

Representação Dinâmica de Objetos em Sistemas de Gerenciamento de Redes

Cybelle Suemi Oda
Edson dos Santos Moreira

Departamento de Ciência da Computação e Estatística
Instituto de Ciências Matemáticas de São Carlos
Universidade de São Paulo

Caixa Postal 668 - 13560-970 - São Carlos- SP

cybelle@hq.rnp.br
edsmorei@icmssc.sc.usp.br

Resumo

Este trabalho apresenta um modelo de representação dinâmica de objetos em sistemas de gerenciamento de redes de computadores, utilizando o protocolo SNMP e recursos do Unix para obter informações sobre os objetos das redes dentro do domínio de gerenciamento. A partir dessas informações é gerada a topologia das redes, bem como o desenho da composição das mesmas.

Um protótipo desse modelo foi implementado em ambiente SunOS 4.3, utilizando o pacote de domínio público 4BSD/ISODE SNMP, o qual implementa o protocolo SNMP, e seguindo o padrão XWindow na interface gráfica com o usuário, através do gerador de interfaces Devguide. Utilizou-se primitivas gráficas do XWindow no desenho da topologia e das redes.

Abstract

This work presents a model for dynamic representation of objects in network management systems, using the SNMP protocol and Unix functions to get relevant information about network objects which are in the management domain. Using these information, the network topology and the network components are drawn, giving to the network manager a spatial notion about the whole domain.

A prototype was implemented in order to validate the model. It was used a public domain packet called 4.3 BSD/ISODE SNMP, in SunOS 4.3 environment, which implements the SNMP protocol. To implement the user interface and to draw the network's topology and components, it was used the interface generator DevGuide and graphical primitives, accordingly to the XWindow standard.

1. Introdução

O gerenciamento de redes de computadores vem se tornando cada vez mais necessário e complexo à medida em que ocorre a expansão das redes de longa distância e aumenta a utilização das redes a nível local, através da descentralização do processamento e da interconexão de computadores e equipamentos em ambientes acadêmicos e comerciais.

A atividade de gerenciamento envolve uma série de tarefas que têm como objetivo prover o funcionamento adequado das redes. São exemplos:

- planejar o crescimento das redes;
- monitorar as redes com o intuito de detectar falhas;
- medir a performance da rede, determinando pontos de estrangulamento;
- adotar mecanismos de segurança;
- adotar mecanismos para recuperação rápida de falhas; e
- manter uma base de dados histórica sobre a configuração da rede.

A ISO (*International Organization for Standardization*) e IAB (*Internet Activity Board*) desenvolveram arquiteturas padrões, para o gerenciamento de redes ISO/OSI e Internet, as quais tornaram-se mais conhecidas pelo seus protocolos de gerenciamento CMIP (*Common Management Information Protocol*) [Br93] [Kl88] e SNMP (*Simple Network Management Protocol*) [Ca90] [Ro91], respectivamente. Através da adoção destas arquiteturas, as ferramentas de gerenciamento, que antes eram vinculadas a redes proprietárias, passaram a gerenciar redes heterogêneas constituídas de diferentes tecnologias, e integradas em um único sistema de gerenciamento.

Uma característica desejável nesses sistemas de gerenciamento é a capacidade de apresentar a topologia das redes, dentro do domínio de gerenciamento, mostrando as conexões através de *gateways*. Além de fornecer uma visão espacial das redes, o desenho da topologia pode ser utilizada para apresentar outras informações ao gerente da rede e dinamizar a interação entre sistema e usuário. Por exemplo, indicação do estado de hosts e *gateways* através da mudança de cor dos símbolos que os representam; e apresentação de informações sobre um determinado host, gateway ou rede (tabela de roteamento, dados estatísticos, etc) selecionando-o através do desenho da topologia.

2. Fontes de Informações sobre Objetos de Redes

Um ponto importante a ser considerado é o levantamento de fontes de informações a partir das quais podem ser obtidos dados para se determinar a topologia e composição das redes gerenciadas. Dentro do ambiente Unix, os arquivos *hosts* e *networks* fornecem informações a respeito de *hosts* (nome, *alias* e endereços Internet) e redes (nome, *alias* e endereço Internet). Considerando que a utilização do protocolo de gerenciamento SNMP vem crescendo dentro dos sistemas de gerenciamento, a base de informações de gerenciamento, MIB-II [Ro91], apresenta uma série de objetos, por exemplo *netmasks* (máscaras de subrede) e endereços IP de interfaces de *gateways*, que permitem a obtenção de dados necessários para determinar a topologia das redes. A seguir são descritos alguns desses objetos, pertencentes aos grupos *System*, *Interface* e *IP*.

Grupo System

- *sysDescr*: descreve o nome e versão do tipo de hardware do sistema, sistema operacional, e software de rede;
- *sysUpTime*: indica a quanto tempo o sistema foi reinicializado, em centésimos de segundos.

Grupo Interface

- *ifNumber*: número da interface de rede através do qual o sistema envia e recebe datagramas IP;
- *ifType*: tipo da interface (ethernet-csma cd, iso-88025-token ring, fddi, rfc877-x25, T1-carrier, hyperchannel, etc);
- *ifPhysAddress*: endereço físico da interface;

Grupo IP

- *ipForwarding*: indica se a máquina está agindo como um *gateway* IP, ou um *host*;

Objetos relacionado à tabela de endereços IP:

- *ipAdEntAddr*: endereço IP;
- *ipAdEntNetmask*: a máscara de subrede associada ao endereço IP;

Objetos relacionados à tabela de roteamento IP, considerando que cada entrada na tabela corresponde a uma rota:

- *ipRouteDest*: o endereço IP destino para a rota;
- *ipRouteIfIndex*: o valor do índice que identifica a interface local através do qual o *next hop* para esta rota poderia ser alcançado
- *ipRouteNextHop*: endereço IP do próximo *gateway* para a rota;

3. Modelo para a Representação Dinâmica

O modelo para a representação dinâmica envolve basicamente duas partes: uma é a obtenção de informações e o armazenamento em estruturas de dados projetadas de forma a facilitar a representação topológica e das redes. A outra corresponde ao desenho da topologia e dos componentes das redes. A definição das estruturas de dados utilizadas e os módulos correspondendo as duas partes são descritos a seguir.

3.1 Estruturas de Dados

Considerando a natureza dinâmica do ambiente de redes onde *hosts* são adicionados e removidos, novas subredes são criadas e interconexões são realizadas, as estruturas de dados que armazenam as informações das redes devem refletir essa flexibilidade. Outro fator a ser considerado no projeto das estruturas de dados é o armazenamento de informação de posicionamento gráfico do objeto de rede, no caso o ponto (x,y) superior esquerdo do quadrilátero (retângulo ou quadrado) que representa o desenho do *gateway*, *host* ou rede na tela. A partir desse ponto e da altura e largura do quadrilátero é possível delimitar uma área ao redor do desenho que capture como entrada um *double-clicking* disparando uma função de apresentação de um outro desenho ou informações, por exemplo uma foto digitalizada.

