**Desenvolvimento de um Sistema Monitor Gráfico
Baseado em Protocolo de Gerenciamento SNMP**

Cybelle Suemi Oda

Orientador: Prof. Dr. Edson dos Santos Moreira

**Dissertação apresentada ao Instituto
de Ciências Matemáticas de São Carlos,
Universidade de São Paulo, como parte
dos requisitos para obtenção do título de
Mestre na área de Ciências de Computação
e Matemática Computacional.**

**São Carlos
Maio de 1994**

*Para meus pais,
Oswaldo e Helena.*

*Para meu irmão,
George.*

Agradecimentos

Ao Prof. Dr. Edson dos Santos Moreira pela orientação, apoio e incentivo.

Aos professores Liane M. Rockenbach Tarouco, Markus Endler, Marcos J. Santana e Agma J. Traina pelas sugestões que foram incorporadas na versão final da tese.

Aos amigos do ICMSC, especialmente ao Rogério, Luis Paulo, Nivaldi, Regina Santana, Júlio, Issamy, Morte, Elder, Mauricio, Tchelo, Geraldo, Marquinhos, Hasegawa, Claudinha, Boca, Welton, Marcelo Stein, e João Batista. Aos meus amigos Alexandre (Pastel), Ana Luisa e Roberto (Fritz) por todos os momentos compartilhados. E às minhas amigas de coração Regina e Renata (Titi).

Aos funcionários do ICMSC, em especial à Bete, Laura, Paulinho, Sandra, Gi, Silvana, Marquinhos (Xerox), Eduardo, Cabral, Sonia, Angelo, Luciano, Zé Angelo, Sandra Doi, Dna. Dolores, Dna. Maria e aos guardas da portaria. Pela dedicação, amizade e serviços prestados.

À Fapesp, Fundação de Amparo à Pesquisa do Estado de São Paulo, e à Capes pelo apoio financeiro.

Ao CNPq pelos equipamentos utilizados.

Aos meus pais e ao meu irmão pelo amor e incentivo em todos os dias da minha vida...

À Deus pelos bons momentos e pela força para superar os obstáculos.

Resumo

A importância de se ter gerenciamento em ambientes de rede computadores vem se tornando cada vez mais evidente, à medida em que esses ambientes se estendem aumentando as interconexões e o número de *hosts*. Este trabalho vem justamente de encontro a essa necessidade, desenvolvendo um sistema monitor gráfico para redes de computadores.

O sistema apresenta uma interface gráfica baseada em janelas seguindo o padrão XWindow. Nas janelas são apresentados graficamente o desenho do mapa das redes, dentro do domínio de gerenciamento, e o desenho dos componentes de uma rede selecionada, proporcionando ao gerente uma visão espacial das redes. Os dados para a descoberta das redes são obtidos dinamicamente pelo sistema, sem a interação com o usuário, através do protocolo de gerenciamento SNMP e de alguns recursos de rede do sistema Unix. Adicionalmente, outras informações são obtidas, tais como: dados estatísticos sobre interfaces, tabela de roteamento, conexões ativas e informações sobre protocolos Internet por host.

O sistema foi projetado para ser o núcleo de um sistema de gerenciamento de redes, o MultView, com recursos de multimídia. Dessa forma, a estrutura de dados armazenada para cada objeto possui um campo que guarda o posicionamento gráfico do objeto nos desenhos. Esse campo pode ser utilizado por funções de multimídia, como por exemplo a apresentação de imagens digitalizadas dos objetos selecionados ou uma foto do administrador de uma dada rede.

Na implementação do sistema utilizou-se o pacote de domínio público 4BSD/ISODE SNMP, e o gerador de interfaces DevGuide e o *toolkit* XView na implementação da interface gráfica.

Abstract

The management task in network environments is becoming more and more vital, as these environments grow, getting interconnected and increasing the number of hosts. This work presents a graphical monitor system for computers network management, which meets the requirements of such task.

The monitor has a graphical interface which follows the XWindow standard. The windows present graphical information, such as the map of networks which are inside the management domain and the components of a given network, providing a spacial vision of the managed networks. The information used to discover the networks are dinamically obtained by the SNMP protocol and some Unix networking resources. Additionally, information such as routing table, active connections and Internet protocols information per host are provided.

The system was designed to be the kernel of a network management system, the MultView, which incorporates multimedia resources. For this reason, the data structures stored for each object have one field which comprises its graphical position, which can then be used by multimedia functions, for example, the presentation of digitalized images of selected objects, or a picture of the administrator of a given network.

The system was developed with the public domain software 4BSD/ISODE SNMP, and the interface generator DevGuide and the toolkit XView to implement the graphical interface.

