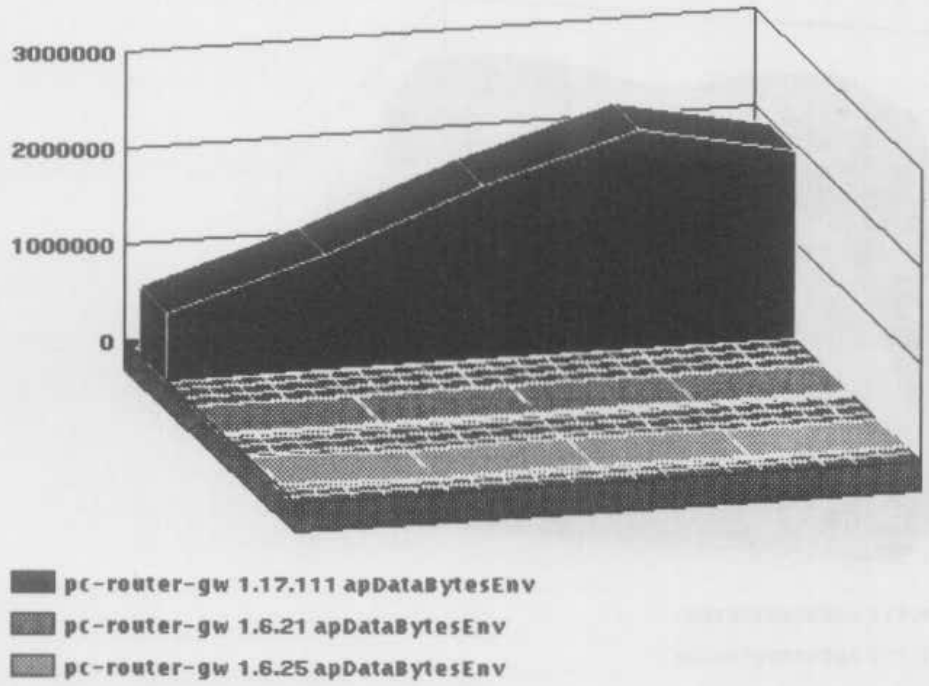

Figura 11: Tráfego interno ao departamento

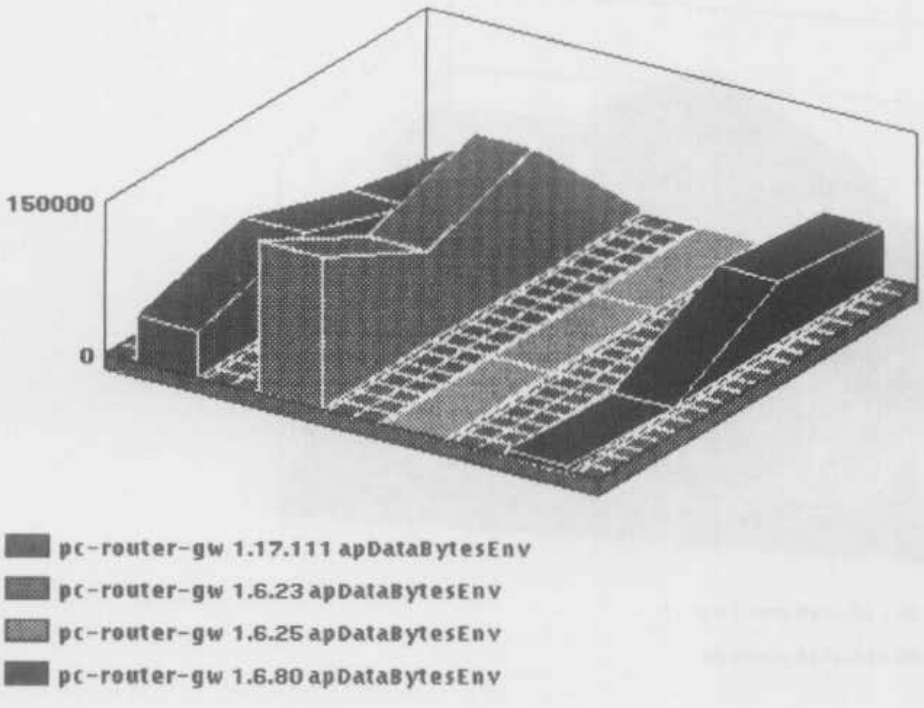

Figura 12: Tráfego para o exterior do departamento

3.5 Ferramentas de Apoio

Duas ferramentas foram desenvolvidas para complementar as informações disponíveis através do esquema proposto. Ambas foram desenvolvidas em ambiente UNIX (SunOs) e realizam operações no arquivo de *log* gerado pelo SunNet Manager com as informações do tráfego de aplicações nos domínios.