As informações das redes e subredes a partir das quais é elaborado o desenho das redes e de seus componentes são armazenadas basicamente em três tipos de listas: lista de *hosts*, lista de *gateways* e lista de redes.

Lista de Hosts

A lista de *hosts* traz armazenados o nome, endereço Internet, endereço físico, localização geográfica, tipo de equipamento (PCs, SUNs, etc), posição (x,y) do canto superior esquerdo da área onde está desenhado, e indicador sobre a existência de agente SNMP sobre cada *host* presente no domínio de gerenciamento.

Lista de Redes

A lista de redes, figura 1, armazena o nome da rede, endereço Internet, *netmask*, a tecnologia utilizada (TCP/IP, Novell, Lan Manager, etc), os *hosts* que pertencem à rede em questão, número de *hosts*, posição (x,y) do canto superior esquerdo do retângulo onde está desenhado, os *gateways* que fazem parte da rede e o número desses.

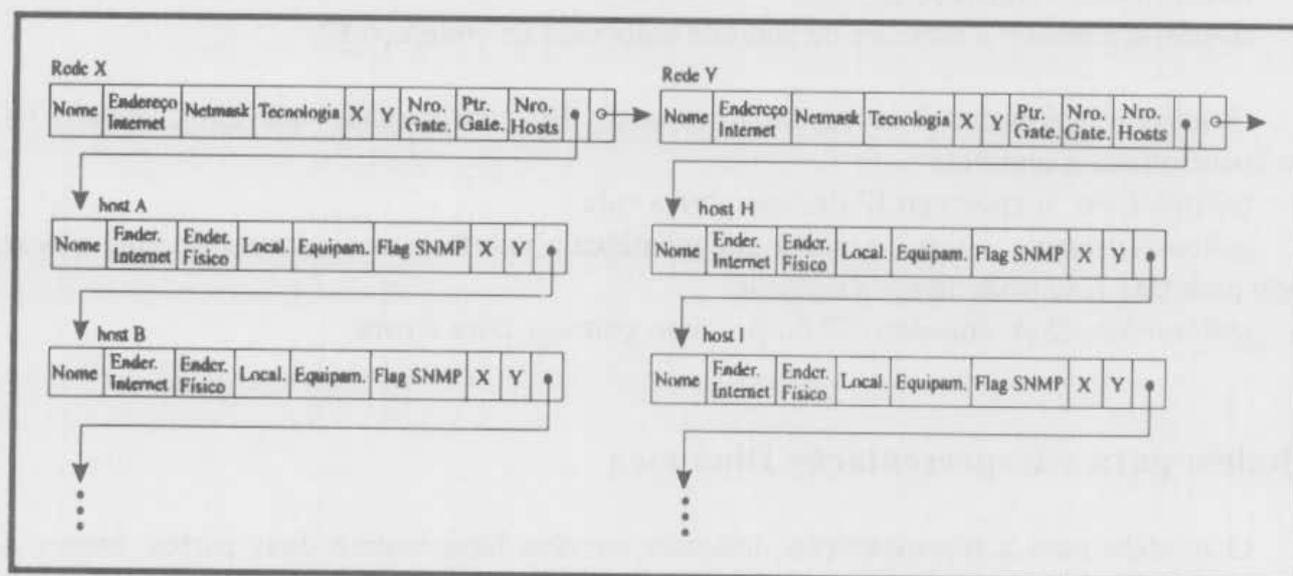
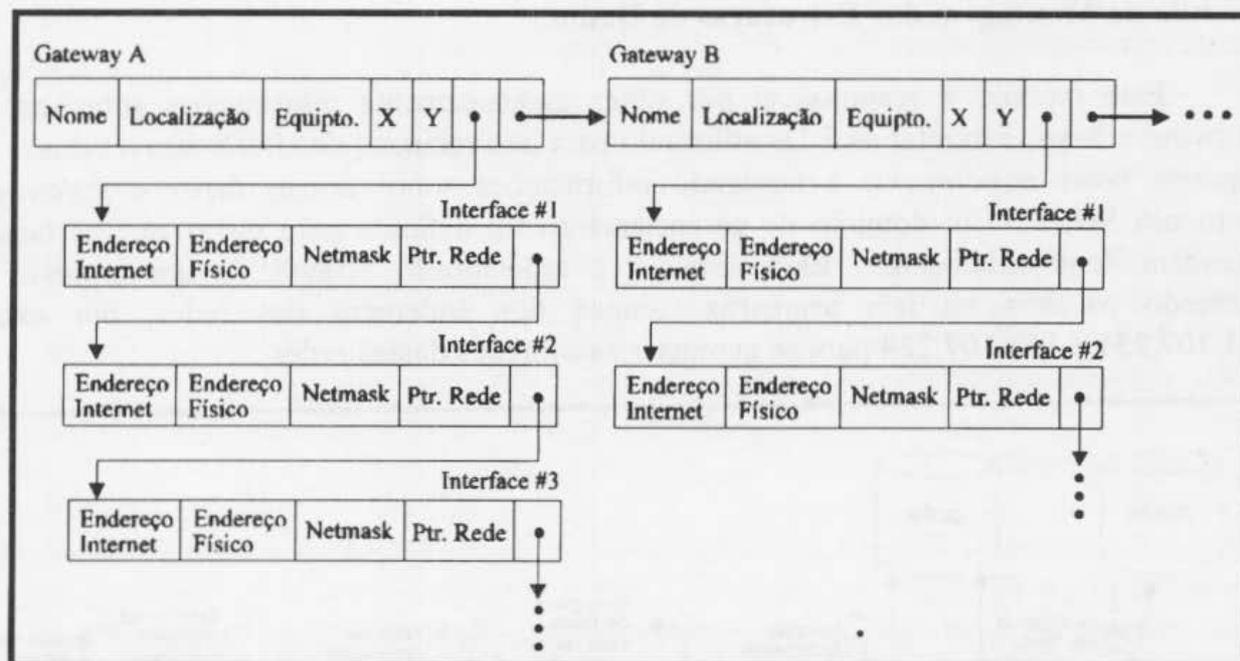


Figura 1 - Lista de redes juntamente com a lista de *hosts*.

Lista de Gateways

A lista de *gateways*, armazena as seguintes informações sobre um *gateway*: nome, localização geográfica, tipo de equipamento, informações sobre suas interfaces (endereço Internet, endereço físico, *netmask* e o ponteiro para o nó da rede na qual o *gateway* está conectado pela interface) e a posição (x,y) do canto superior esquerdo do quadrado onde está o desenho que o representa.