Índice

Lista de Figuras e Tabelas	iv
1 Introdução	1
2 Redes de Computadores	3
2.1 INTRODUÇÃO	3
2.2 CONCEITOS BÁSICOS	4
2.2.1 LANs, WANs e MANs	4
2.2.2 Topologias	4
2.2.3 Meios de Transmissão	7
2.2.4 Protocolos de Comunicação	9
2.3 O MODELO DE REFERÊNCIA ISO/OSI	9
2.3.1 Estrutura	9
2.3.2 Serviços e Interfaces	11
2.4 PROTOCOLOS DE BAIXO NÍVEL	14
2.4.1 O Padrão IEEE 802.3 e Ethernet	14
2.4.2 O Padrão IEEE 802.5: Token Ring	16
2.4.3 FDDI	17
2.5 O CONJUNTO DE PROTOCOLOS INTERNET	18
2.5.1 Camada de Interface de Rede	19
2.5.2 Camada Internet	21
2.5.3 Camada de Transporte	24
2.5.4 Camada de Aplicação	26
2.6 DISPOSITIVOS DE INTERCONEXÃO DE REDES	26
2.6.1 Pontes (<i>Bridges</i>)	27
2.6.2 Roteadores (<i>Gateways</i>)	28
3 Gerenciamento de Redes de Computadores	30
3.1 INTRODUÇÃO	30

3.2 A ESTRUTURA DE GERENCIAMENTO INTERNET	31
3.2.1 Conceitos	32
3.2.2 Objetos Gerenciados	34
3.2.3 Estrutura Administrativa	37
3.2.4 O Protocolo de Gerenciamento SNMP	38
3.3 A VERSÃO 2 DA ESTRUTURA DE GERENCIAMENTO INTERNET - SNMPv2	40
3.4 A ARQUITETURA OSI DE GERENCIAMENTO	41
3.4.1 Os Modelos de Gerenciamento	42
3.4.2 Componentes da Arquitetura	43
3.5 PRODUTOS COMERCIAIS PARA GERENCIAMENTO DE REDES DE COMPUTADORES	46
3.5.1 SunNet Manager	47
3.5.2 AIX NetView/6000	48
4 Projeto do Sistema Monitor Gráfico de Redes	50
4.1 INTRODUÇÃO: UMA DESCRIÇÃO GERAL DO SISTEMA	50
4.2 FONTES DE INFORMAÇÕES SOBRE OBJETOS DE REDE	51
4.3 FUNÇÕES DO SISTEMA MONITOR	53
4.3.1 Apresentação Gráfica das Redes Gerenciadas	53
4.3.2 Apresentação Gráfica dos Componentes de uma Rede	54
4.3.3 Apresentação de Informações	55
4.4 INTERFACE DO SISTEMA COM O USUÁRIO	55
4.5 ESTRUTURAS DE DADOS	56
4.6 DIAGRAMA DE MÓDULOS DO SISTEMA	58
4.6.1 Módulo de Montagem das Estruturas de Dados	59
4.6.2 Módulo Gráfico	61
4.6.3 Módulo de Funções	64
5 Implementação do Sistema	66
5.1 INTRODUÇÃO	66
5.2 AMBIENTE DE DESENVOLVIMENTO	66
5.2.1 O Sistema XWindow	67
5.2.2 O Pacote ISODE	69

5.3 IMPLEMENTAÇÃO DAS ESTRUTURAS DE DADOS	71
5.4 IMPLEMENTAÇÃO DOS MÓDULOS	73
5.4.1 Módulo Gráfico	73
5.4.2 Módulo de Montagem das Estruturas de Dados	84
5.4.3 Módulo de Funções	89
6 Conclusões e Trabalhos Futuros	90
Apêndice A - Programas em gawk SNMP-capable	94
Glossário	103
Referências Bibliográficas	107

Lista de Figuras e Tabelas

2.1	Topologia em malha irregular	5
2.2	Topologia em estrela	5
2.3	Topologia em barramento	6
2.4	Topologia em anel	6
2.5	Quatro estações conectadas via um nó central	7
2.6	O Modelo de Referência ISO/OSI	9
2.7	Relação entre as primitivas de serviço	12
2.8	Formato de um PDU	13
2.9	Formato de um IDU	13
2.10	Relação entre camadas na interface	13
2.11	Configuração usual de uma Ethernet	15
2.12	Formato do pacote Ethernet	16
2.13	Token Ring	17
2.14	FDDI	18
2.15	Estrutura em camadas do conjunto de protocolos Internet	19
2.16	Tabela ARP simplificada	20
2.17	Formato do pacote ARP	20
2.18	As três classes primárias de endereços Internet	22
2.19	Exemplo de uma máscara de sub-rede	23
2.20	Tabela simples de roteamento	23
2.21	Formato do datagrama IP	24
2.22	Formato do datagrama UDP	25
2.23	Formato do segmento TCP	25
2.24	Estrutura dos protocolos Internet e algumas aplicações	26
2.25	Operação de uma ponte a nível de camadas OSI	27
2.26	Uma configuração com quatro LANs e duas pontes	27
2.27	Operação de um roteador a nível de camadas OSI	28
3.1	Sistema de gerenciamento SNMP	32
3.2	Gerenciamento através do agente <i>proxy</i>	33
3.3	Árvore de Identificação ASN.1	35
3.4	Buscando informações de gerenciamento no agente	38
3.5	Buscando informações de gerenciamento transversalmente no agente	38
3.6	Atualizando informações de gerenciamento no agente	39
3.7	Reportando a ocorrência de um evento ao gerente	39
3.8	Agente proxy entre SNMPv1 e SNMPv2	41
3.9	Modelo de arquitetura do gerenciamento OSI	43
3.10	Sistema de gerenciamento CMIP	44
3.11	Interação entre o gerente SunNet e os agentes	47
3.12	Interação entre o gerente AIX NetView e os agentes	48
4.1	Apresentação do mapa das redes gerenciadas	54
4.2	Apresentação dos componentes de uma rede	54
4.3	Lista de redes juntamente com a lista de hosts	57