A primeira foi denominada de **percent** e fornece, para toda uma série de medidas em um arquivo de *log*, o percentual devido a cada aplicação. Por exemplo, aplicando-se a ferramenta **percent** sobre o arquivo de *log* contendo os dados relativos à figura 12, obtivemos:

```
Percentual por aplicacao:

1.6.25          1.38%
1.6.80         17.12%
1.17.111       26.60%
1.6.23         54.90%
```

A outra ferramenta desenvolvida foi denominada de **combine**. Esta ferramenta tem a finalidade de combinar os dados de domínios de um mesmo arquivo de *log* e gerar um outro arquivo de *log* contendo dados combinados. Por exemplo, para se obter os dados relativos à figura 12, monitoramos as aplicações citadas acima para os seguintes domínios:

DomIndice	FirstAddr1	LastAddr1	FirstAddr2	LastAddr2	DomStatus
1	143.106.7.0	143.106.7.159	0.0.0.0	143.106.6.255	1
2	143.106.7.0	143.106.7.159	143.106.7.160	255.255.255.255	1
3	143.106.7.0	143.106.7.159	143.106.23.0	143.106.23.31	1
4	143.106.23.0	143.106.23.31	0.0.0.0	143.106.22.255	1
5	143.106.23.0	143.106.23.31	143.106.23.32	255.255.255.255	1
6	143.106.23.0	143.106.23.31	143.106.7.0	143.106.7.159	1

apDomTable

Fazendo-se:

```
combine <arquivo-log1> +1 +2 -3 +4 +5 -6 <arquivo-log2>
```

obtém-se em **arquivo-log2** as informações presentes na figura 12.

Uma observação importante é que a ferramenta **combine** somente pode ser utilizada para combinar dados de domínios cujas aplicações monitoradas são iguais.

Através destes dados temos uma informação interessante a respeito do tráfego na rede do DCC-Unicamp, e podemos ter idéia das informações que o esquema proposto pode oferecer. Pode-se verificar como é possível monitorar o tráfego entre grupos diversos de computadores, para aplicações diferentes, e com intervalos de monitoração adequados a cada caso.

Foi observado, após as monitorações iniciais, uma certa dificuldade em localizar quais as aplicações (caracterizadas pelo código da porta do servidor) utilizavam mais a rede. Para que se pudesse minorar tal problema, foi realizada uma modificação na MIB de forma a monitorar o tráfego, conjuntamente, de uma faixa de códigos de porta. Desta forma, foi possível localizar as faixas responsáveis pelo maior tráfego e, através de aproximações sucessivas, encontrar, de fato, as aplicações que mais carregavam a rede.

4 Conclusão

Como visto, o SNMP fornece uma base de dados de gerenciamento padrão e possibilita a extensão do escopo de gerenciamento através da definição de novos objetos de gerenciamento por empresas ou grupos de pesquisa. Desta forma, à medida que os usuários de computadores solicitam aumentos de capacidade de gerenciamento os fornecedores de tecnologia podem atendê-los.

Dentre as áreas que se vem solicitando um aumento de capacidade de gerenciamento temos o gerenciamento de aplicações. As aplicações são a verdadeira finalidade da rede. As pessoas usam a rede para corresponder-se eletronicamente, navegar através do *mosaic*, *gopher*, entre outras aplicações. Assim, é natural indagar se a rede suporta o tráfego gerado por uma determinada aplicação ou se um dado servidor está, ou não, bem localizado, entre outras dúvidas.

Este trabalho contribui com o gerenciamento de aplicações propondo um esquema para o gerenciamento de tráfego. Por ser o conjunto de protocolos mais difundido, o esquema foi idealizado especificamente para redes TCP-IP.

Foi proposto um esquema de gerenciamento em domínios, onde cada domínio é composto por um par de conjuntos de elementos de rede entre os quais se quer monitorar o tráfego. Propôs-se também uma base de dados de gerenciamento que possibilitou ao gerente de rede a inclusão de domínios e a definição das aplicações, para cada domínio, que se quisesse monitorar. O agente de gerenciamento coletaria todos os pacotes da rede à qual estivesse conectado, podendo, assim, monitorar o tráfego de qualquer domínio para o qual sua rede fosse passagem obrigatória.