Como o número de interfaces de um *gateway* é variável, criou-se uma lista onde cada nó armazena as informações, descritas acima, de uma de suas interfaces. A figura 2 ilustra a lista de *gateways*, e sua relação com a lista de interfaces.

Figura 2 - Lista de *gateways*.

3.2 A Estrutura do Modelo

A estrutura do modelo compreende o módulo de montagem das estruturas de dados descritas anteriormente, o módulo gráfico responsável pelo desenho da topologia e das redes e o módulo de funções, bem como a relação entre eles. O esquema geral é apresentado na figura 3, e o detalhamento dos módulos descrito a seguir.

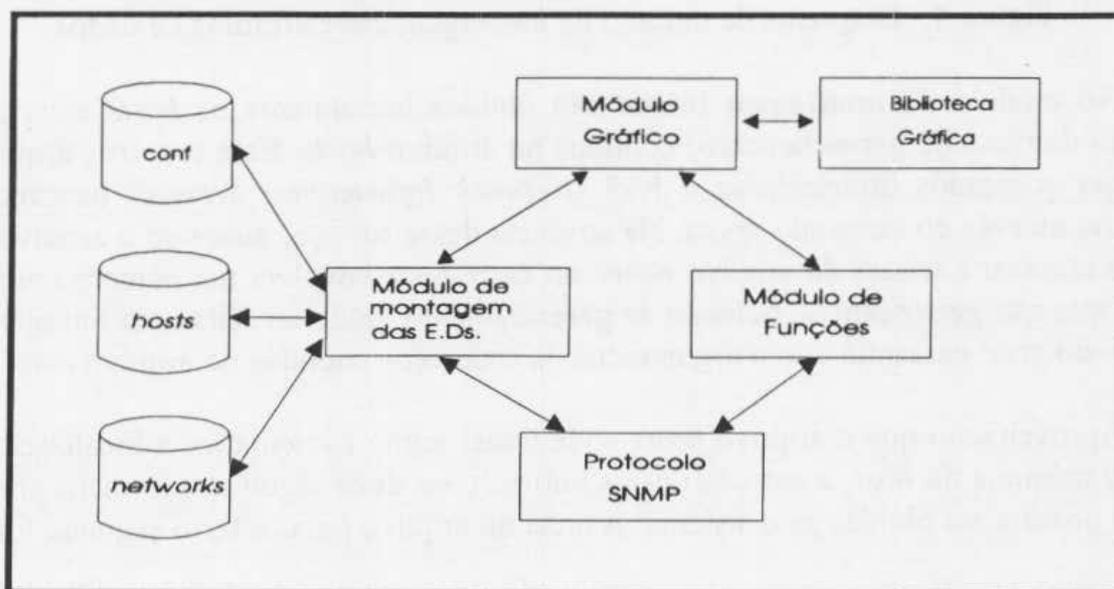


Figura 3 - Diagrama de Módulos do Modelo.

Módulo de Montagem das Estruturas de Dados

Este módulo é responsável por obter dinamicamente informações sobre as redes, *gateways* e *hosts*, e montar as E.Ds utilizando para isso recursos do UNIX através dos arquivos *hosts* e *networks*, e buscando informações sobre outros *hosts* e *gateways* via protocolo SNMP. Um domínio de gerenciamento foi definido para evitar que as buscas se estendam demasiadamente. Esse domínio é especificado através de um arquivo, *conf*, contendo os dois ou três primeiros campos dos endereços das redes, por exemplo, 143.107.231 e 143.107.224 para se gerenciar as subredes dessas redes.

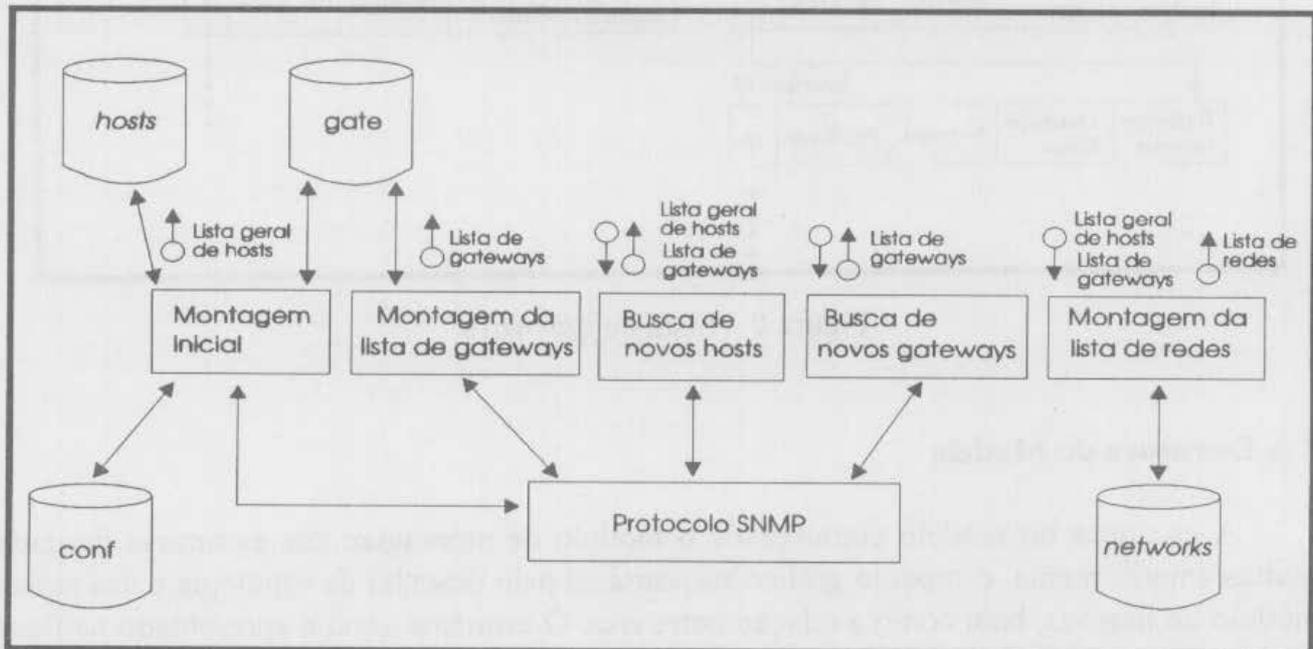


Figura 4.- Diagrama de módulo de montagem das estruturas de dados.