4.4	Lista de gateways	58
4.5	Diagrama de módulos do Sistema Monitor	58
4.6	Diagrama do módulo de montagem das estruturas de dados	59
4.7	Diagrama do módulo gráfico	61
4.8	Diagrama do módulo de desenho do mapa das redes gerenciadas	62
4.9	Diagrama do módulo de funções	64
5.1	Janela de controle do Devguide	68
5.2	Estrutura dos módulos do ISODE	69
5.3	Implementação em C das E.Ds	72
5.4	Janela de apresentação do mapa das redes gerenciadas	74
5.5	O menu principal	74
5.6	Janela de símbolos gráficos	75
5.7	Janela de apresentação dos componentes de uma rede	75
5.8	Janela de apresentação das informações sobre o host ou gateway	76
5.9	Menu de acesso às funções	76
5.10	Janela de apresentação de informações sobre a interface de rede	77
5.11	Janela de apresentação das conexões ativas	78
5.12	Janela de apresentação de informações sobre protocolos TCP/IP	79
5.13	Janela de apresentação da tabela de roteamento	80
5.14	Janela de apresentação de informações sobre a alocação de mbufs	81
5.15	Janela de apresentação de informações sobre a tabela de impressão	82
5.16	Janela de apresentação de informações sobre os system users	83
5.17	Janela de apresentação dos objetos do grupo system	83
5.18	Programa TSNMP	84
5.19	O módulo de montagem inicial juntamente com o programa TSNMP	85
5.20	Programa TGATE	85
5.21	O módulo de montagem da lista de gateways e o programa TGATE	86
5.22	Programa THOSTS	86
5.23	Programa TLOCAL	87
5.24	O módulo de busca de novos hosts e os programas em gawk SNMP-capable	87
5.25	Programa TOGATE	88
5.26	O módulo busca de novos gateways e os programas em gawk SNMP-capable	89

CAPÍTULO 1

Introdução

A expansão das redes de longa distância e o aumento da utilização de redes a nível local através da descentralização do processamento e da interconexão de computadores e equipamentos em ambientes acadêmicos e comerciais, vem aumentando a necessidade e a complexidade das atividades relacionadas ao gerenciamento de redes.

O gerenciamento de redes envolve basicamente duas tarefas: diagnóstico e solução de problemas. São exemplos: problemas de comunicação causados por queda de linhas ou roteadores, sobrecarga de redes e queda de servidores (arquivos, impressoras, nomes, etc). Para auxiliar o gerente da rede a desempenhar tais tarefas, desenvolveram-se ferramentas de gerenciamento capazes de monitorar a rede e detectar problemas. A princípio, essas ferramentas eram vinculadas a redes proprietárias, mas posteriormente com o surgimento de protocolos padrões integrando diferentes tecnologias, a atividade de gerenciamento acompanhou essa tendência sendo desenvolvidas estruturas de gerenciamento pela Internet e pela ISO (*International Organization for Standardization*), mais conhecidas através dos seus protocolos de gerenciamento SNMP (*Simple Network Management Protocol*) [Ca90] [Ro90] e CMIP (*Common Management Information Protocol*) [Br93] [Kl88] [Wa89] respectivamente. Dessa forma, ferramentas de controle e monitoração que antes eram empregadas separadamente, começaram a ser unificadas em um único sistema de gerenciamento seguindo uma das estruturas padrões.

O trabalho de mestrado, tema desta dissertação, envolve o desenvolvimento de um sistema monitor gráfico de redes baseado na estrutura Internet, apresentando:

- interface gráfica seguindo o padrão XWindow,
- implementação de um gerente SNMP,
- descoberta dinâmica de redes dentro do domínio de gerenciamento,
- desenho do mapa das redes e de seus componentes,
- levantamento de dados estatísticos,
- status de *hosts* e roteadores,
- apresentação de tabelas de roteamento e
- apresentação de conexões ativas.

Dentro de um contexto maior, o sistema monitor é a implementação do núcleo de um sistema de gerenciamento com recursos de multimídia, o MultiView. O projeto deste

sistema inclui a coexistência dos protocolos SNMP e CMIP, funções de gerenciamento básicas, como por exemplo, coleta de dados sobre tráfego e performance da rede em tempo real e monitoração de *hosts* e outros dispositivos de rede; recursos de multimídia através dos quais são apresentadas imagens (foto do gerente da rede, imagem digitalizadas de equipamentos, imagens em tempo real captadas através de uma câmera de vídeo, etc) e transmitidos sons (mensagens gravadas, *beeps* de alerta, e mensagens *on-line*). Nesse sentido, o sistema monitor foi projetado para permitir a extensão e dar suporte ao desenvolvimento das funções do MultView.

O suporte teórico para o desenvolvimento do sistema monitor gráfico de redes foi adquirido através de uma revisão bibliográfica descrita nos capítulos 2 e 3. O projeto do sistema monitor gráfico de redes encontra-se no capítulo 4 e aspectos de implementação no capítulo 5. Esses capítulos são detalhados a seguir, juntamente com o capítulo de conclusões e trabalhos futuros.