Algumas vantagens do esquema de gerenciamento proposto são: seguir o modelo SNMP, que o torna compatível com qualquer aplicativo de gerenciamento que siga tal padrão, a possibilidade de monitorar qualquer aplicação em redes TCP-IP e, de acordo com a localização do agente, a possibilidade de se gerenciar o tráfego para uma variedade bastante grande de configurações de domínios.

O gerenciamento de aplicações permite uma discriminação de tráfego da rede, fornecendo inclusive a noção dos elementos originadores e destinatários de tráfego, fornecendo uma boa base para se fazer uma melhor alocação de servidores na rede, auxiliando também na decisão sobre a formação de sub-redes.

Com o intuito de validar o esquema de gerenciamento, um protótipo foi implementado. O agente de gerenciamento foi desenvolvido em um computador IBM-PC, na linguagem *Borland C*. O aplicativo de gerenciamento utilizado foi o SunNet Manager, onde compilou-se a nova MIB. Obtiveram-se resultados úteis demonstrando a capacidade do esquema.

Ao optar-se por implementar o esquema de gerenciamento de tráfego usando um PC levou-se em conta que é um hardware barato e amplamente difundido. Além disso, foram utilizados softwares de domínio público (*packet driver* e *waterloo TCP*) existentes para PCs que forneceram a base para a comunicação de rede. Por fim, ressalta-se que a implementação do agente em um PC não interfere com o tráfego da rede e nem no desempenho de qualquer computador que estiver sendo utilizado, deixando a coleta de dados de gerenciamento imperceptível aos usuários.

Referências

[Car94] J. A. Carrilho. Um modelo para o gerenciamento do protocolo ftp baseado em

dominios. Master's thesis, DCC-Unicamp, dezembro de 1994.

[CFSD90] J. D. Case, M. Fedor, M. L. Schoffstall, e C. Davin. Simple network management protocol (SNMP). Internet Request for Comments 1157, Maio de 1990. Obsoletes RFC 1098.

[Cic94] R. Cicilini. Desenvolvimento de um agente snmp para plataformas rodando dos. Master's thesis, ICMSC-USP, maio de 1994.

[CM94a] J. A. Carrilho e E. R. M. Madeira. A scheme for ftp management. *Proc. of INET'94/JENC5- The Annual Conference of Internet Society and Joint-European Network Conference*, junho de 1994.

[CM94b] J. A. Carrilho e E. R. M. Madeira. Um esquema para o gerenciamento do protocolo ftp baseado em dominios. *12o. Simposio Brasileiro de Redes de Computadores*, maio de 1994.

[Com91] D. E. Comer. *Internetworking with TCP-IP, vol. 1.* Prentice Hall International, segunda edição, 1991.

[Eng91] Erick Engelke. *Watterloo TCP.* University of Waterloo, 1991. Disponivel por ftp em ftp.unicamp.br.

[Kil94a] S. Kille. Mail monitoring mib. Internet Request for Comments 1566, janeiro de 1994. Editor N. Freed.

[Kil94b] S. Kille. Network services monitoring mib. Internet Request for Comments 1565, janeiro de 1994. Editor N. Freed.

[LR93] D. Lynch e M. Rose, editores. *Internet system handbook.* Addisson-Wesley, 1993.

[MR90] K. McCloghrie e M. T. Rose. Management information base for network management of TCP/IP-based internets. Internet Request for Comments 1156, Maio de 1990. Obsoletes RFC 1066.

[Rom89] John Romkey. *PC/TCP Packet Driver Specification.* FTP Software Inc., 1.09 edição, setembro de 1989.

[Ros91] M. T. Rose. *The simple book - an introduction to management of TCP-IP internets.* Prentice-Hall, 1991.

[RW94] M. A. Rocha e C. B. Westphall. Gerencia de redes de computadores atraves de novos agentes. *Anais do 12o. Simposio de Redes de Computadores*, maio de 1994.

[Sun93a] Sun Microsystems Inc. *SunNet Manager Reference Guide*, 1993. Versao 2.2.

[Sun93b] Sun Microsystems Inc. *SunNet Manager User Guide*, 1993. Versao 2.2.

[TdS94] L. Tarouco e A.C. Benso da Silva. Uma proposta para gerencia de correio eletronico. *12o. Simposio Brasileiro de Redes de Computadores*, maio de 1994.