No módulo de **montagem inicial** são obtidos inicialmente os *hosts* e os *gateways*, dentro do domínio de gerenciamento, contidos no arquivo *hosts*. Esse e outros arquivos Unix podem ser acessados utilizando-se o NIS (*Network Information Service*) buscando-os em */var/etc* ou através do comando *ypcat*. Na ausência desse serviço, busca-se o arquivo em */etc*. A fim de otimizar a leitura do arquivo *hosts*, no caso desse envolver um número muito grande de *hosts* que não pertençam ao domínio de gerenciamento, pode ser feita uma filtragem através do comando *grep* passando como argumentos os endereços contidos no arquivo *conf*.

Aproveitando que o arquivo *hosts* pode trazer como comentários a localização física e o tipo de máquina do *host*, a entrada dessas informações deve seguir um formato pré-definido para que possam ser obtidas pelo sistema. A linha do arquivo passa a ter o seguinte formato:

<endereço Internet> <nome> <alias> #<tipo de máquina> (<localização>).

A presença do comentário continua sendo facultativo. Um exemplo de uma entrada do arquivo:

143.107.231.1 xavante #Sparc Station 2 (sala 27-j).

Para determinar se uma máquina definida em *hosts* é um *gateway* ou não, é necessário obter, via SNMP, o objeto *ipForwarding* cujo valor pode ser 1 (*gateway*) ou 2 (*host*). Caso esse objeto não possa ser acessado dentro de um período de tempo, a máquina é considerada um *host*, e assume-se que provavelmente não tenha implementado o agente SNMP. Os *hosts* obtidos a partir do arquivo *hosts* são inseridos na lista geral de *hosts*, a qual é implementada como uma lista de *hosts*, e os *gateways* armazenados em um arquivo (*gate*), para serem processados posteriormente.

O módulo de **montagem da lista de gateways** processa o arquivo *gate* que contém os *gateways* encontrados em *hosts*. Para cada *gateway* são obtidas informações sobre suas interfaces (endereço Internet, endereço físico e *netmask*) buscando-se os objetos *ipAdEntAddr*, *ifPhysAddress* e *ipAdEntNetMask*, através do protocolo SNMP. Os *gateways* são inseridos na lista de *gateways*.

Para suprir a deficiência do arquivo *hosts*, que não possui necessariamente todos os *hosts* das redes, o módulo **busca de novos hosts** percorre a lista de *gateways* e busca na tabela ARP os endereços Internet e físico de novos *hosts* que não estejam mencionados em *hosts*. Essa busca é feita através da aquisição dos objetos *ipNetToMediaNetAddress* e *ipNetToMediaPhysAddress*. Antes de inseri-los na lista geral de *hosts* é necessário verificar se são realmente *hosts* ou *gateways* utilizando o mesmo procedimento descrito no módulo de montagem inicial. Se forem *gateways*, busca-se informações sobre suas interfaces. Os *gateways* são inseridos na lista de *gateways*.

O módulo **busca de novos gateways** busca outros *gateways* através do *next hop* (endereço IP do *gateway* para o qual os pacotes devem ser enviados para se chegar à rede destino) da tabela de roteamento dos *gateways* contidos na lista de *gateways*. Essas informações são obtidas trazendo-se os objetos *ipRouteNextHop* via SNMP. Para cada *gateway* obtido, são buscadas informações sobre suas interfaces, caso tenha o agente SNMP implementado. Caso contrário é necessário verificar se não está na lista geral de *hosts* para evitar ambiguidades. O *gateway* é inserido na lista de *gateways*.

Uma vez obtidos os *hosts* e os *gateways* do domínio de gerenciamento, armazenados na lista de *gateways* e na lista geral de *hosts*, parte-se para a montagem da lista de redes realizada pelo módulo de **montagem da lista de redes**. Através das interfaces de rede dos *gateways* é possível determinar o endereço das redes nas quais os *gateways* estão conectados. Isto é possível a partir do endereço Internet e da *netmask* (máscara de sub-rede) associada a cada interface do *gateway*. Para cada endereço de rede obtém-se o nome da rede, se o endereço estiver contido no arquivo *networks*, e os *hosts* que a compõem buscando-os na lista geral de *hosts* e inserindo na lista de *hosts* da rede. A rede é inserida na lista de redes.

Para ilustrar como o endereço da rede é obtido a partir do endereço Internet e da *netmask*, considere como exemplo a interface de rede X com endereço Internet 143.107.231.66 e *netmask* 225.225.225.192. O número 192 em binário é 11000000, logo podem haver 4 subredes (143.107.231.0, 143.107.231.64, 143.107.231.128 e 143.107.231.192) com 62 *hosts* cada. O endereço 143.107.231.66 está na subrede 143.107.231.64, assim pode-se concluir que a rede na qual a interface X está conectada é a 143.107.231.64. Os *hosts* que fazem parte dessa rede são aqueles cujos endereços estão no intervalo de 143.107.231.65 a 143.107.231.126.

Módulo Gráfico

O módulo Gráfico é responsável pelo desenho do mapa das redes gerenciadas, figura 12, e dos componentes de uma determinada rede selecionada a partir do desenho do mapa das redes, figura 13.

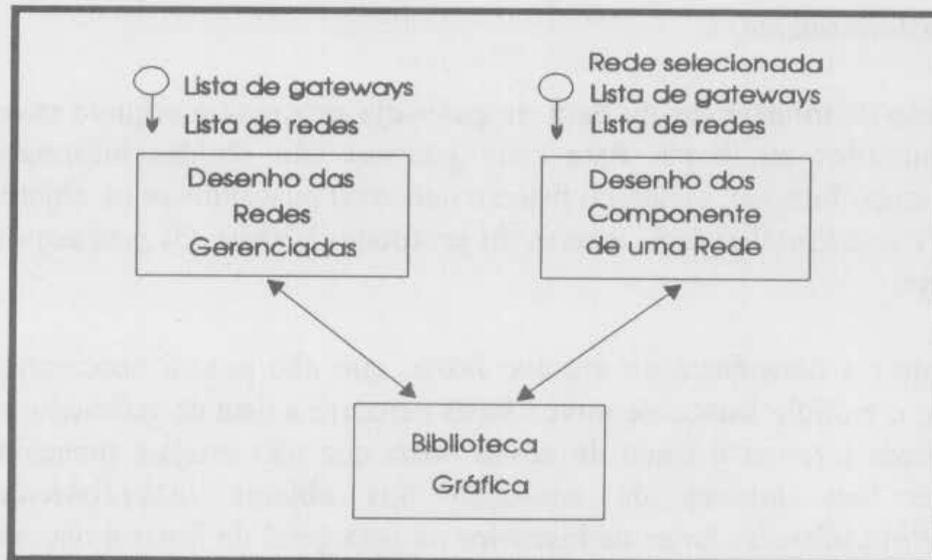


Figura 5 - Diagrama do módulo gráfico.