- **Capítulo 2:** discorre sobre conceitos relacionados a redes de computadores como topologias, meios de transmissão, redes locais, redes metropolitanas, redes de longa distância, protocolos de comunicação, tecnologias de rede, o modelo de referência ISO/OSI, o conjunto de protocolos TCP/IP, e elementos de interconexão física como pontes e roteadores.
- **Capítulo 3:** aborda o gerenciamento de redes de computadores apresentando um breve histórico e motivação; introduz as estruturas de gerenciamento OSI e Internet juntamente com os seus respectivos protocolos de gerenciamento CMIP e SNMP versão 1 e 2. Apresenta dois softwares comerciais de gerenciamento, o SunNet Manager e o AIX NetView/6000, fazendo uma breve comparação com o sistema de gerenciamento Multview.
- **Capítulo 4:** descreve o projeto do sistema monitor definindo suas funções, a interface gráfica com o usuário, as estruturas de dados referentes ao armazenamento das informações sobre as redes e roteadores e a estrutura modular do sistema. Faz também um levantamento de objetos da MIB-II [Ro90b] e de arquivos UNIX que podem ser utilizados na obtenção de informações sobre as redes.
- **Capítulo 5:** aborda o ambiente de desenvolvimento da implementação do sistema descrevendo os equipamentos e softwares utilizados, e como as funções do sistema foram implementadas dentro desses ambientes. Faz uma apresentação das telas e funcionalidades do sistema monitor.
- **Capítulo 6:** apresenta as conclusões e sugestões para trabalhos futuros.

CAPÍTULO 2

Redes de Computadores

2.1 Introdução

Nas últimas décadas a indústria eletrônica evoluiu rapidamente fornecendo suporte para o desenvolvimento de computadores cada vez mais potentes e velozes. Através dos avanços na microeletrônica e da crescente diminuição do custo do hardware, tornou-se viável a produção de computadores de menor porte, com custo mais acessível para os usuários em geral. Conseqüentemente, essas máquinas passaram a fazer parte do cotidiano de indústrias, empresas, centros de pesquisas, e universidades, automatizando atividades diversas. Porém, o processamento pesado continuou por algum tempo sendo realizado em *mainframes* localizados em Centros de Processamento de Dados.

Nessa mesma época, já se tinha pleno domínio da tecnologia de comunicação através das redes telefônicas, de rádio e televisão. O cenário encontrava-se então propício para o desenvolvimento de um novo conceito que viria revolucionar a informática: as **Redes de Computadores**.

Conceitualmente as redes de computadores são definidas como uma coleção de computadores autônomos interconectados capazes de se comunicarem [Ta89], onde os subsídios para a interconexão física e a troca de dados foram fornecidos pela tecnologia de comunicação.

As redes apresentam uma série de vantagens que favorecem sua utilização, dentre elas:

- troca de informações entre computadores;
- compartilhamento de recursos (CPUs, dados, impressoras, etc);
- alta confiabilidade a nível de fontes alternativas de recursos; e
- relação custo/desempenho melhor que os *mainframes*.

Adicionalmente, aplicativos desenvolvidos sobre as redes tais como correios eletrônicos, transferência de arquivos remotos e *login* remoto contribuem sensivelmente para sua popularização.

Neste capítulo são abordados conceitos básicos relacionados às redes de computadores, tecnologias de rede com ênfase em Ethernet [Co91a] [Me76], o modelo de referência ISO/OSI [Ta89], o conjunto de protocolos Internet [Co91a] [Ro91] [So91], e dispositivos de interconexão física de rede [Ta89] [Be88] [Se88] os quais juntamente com os protocolos de comunicação, fornecem os meios para a interconexão de redes heterogêneas.

2.2 Conceitos Básicos

Neste item são apresentados conceitos básicos relativos a redes de computadores, os quais são utilizados no decorrer deste capítulo. Esses conceitos referem-se à classificação e topologia das redes, meios físicos de transmissão e protocolos de comunicação.

2.2.1 LANs, WANs, e MANs

As Redes de Computadores podem ser classificadas em três categorias, levando em consideração a área geográfica que abrangem [Ta89]:

- **LAN (*Local Area Network*) ou Rede Local:** abrange uma área limitada como um prédio ou o campus de uma universidade. Três características básicas definem uma rede local:
 1. diâmetro de não mais que alguns quilômetros;
 2. taxa de transferência de dados de pelo menos alguns Mbps; e
 3. administração sob uma única organização.
- **WAN (*Wide Area Network*) ou Rede de Longa Distância:** engloba áreas geográficas extensas, por exemplo países inteiros; possui taxa de transferência de dados inferior a 1Mbps, e estão sob o domínio de várias organizações.
- **MAN (*Metropolitan Area Network*) ou Rede Metropolitana:** corresponde ao meio termo entre as LANs e WANs, abrangendo, por exemplo, uma cidade inteira. Utiliza tecnologia de rede local.

2.2.2 Topologias

A topologia de uma rede reflete a estrutura de interconexão física dos nós que a compõe. A princípio existem várias maneiras de se configurar uma rede, principalmente levando-se em conta a relação custo/desempenho, a posição geográfica dos nós da rede e as necessidades do usuário. Cada topologia possui características próprias com diferentes implicações quanto ao desenvolvimento, operação e manutenção.

A seguir são apresentadas as principais topologias com suas características, baseadas em [Gi86].

Malha Irregular

Nessa topologia, cada estação ou nó da rede de comunicação pode estar conectada diretamente a outras estações, dando origem a uma estrutura de interconexão irregular como exemplificada na figura 2.1.

Figura 2.1 -Topologia em malha irregular

Quando uma mensagem é enviada de uma estação para outra, passa por todos os nós intermediários, sendo que em cada um, a mensagem é armazenada até que uma linha de saída esteja livre, quando é então passada adiante até que chegue à estação destino. Esse tipo de configuração é normalmente utilizada em redes de longa distância, como por exemplo a Internet e a Bitnet. Apresenta como vantagem o fato do arranjo de interconexões poder ser realizado de acordo com o tráfego de informações permitindo, portanto, a otimização nos meios de comunicação. Por outro lado, a diversidade de caminhos torna necessária a existência de softwares de roteamento nos nós da rede, o que implica numa demanda de considerável capacidade de processamento e armazenamento.

A configuração de redes locais utilizando essa topologia não é conveniente, uma vez que a complexidade de cada nó, devido à tarefa de roteamento, exige recursos exclusivos, o que tende a inviabilizar economicamente a maior parte das aplicações de redes locais de computadores.

Estrela

Nessa estrutura, um nó central é utilizado para conectar todos os outros nós da rede, figura 2.2. Esse é o único que possui decisão de roteamento, simplificando bastante a implementação dos demais nós.

Figura 2.2 -Topologia em estrela

Quando uma estação deseja transmitir dados, envia um pedido de conexão ao nó central, o qual então estabelece um caminho dedicado entre as duas estações. A desvantagem dessa configuração consiste principalmente na dependência das estações do nó central, onde uma falha nesse inviabilizaria a comunicação em toda a rede. Além disso, o aumento de estações conectadas eleva a complexidade do nó central, bem como o número de conexões físicas, elevando o custo associado ao meio físico.

Barramento

Essa topologia é caracterizada pela utilização de um meio físico multiponto, sendo bastante utilizada em redes locais.

Figura 2.3 -Topologia em Barramento.

Como na topologia em anel, não existem decisões de roteamento e apenas uma única estação pode transmitir dados por vez, sendo esses transmitidos na forma de pacotes. Quando as mensagens são transmitidas, a única decisão necessária em cada nó de comunicação é a identificação das mensagens que lhes são destinadas, não existindo armazenamento intermediário.

Devido ao compartilhamento do meio físico, é necessário haver um mecanismo de controle de acesso e utilização do mesmo pelas estações, a fim de se evitar colisões dos sinais transmitidos simultaneamente. Os nós de comunicação intermediários têm geralmente um comportamento passivo no processo de transmissão, o que constitui uma vantagem quanto à confiabilidade, porém limita o alcance da rede.

Anel

Na topologia em anel todos os nós estão conectados, através de elementos repetidores, a um meio físico de transmissão cujo formato é um anel.

Figura 2.4 -Topologia em anel.

Não há necessidade de decisão de roteamento, os dados circulam

unidirecionalmente no anel de nó em nó. Assim como na topologia em barramento, existe um mecanismo de controle de acesso ao meio físico realizado através da utilização de um pequeno pacote denominado *token*. Esse pequeno pacote circula no anel e uma estação que deseje transmitir deve capturá-lo para poder realizar a transmissão. Como existe apenas um *token* no anel, somente uma estação pode transmitir por vez. Os dados são enviados na forma de um pacote contendo os endereços fonte e destino. O pacote circula pelo anel até chegar ao nó destino, o qual reconhece o endereço como sendo o seu e copia o pacote para um *buffer* local; o pacote continua a circular até chegar ao nó de origem fornecendo, inclusive, um meio de reconhecimento (*acknowledgement*).

O fato dos nós intermediários funcionarem como repetidores afeta a confiabilidade da rede. Uma técnica utilizada para contornar esse problema é o nó central [Ta89], ilustrado na figura 2.5.

Figura 2.5 -Quatro estações conectadas via um nó central.

Cada estação é conectada ao nó central através de um cabo com pelo menos dois pares trançados. As chaves são energizadas pela corrente que vem das estações; caso uma delas seja desligada, ou o anel se quebre, a chave se fecha isolando a parte do anel que está com problemas, evitando a queda da rede.

2.2.3 Meios de Transmissão

Os dados são transmitidos de uma estação para outra na forma de uma sequência de bits através de um meio físico ou canal de comunicação. Neste item são apresentados os meios mais utilizados, baseados em [Ta89] e [Gi86], juntamente com suas características principais.

Par Trançado

O par trançado consiste de dois fios metálicos (geralmente cobre) enrolados em espiral, para evitar interferências elétricas, principalmente, de pares similares próximos a ele. Este meio físico é de alta disponibilidade por ser utilizado nos sistemas telefônicos, é de fácil manuseio e instalação. Possui razoável imunidade a ruídos.

Cabo Coaxial

Outro meio de transmissão bastante comum é o cabo coaxial que consiste basicamente de um condutor cilíndrico central, geralmente de cobre, circundado por uma malha de fios de cobre ou de uma manga de alumínio estruturada que serve como condutor de retorno e blindagem eletrostática. Essa estrutura fornece uma boa combinação de alta largura de banda (*bandwidth*) e segurança (imunidade a ruído).