O módulo **desenho das redes gerenciadas** tem como dados de entrada a lista de *gateways* e a lista de redes. O desenho das redes é feito percorrendo-se a lista de *gateways*, e verificando cada uma das interfaces de cada *gateway*. Para cada rede desenhada, os seus campos *x* e *y* são atualizados com o valor do ponto (*x,y*) superior esquerdo da área na qual está desenhada. O controle das áreas desenhadas é feito através de uma matriz que representa a tela dividida em pequenas áreas dentro das quais são feitos os desenhos.

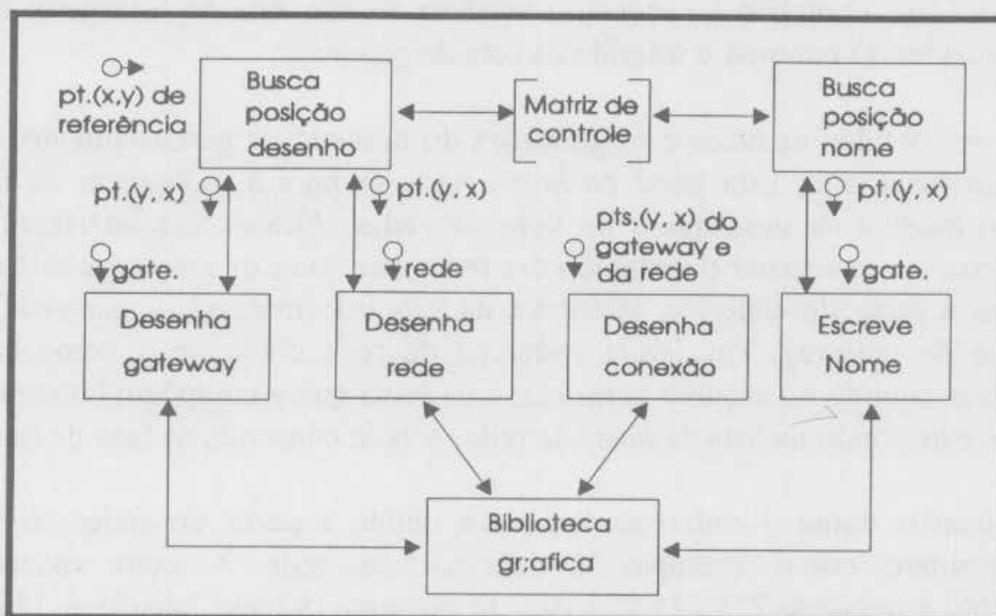


Figura 6 - Diagrama do módulo de desenho do mapa das redes gerenciadas.

De maneira geral, o algoritmo utilizado para se desenhar o mapa das redes gerenciadas é mostrado a seguir, e se utiliza de informações contidas na lista de *gateways* e na lista de redes. O módulo **busca posição desenho** é utilizado para se fazer a busca de uma posição

adequada na tela, tendo como entrada um ponto de referência. Na posição retornada é feito o desenho do *gateway* ou da rede, através do módulo **desenha gateway** e ou do módulo **desenha rede**. Toda vez que um desenho de um *gateway* ou de uma rede é feito, a posição do desenho é armazenada na estrutura que contém as informações dos *gateways* ou das redes.

Para cada gateway G1 da lista de gateways

Para cada interface do gateway G1

Busca outro gateway G2 que esteja na mesma rede que G1

Se G2 existir então

Se a rede já está desenhada então

Se G2 não está desenhado então

busca posição adequada na tela, em relação à rede, para desenhá-lo G2

desenha G2 na posição obtida

Fim se

desenha a conexão entre G2 e a rede

Senão

Se G2 já está desenhado então

busca posição adequada na tela, em relação a G2, para desenhá-lo rede,

desenha a rede na posição obtida

desenha a conexão entre G2 e a rede

Se G1 não está desenhado então

busca posição adequada na tela, em relação à rede, para desenhá-lo G1

desenha G1 na posição obtida

Fim se

Desenha a conexão entre G1 e a rede

Senão

Se G1 não está desenhado

busca uma posição adequada na tela, sem se considerar outra posição, para desenhá-lo G1

desenha G1 na posição obtida

busca posição adequada na tela, em relação a G1, para desenhá-lo rede,

desenha a rede na posição obtida

desenha a conexão entre G1 e a rede

busca posição adequada na tela, em relação a rede, para desenhá-lo G2,

desenha G2 na posição obtida

desenha a conexão entre G2 e a rede

Senão

busca posição adequada na tela, em relação a G1, para desenhá-lo rede,

desenha a rede na posição obtida

desenha a conexão entre G1 e a rede

busca posição adequada na tela, em relação à rede, para desenhá-lo G2,

desenha G2 na posição obtida

desenha a conexão entre G2 e a rede

Fim senão

Fim senão

Fim senão

Senão

Se G1 não está desenhado então

```

busca posição adequada na tela, sem se considerar outra posição, para desenhar
G1
desenha G1 na posição obtida
busca posição adequada na tela, em relação a G1, para desenhar a rede
desenha a rede na posição obtida
Senão
busca posição adequada na tela, em relação a G1, para desenhar a rede
desenha a rede na posição obtida
Fim senão
desenha a conexão entre G1 e a rede
Fim senão
Fim para
Fim para

```

Figura 7 - Algoritmo para desenhar o mapa das redes gerenciadas.

Os nomes dos *gateways* são escritos buscando-se também posições apropriadas através do módulo **busca posição nome**. As interfaces são marcadas para não se refazer desnecessariamente os desenhos. Esse processo é repetido até que todas as interfaces de todos os *gateways* tenham sido verificadas. Para cada área desenhada, a área correspondente na matriz é marcada.

O módulo **desenho dos componentes de uma rede** é bem mais simples, e tem por função desenhar os componentes (*hosts* e *gateways*) da rede selecionada, juntamente com seus nomes e endereços Internet. Uma vez selecionada a rede onde se deseja visualizar sua composição, busca-se a rede na lista de redes. A partir do número de *hosts* e de *gateways* conectados à rede, determina-se o número de dobras que o desenho, representando o meio físico, deve fazer (a figura 13 apresenta duas dobras) para acomodar todos os componentes na tela. Percorre-se a lista de *hosts* e *gateways* da sub-rede e desenha-os juntamente com o nome e endereço Internet, atualizando os seus campos x e y.

Módulo de Funções

O módulo de funções deve conter as funções de gerenciamento que venham a ser inseridas no modelo proposto por este artigo. Por exemplo, verificação do status de *hosts*, *gateways* e conexões, apresentação de gráficos estatísticos, apresentação de tabelas de roteamento, etc.