Os cabos coaxiais são bastante utilizados em redes locais, por exemplo em barramentos. Existem duas maneiras de se conectar um computador a um cabo coaxial: na primeira corta-se o cabo e utiliza-se uma junção T; na segunda, onde não há a necessidade de se cortar os cabos, utiliza-se uma *vampire tap*, um dispositivo que fura o cabo coaxial até atingir o condutor central. Existem dois tipos de cabos coaxiais: o de banda básica, 50 ohms, utilizado para transmissão de sinais digitais, e o de banda larga, 75 ohms, para transmissão analógica.

Fibras Óticas

Com o desenvolvimento da tecnologia ótica foi possível transmitir dados através de pulsos de luz - um pulso de luz pode ser utilizado para assinalar bit 1, e a ausência do sinal bit 0.

Um sistema transmissor ótico possui três componentes:

- Dispositivo transmissor de luz: em geral diodos emissores de luz (LED) ou diodos lasers, que emitem pulsos de luz quando uma corrente elétrica é aplicada.
- Dispositivo fotodetector: geralmente fotodiodos PIN ou fotodiodos de avalanche (APD), o qual gera um pulso elétrico quando recebe um pulso de luz.
- Fibra ótica: suporte de transmissão para luz infravermelha. É uma fibra ultrafina de vidro ou sílica fundida.

As fibras óticas apresentam imunidade eletromagnética, isolamento elétrica, segurança de informação e alta bandwidth sendo muito utilizada como *backbone* em redes locais.

2.2.4 Protocolos de Comunicação

Com os computadores conectados através de um meio físico, segundo uma determinada topologia, é necessário estabelecer critérios para a comunicação entre essas máquinas. Esses critérios são determinados pelo **protocolo de comunicação**, um conjunto de regras formais que determina como software e hardware devem interagir a fim de transmitirem informações. Em outras palavras, o protocolo especifica o formato das mensagens, como as mensagens devem ser transmitidas e tratadas, descreve como os erros devem ser manipulados, etc [Co91a].

Os protocolos podem permitir comunicação a diversos níveis. A baixo nível descrevem detalhes de interface máquina à máquina, por exemplo a ordem em que os bits devem ser enviados através do meio físico. Os de alto nível descrevem, por exemplo, a comunicação entre programas de aplicação.

2.3 O Modelo de Referência ISO/OSI

A necessidade de um sistema aberto, não proprietário, fez com que a ISO (*International Organization for Standardization*) juntamente com o CCITT (*International Telephone and Telegraph Consultative Committee*) desenvolvessem em 1979 um modelo de referência para interconexão de sistemas abertos [Ro91]. Este modelo descreve os diversos níveis de protocolos (e suas funcionalidades) a serem usados para a comunicação entre computadores, sem especificar como estes devem ser implementados,

Através da implementação desse modelo é possível realizar a comunicação entre redes heterogêneas, formando assim uma única rede virtual homogênea.

2.3.1 Estrutura

O Modelo de Referência possui uma estrutura hierárquica consistindo de sete camadas [Ta89], figura 2.6.

Figura 2.6 -O Modelo de Referência ISO/OSI.

A **camada física** é a mais baixa do modelo. Está diretamente associada aos meios físicos que interconectam os sistemas e aos mecanismos de transmissão de bits através desses meios. A camada acima, **enlace de dados**, subdivide-se em duas subcamadas: **MAC** (*Medium Access Control*) a qual define o método de acesso ao meio, e **LLC** (*Logical Link Control*) que executa as funções de enlace de dados independentemente do meio e do controle de acesso ao meio. A camada de enlace de dados é responsável por agrupar os dados em pacotes, transmiti-los sequencialmente e processar os pacotes de reconhecimento que chegam. Nos casos de pacotes perdidos, danificados ou duplicados, esta camada lida com esses problemas retransmitindo pacotes, ou descartando pacotes inadequados.

A **camada de rede** está relacionada com o controle das operações de uma sub-rede de comunicação (meio de transmissão juntamente com os elementos chaveadores). Dentre as tarefas desempenhadas estão roteamento de pacotes e controle de congestionamento. Quando um pacote deve ser transmitido através de várias redes, pode haver problemas de incompatibilidade de formas de endereçamento ou de tamanho de pacotes. É tarefa da camada de rede lidar com esses problemas permitindo, dessa forma, a interconexão de redes heterogêneas.

A função básica da **camada de transporte** é aceitar dados da camada de sessão, dividi-los em pedaços menores, se necessário, e passá-los para a camada de rede. A cada conexão requisitada pela camada de sessão, é estabelecida uma conexão de rede. Podem ser criadas múltiplas conexões, ou várias conexões de transporte podem ser multiplexadas na mesma conexão da rede. A camada de transporte oferece tipos de serviços que são determinados quando a conexão é estabelecida. O mais popular é o canal ponto-a-ponto e sem erros, o qual transporta mensagens na ordem em que foram enviadas. Outros serviços são: transporte de mensagens isoladas sem garantia sobre ordem de entrega, e envio de mensagens para múltiplos destinos (*broadcasting*). É uma camada fim-a-fim e fonte-destino, uma vez que os processos na máquina fonte se comunicam com os processos similares na máquina destino.

A **camada de sessão** permite que usuários em máquinas diferentes estabeleçam sessões entre eles. Uma sessão deve permitir a um usuário realizar um *login* em um sistema de tempo compartilhado remoto ou transferir um arquivo entre duas máquinas. As sessões permitem que haja fluxo de informações em ambas as direções simultaneamente ou em apenas uma. Nesse último caso, a camada de sessão auxilia no controle de turnos. Fornece ainda os serviços de gerenciamento de *token* e sincronização. No primeiro, a camada de sessão fornece *tokens* que controlam os turnos quando a apenas um dos lados é permitido realizar a tarefa crítica. No segundo, a camada de sessão fornece um modo de inserir pontos de verificação no fluxo de dados, assim quando ocorre uma quebra durante a transmissão de fluxo, apenas o dado após o último ponto de verificação deve ser repetido.

Ao contrário das demais camadas que somente se preocupam em transferir dados de modo confiável, a **camada de apresentação** lida com aspectos semânticos e sintáticos da informação transmitida. Os dados a serem transmitidos, obtidos do programa do usuário, são normalmente representados por cadeias de caracteres, números reais e inteiros, estruturas de dados, etc. Visto que máquinas diferentes possuem códigos diferentes para representá-los, é tarefa da camada de apresentação permitir que máquinas heterogêneas se

comuniquem. Isto é possível através da definição de dados em um modo abstrato, os quais podem ser convertidos para a representação padrão da rede. Essa camada trata também outros aspectos de representação de informação, tais como criptografia e compressão de dados.

Na **camada de aplicação** estão facilidades de propósito geral e especial, tais como correio eletrônico, entrada para tarefas (*jobs*) remotas, e transferência de arquivos remotos. A camada de aplicação trata de tornar as incompatibilidades causadas por dispositivos ou sistemas transparentes ao processo do usuário.

2.3.2 Serviços e Interfaces

Como visto no item anterior, o modelo de referência ISO/OSI divide a atividade de comunicação entre computadores em sete camadas funcionais, onde cada uma oferece serviços para a camada superior. Esses serviços são implementados por elementos denominados **entidades**, os quais podem ser entidades de software (ex. um processo), ou de hardware (chip de I/O inteligente) [Ta89], e são referidas de acordo com a camada em que se localizam. Por exemplo, entidades de aplicação, entidades de apresentação e entidades de transporte. As entidades de mesma camada mas em máquinas diferentes são denominadas **entidades pares**.

Os serviços da camada N-1 são acessados pela camada N através de pontos de acesso denominados **SAPs** (*Service Access Points*), identificáveis através de um endereço. O mecanismo de comunicação interprocessos *socket* [Le89], é um exemplo de SAP.

Primitivas de Serviços

Um serviço é especificado formalmente através de um conjunto de primitivas (operações) que especificam qual a ação ou notificação que o serviço deve realizar sobre ações tomadas por uma entidade par. As primitivas de serviço podem ser divididas em quatro classes [Ta89]:

- primitiva *request*: utilizada pela entidade para informar que um serviço deve ser realizado;
- primitiva *indication*: indica que uma entidade deve ser informada sobre um evento;
- primitiva *response*: indica que uma entidade quer responder a um evento;
- primitiva *confirm*: uma entidade deve ser informada sobre seu pedido.

O diagrama a seguir apresenta a relação entre as primitivas.

Figura 2.7 -Relação entre as primitivas de serviço.

Dentro desse contexto, o protocolo é utilizado pela entidade para implementar a definição de seus serviços.

Serviços Orientados a Conexão e Serviços sem Conexão

As camadas podem oferecer dois tipos de serviços: orientado a conexão ou sem conexão. O serviço orientado a conexão apresenta três fases distintas [Ro91]:

- estabelecimento de conexão: antes de haver a troca de dados é necessário estabelecer uma conexão entre as entidades pares;
- transferência de dados;
- liberação da conexão.

No serviço sem conexão há apenas a transferência de dados sem a necessidade de um estabelecimento prévio de conexão. Nesse caso, as mensagens possuem o endereço da máquina destino e são roteadas através do sistema independentemente umas das outras [Ta89].

A qualidade dos serviços pode ser caracterizada como confiável (garantia da entrega dos dados), ou não confiável. Os serviços confiáveis são geralmente implementados através de *acknowledgements*, ou seja, o receptor sempre envia um pacote de reconhecimento para o emissor referente à mensagem recebida, o qual fica então sabendo se os dados foram ou não recebidos. Porém, se por um lado há confiabilidade, por outro há perda de tempo e *bandwidth* devido aos pacotes de reconhecimento extras e processamento destes. Os serviços sem conexão e não confiáveis são denominados **serviços de datagramas**, enquanto os confiáveis são chamados **serviços de datagrama confiáveis** [Ta89].

Interfaces

Na comunicação entre duas entidades pares, as informações na máquina local são passadas da camada superior para a inferior até atingirem o meio físico, por onde são

transmitidas até alcançarem a máquina remota. Nesta, as informações seguem o caminho reverso atingindo a entidade para a qual foram destinadas [Ro91] [Ta89].

A ISO formaliza as interfaces entre as camadas através da definição de regras para a passagem de dados de uma camada para outra, através de um mecanismo denominado **encapsulamento** [Ro91]. Nesse mecanismo, antes de uma camada N passar os dados *Service Data Units* (SDU) para a camada N-1, anexa um pequeno cabeçalho denominado *Protocol Control Information* (PCI), formando o *Protocol Data Unit* (PDU), unidade de informação trocada por entidades pares.