4. Implementação

Para validar o modelo, um protótipo foi implementado [Od94] utilizando o pacote 4BSD/ISODE SNMP, uma implementação do *framework* de gerenciamento para sistemas Berkeley Unix. Esse pacote inclui a implementação completa de um agente SNMP e ferramentas para a implementação de protótipos de aplicações de gerenciamento de redes. Esse pacote na verdade faz parte de um outro pacote de domínio público, o ISODE (*ISO Development Environment*), o qual contém um conjunto de bibliotecas e programas de aplicação que conjuntamente fornecem as camadas superiores do OSI. Adicionalmente fornece

ferramentas para a construção de protótipos de novas aplicações OSI [On89]. A versão 8.0 desse software pode ser obtida através de *anonymous FTP* no endereço *ftp.psi.com* (38.145.211.6) dentro do diretório */isode*. O nome do arquivo é *isode-8.tar.Z*.

Existem duas possibilidades de se acessar objetos remotos através do protocolo SNMP utilizando o pacote 4BSD/ISODE SNMP. A primeira consiste na utilização de primitivas em C para as operações *get* e *get-next*. Nesse pacote existe um programa de aplicação capaz de buscar objetos em MIBs remotas dado o endereço e a comunidade do agente SNMP, o *snmpi*. O código fonte desse programa faz uso dessas primitivas constituindo-se num modelo para a implementação da parte de comunicação SNMP, uma vez que o pacote não traz nenhum tipo de documentação sobre implementação de aplicações de gerenciamento utilizando essas primitivas.

A segunda opção trata-se da linguagem *Gawk SNMP-capable*. A linguagem *gawk* é a implementação GNU da linguagem *awk*, de onde advém o nome (GNU *awk*), com algumas extensões. O diretório *snmp/gawk-2.11* contido no pacote 4BSD/ISODE SNMP contém as modificações necessárias para torná-lo *SNMP-capable*, ou seja, capaz de buscar objetos em MIBs. Essa linguagem é muito utilizada na implementação de protótipos de funções de gerenciamento, uma vez que os valores de objetos da MIB são obtidos facilmente através de um simples *print* ou *printf*. A versão 2.11 da linguagem *gawk* pode ser obtida através de *anonymous FTP* no endereço *gumby.dsd.trw.com* (129.193.72.50), no subdiretório */pub/networking/isode/isode-7*. O nome do arquivo é *gawk-2.11.tar.Z*.

4.1 Implementação das Estruturas de Dados

A implementação das estruturas de dados especificadas no item 3.1 exigiu a adição dos campos *anhost* e *nroiface*, necessários para tornar mais eficiente a execução das funções. A definição em C das estruturas é apresentada a seguir.

```
/*Estrutura de dados que contem o endereco dos gateways conectados a uma rede*/
struct netgate {
    int x;           /*posicao (y,x) do gateway na tela de apresentacao
                    dos componentes de uma rede*/
    int y;
    struct gateway *ptrgate; /*ptr. para o gateway conectado a rede*/
    struct giface *ptriface; /*ptr. para a interface através do qual o gateway
                             esta conectado a rede*/
    struct netgate *proxng; /* ptr. para o próximo gateway conectado a rede*/
};
```

```
/* Estrutura de dados que contem as informacoes sobre as subnets */
struct subnet {
    char snetname [30]; /* nome da subrede */
    char snetaddr [16]; /* endereco Internet da subrede*/
    char snetmask [16]; /* netmask da subrede */
    char snetecnol [50]; /* tecnologia da subrede*/
    int nro;           /* nro. de hosts na subrede */
    int x;             /*posicao (y,x) da rede na tela de apresentacao
                    do mapa das redes*/
    int y;
    struct host *ptrhost; /* ponteiro para a lista de hosts*/
    int nrog;          /* numero de gateways conectados a rede*/
};
```

```

    struct netgate *ptrnetg;      /* ponteiro para a lista de gateways conectados
                                  a rede*/
    struct subnet *nextsnet;     /* ponteiro para a proxima subrede*/
};
/* Estrutura de dados que contem as informacoes sobre os hosts */
struct host {
    char hostname[30];          /* nome do host*/
    char hfisaddr[20];          /* endereco fisico do host*/
    char hintaddr[16];          /* endereco Internet do host*/
    char hglocal[30];           /* localizacao geografica do host*/
    char hhard [30];            /* tipo de hardware do host*/
    char snmp[2];               /* indica se o host tem agente SNMP*/
    int x;                      /* posicao (y,x) do host na tela de apresentacao
dos componentes de uma rede*/
    int y;
    struct host *nexthost;      /* ponteiro para o proximo host na lista*/
    struct host *anhost;       /* ponteiro para o host anterior na lista*/
};
/* Estrutura de dados que contem as informacoes sobre os gateways e sobre suas interfaces*/
struct giface {
    char g_fisaddr[20];         /* endereco fisico da interface de rede*/
    char g_intaddr[16];         /* endereco Internet da interface de rede*/
    char g_netmask[16];         /* netmask da interface de rede*/
    struct subnet *ptrsnet; /* ptr. para a subrede a qual a interface esta
conectada*/
    int mark;                   /* marca se a interface ja foi verificada no desenho
do mapa das rede*/
    struct giface *nextiface;   /* ponteiro para a proxima interface de rede do
gateway*/
};

struct gateway {
    char gatename[30];          /* nome do gateway*/
    char g_glocal[30];          /* localizacao geografica do gateway*/
    char g_hard [30];           /* tipo de maquina do gateway */
    int nroiface;               /* numero de interfaces do gateway*/
    int xg;                     /* posicao (y,x) do gateway na tela de
apresentacao do mapa das redes*/
    int yg;
    struct giface *ifaces;      /* ponteiro para a lista de interfaces do gateway */
    struct gateway *nextgate;   /* ponteiro para o proximo gateway*/
};

```

Figura 8 - Implementação em C das estruturas de dados.

4.2 Implementação dos Módulos

Devido à restrição de espaço são apresentados apenas os detalhes mais relevantes relacionados ao módulo de montagem das estruturas de dados e ao módulo gráfico.

Módulo de Montagem das Estruturas de Dados

Nesse item são apresentados basicamente alguns programas escritos em gawk que possibilitaram o acesso ao objetos da MIB no módulo referente à montagem das estruturas de

dados. Esses programas foram chamados através do comando *system* e os resultados direcionados para um arquivo para serem processados.

- Função que obtém as informações referentes às interfaces do *gateway*.

```
BEGIN {
    # gawk -f <arq> <end. agente>
    # passando o endereço Internet do host através de argv
    # Obtem o endereço Internet, a netmask eo endereço físico
    # da maquina onde esta o agente SNMP

    AGENT = ARGV[1];
    RETRIES = 1;
    TIMEOUT = 1;
    for (i in ifIndex) { # para cada uma das interfaces
        for (j in ipAdEntAddr) { #para cada endereço Internet da tabela IP
            if (ipAdEntIfIndex == ifIndex) {
                printf "%s %s %s\n",ipAdEntAddr, ipAdEntNetMask, ifPhysAddress;
                break;
            }
        }
    }
}
```

Figura 9 - Programa TGATE.

- Função que busca os *hosts* (endereço Internet e físico) presentes na tabela ARP.

```
BEGIN {
    # gawk -f <arq> <end. agente>
    # passando o endereço Internet do host através de argv
    # Obtem outros hosts através da tabela ARP do gateway que possui
    # o agente SNMP

    AGENT = ARGV[1];
    RETRIES = 1;
    TIMEOUT = 1;
    didone = 0;
        for (i in ipNetToMediaIfIndex) {# percorre a tabela ARP
            didone = 1;
            # Obtem o endereço Internet e físico do host
            printf "%s %s\n", ipNetToMediaNetAddress, ipNetToMediaPhysAddress
        }#for
}
```

Figura 10 - Programa THOSTS.

- Função que busca o next hop na tabela de roteamento.

```
BEGIN {
    # gawk -f <arq> <end. agente>
    # passando o endereço Internet do host através de argv
    # Obtem outros gateways através do ipRouteNextHop.
    AGENT = ARGV[1];
    RETRIES = 1;
    TIMEOUT = 1;
    j=0;
    for (i in ipRouteIfIndex) { # para cada entrada na tabela de roteamento
        gate = ipRouteNextHop; #obtem o endereço do next hop
        if ( gate != AGENT){
```

```

sai = 0;
for(k in a){ # nao pega gateways repetidos
  if (gate == a[k])
    sai = 1;
}
if (sai == 0){
  a[j] = gate;
  j++;
}
}
}#for
for (i in a){
  if (a[i] != "0.0.0.0")
    printf "%s\n", a[i];
}
}

```

Figura 11 - Programa TOGATE.

Módulo Gráfico

A interface com o usuário foi implementada utilizando-se o DevGuide [Su90a] e XView [He90], incluindo toda a parte de criação de janelas, menus, botões e outros itens de interface.

Na parte dos desenhos (topologia e apresentação das redes) utilizou-se as primitivas gráficas do XWindow [Jo89] para desenhar retas, retângulos, preencher polígonos, etc. As figuras 12 e 13 mostram as janelas com o desenho da topologia das redes e os seus componentes.

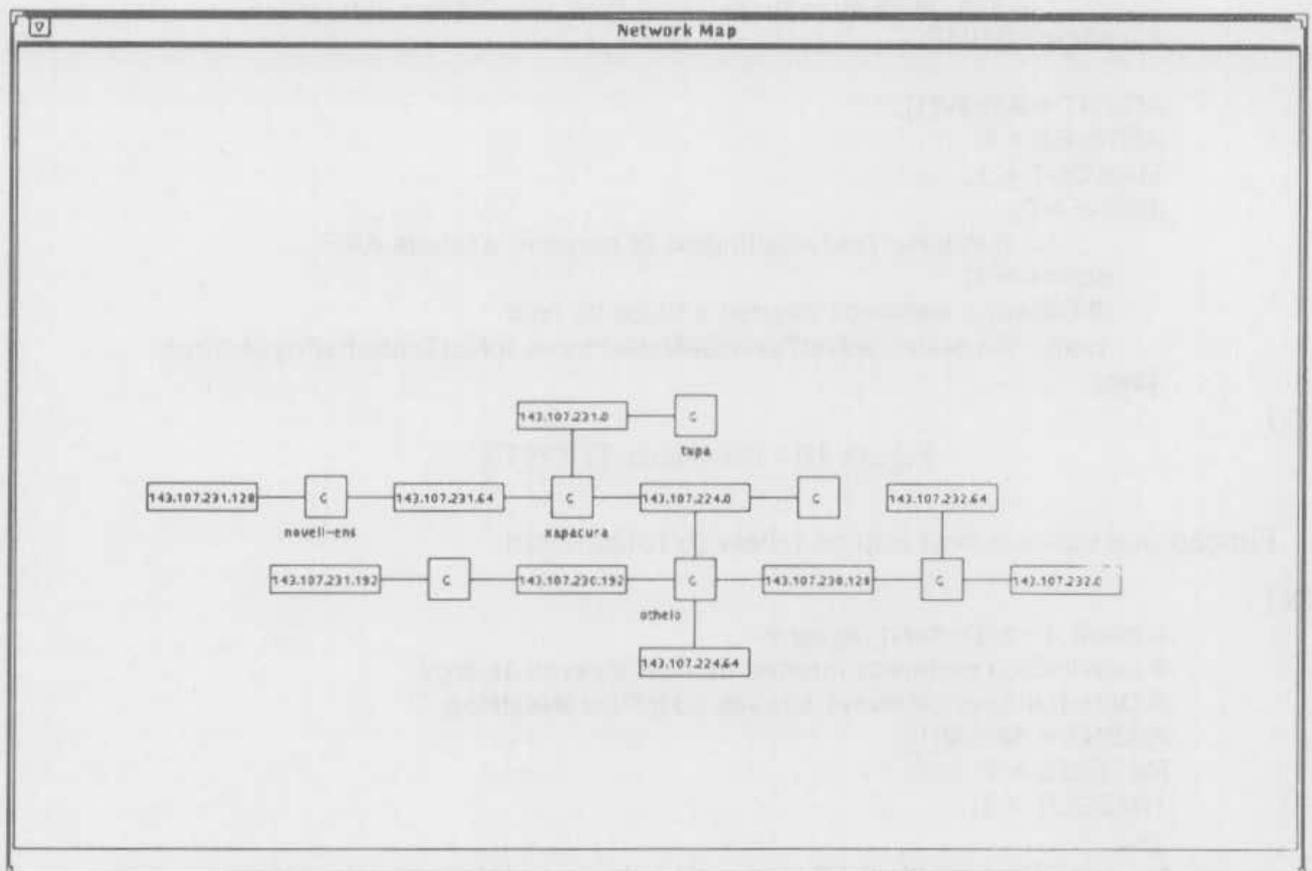


Figura 12-Janela de apresentação do mapa das redes gerenciadas.

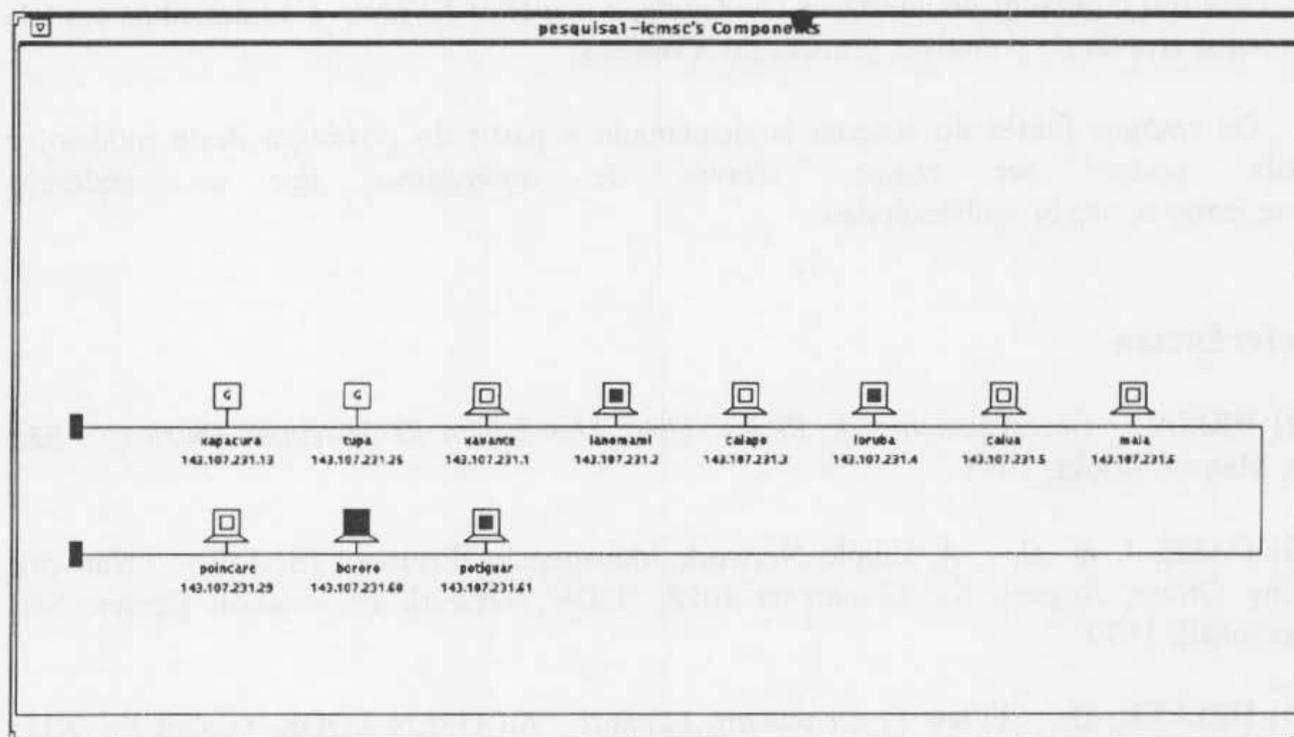


Figura 13 - Janela de apresentação dos componentes de uma rede.

5. Conclusão

A representação dinâmica da topologia das redes gerenciadas e de seus componentes em sistemas de gerenciamento é importante para apresentar ao gerente a disposição espacial das redes a qualquer momento. Adicionalmente, informações sobre objetos da rede podem ser apresentadas no próprio desenho facilitando a identificação por parte do gerente, por exemplo pintando de vermelho os *gateways* e *hosts* desligados, fazendo piscar o desenho da rede que está com problemas de comunicação, etc.

Para se obter uma boa representação, uma cuidadosa escolha das estruturas de dados torna-se necessária não só para armazenar informações de configuração pertinentes às redes e *hosts*, mas também informações gráficas referentes à representação do objeto na tela. Essas estruturas devem permitir que a posição do objeto seja facilmente obtida, para suportar funções ativadas através da seleção do objeto em tela através do *mouse*. Por exemplo, "clique" o *mouse* em cima do desenho de um *gateway*, aparece em tela uma janela contendo a imagem digitalizada do mesmo.

Nesse contexto, este trabalho apresentou um modelo para representação dinâmica de objetos em sistemas de gerenciamento, e mostrou como pode ser implementado utilizando recursos do Unix e o protocolo SNMP para se obter as informações necessárias para se construir a topologia e determinar componentes das redes. O protocolo SNMP foi utilizado considerando-se que esse trabalho se encaixa em sistemas de gerenciamento de redes. A implementação ocorreu em ambiente SunOS 4.3 e utilizou o pacote de domínio público 4.3/BSD SNMP para a comunicação SNMP. A implementação da interface gráfica com o

usuário utilizou o gerador de interfaces DevGuide, e o *toolkit* XView, e os desenhos em tela foram feitos através de primitivas gráficas do XWindow.

Os códigos fontes do sistema implementado a partir do protótipo deste modelo, e manuais, podem ser obtidos através de *anonymous ftp* no endereço xavante.icmsc.sc.usp.br:/pub/multview.

6. Referências

- [Br93] BRISA. *Gerenciamento de Redes -Uma Abordagem de Sistemas Abertos*. São Paulo, Makron Books, 1993.
- [Ca90] CASE, J. et. al. A Simple Network Management Protocol (SNMP). Network Working Group, *Request for Comments 1098*, (DDN Network Information Center, SRI International), 1990.
- [He90] HELLER, D. *XView Programming Manual - An OPEN LOOK Toolkit for X11*. O'Reilly&Associate, 1990. vol.7. v.2.
- [Jo89] JONES, O. *Introduction to the X Window System*. Englewood Cliffs, Prentice-Hall, 1989.
- [Kl88] KLERER, S. M. The OSI Management Architecture: an Overview. *IEEE Networks*, v.2, n. 2, p. 20-29, 1988.
- [Od94] ODA, C. S. Desenvolvimento de um Sistema Monitor Gráfico Baseado em Protocolo de Gerenciamento SNMP. Tese de Mestrado defendida no Instituto de Ciências Matemáticas de São Carlos em maio de 1994.
- [On89] ONIONS, J. ISODE: In Support of Migration. *Computer Networks and ISDN Systems*, v.17, p. 362-366, 1989.
- [Ro91] ROSE, M. T. Management Information Base for Network Management of TCP/IP-based internets: MIB-II. *Request for Comments 1158*, (DDN Network Information Center, SRI International), 1991.
- [Ro91] ROSE, M. T. *The Simple Book-An Introduction to Management of TCP/IP-based Internet*. Englewood Cliffs, Prentice-Hall, 1991.
- [Su90a] SUN MANUAL. *OpenWindows Developer's Guide 1.1 -Installation Manual*. Sun Microsystems, 1990.
- [Su90b] SUN MANUAL. *SunOS Reference Manual-Maintenance commands*. Sun Microsystems, 1990. v. 3.