Figura 2.8 -Formato de um PDU.

A seguir, anexa um outro cabeçalho denominado *Interface Control Information* (ICI), que identifica a primitiva de serviço invocada do serviço N-1. O objeto resultante é denominado *Interface Data Unit* (IDU), o qual é passado para a camada N-1 através do SAP da camada N-1.

Figura 2.9 -Formato de um IDU.

Ao receber o IDU, a camada N-1 separa o ICI do PDU, o qual para ela é um SDU (N-1), e chama a primitiva especificada.

Figura 2.10 -Relação entre camadas na interface.

Na máquina remota ocorre o mesmo mecanismo, mas em sentido inverso. Essas nomenclaturas podem ser substituídas por termos mais familiares:

SDU = dados
PCI = cabeçalho
PDU = pacote

Nesse contexto, o encapsulamento consiste em colocar o pacote (PDU) da camada superior na parte de dados (SDU) da camada inferior [Ro91].

2.4 Protocolos de Baixo Nível

Ilustrando os protocolos de baixo nível que englobam as camadas física e de enlace de dados do modelo ISO/OSI, tomam-se como exemplo os padrões IEEE 802.3 e Ethernet, IEEE 802.5 e FDDI por serem alguns dos mais utilizados atualmente.

2.4.1 O Padrão IEEE 802.3 e Ethernet

Ethernet [Me76] [Ta89] é o nome dado a uma tecnologia de barramento para troca de pacotes em redes locais, com controle de acesso distribuído. Foi desenvolvida pela Xerox PARC no início dos anos setenta e, devido ao seu sucesso, a Xerox, DEC e Intel projetaram um padrão para a Ethernet de 10 Mbps. Esse padrão formou a base para o padrão IEEE 802.3, o qual descreve toda uma família de CSMA/CD 1-persistente rodando à velocidade de 1 a 10 Mbps em diversos meios.

Método de acesso ao meio

O meio utilizado pela Ethernet é inerentemente *broadcast*, implicando na necessidade de um método de acesso. O mecanismo utilizado denomina-se CSMA/CD *Carrier Sense Multiple Access with Collision Detection* [Co91a] [Ta89].

Quando uma estação deseja transmitir dados, o primeiro passo é monitorar o canal até encontrá-lo livre, começando assim a transmissão dos dados. Tendo em vista que o sinal não se propaga instantaneamente pelo canal, é possível que duas ou mais estações comecem a transmitir simultaneamente ocorrendo uma **colisão**. A Ethernet lida com o problema de colisão da seguinte forma: cada estação fica monitorando o canal enquanto transmite seus dados para verificar se não há nenhum distúrbio no sinal; quando o distúrbio é detectado, a transmissão é imediatamente abortada e a estação espera um tempo aleatório antes de retransmitir os dados. Esse monitoramento é tecnicamente denominado detecção de colisão (*collision detection*-CD). O termo 1-persistente refere-se ao fato da estação sempre transmitir ao encontrar o meio livre.

Configuração de uma rede Ethernet

A configuração usual de uma rede Ethernet é descrita no esquema abaixo [Me76]:

Figura 2.11 -Configuração usual de uma Ethernet.

- **Cabo de interface:** conecta o *transceiver* à placa de interface. Possui comprimento de até 50 m, sendo formado por cinco pares trançados (dois para entrada e saída de dados, dois para entrada e saída de sinais de controle e o quinto para controlar o transceiver).
- **Segmento ETHER:** meio passivo de propagação de sinais digitais. Pode ser par trançado, cabo coaxial ou fibra ótica.
- **Transceiver:** dispositivo que conecta a interface da estação à rede local. Lida com a detecção da portadora e com a detecção de colisões.
- **Interface serial Ethernet:** responsável por serializar e deserializar dados paralelos nas estações. Essa interface possui hardware adequado para somente aceitar os pacotes destinados à estação.
- **Controlador Ethernet:** sistema de baixo nível responsável por receber e enviar pacotes de e para o transceiver. Realiza o agrupamento dos dados no formato do pacote adequado, e em caso de colisão, gera o intervalo aleatório de retransmissão baseado na contagem de colisões. Computa o *checksum* dos dados quando são transmitidos e recebidos, a fim de verificar se não houve erros durante a transmissão.
- **Repetidores:** dispositivo físico que amplifica e retransmite sinais em ambas as direções.

Utilizado para estender o alcance máximo de uma rede local.

Endereçamento

Os endereços Ethernet são inteiros de 48 bits associados ao hardware de interface Ethernet pelos fabricantes, os quais para assegurarem a unicidade de endereços, adquirem blocos de endereços físicos e os associam em sequência conforme são fabricadas as interfaces. Assim, se a interface Ethernet de uma máquina é trocada, o seu endereço físico é alterado.

Existem três tipos de endereços Ethernet [Co91a]:

- endereço físico de uma interface Ethernet;
- endereço *broadcast*: quando se deseja enviar o pacote para todas as estações na rede;
- endereço *multicast*, quando se deseja endereçar o pacote a apenas um determinado grupo de estações na rede.

Formato do pacote Ethernet

Os pacotes Ethernet são de tamanhos variados tendo no máximo 1526 bytes e no mínimo 64 bytes [Co91a] [Ta89].

Figura 2.12 -Formato do pacote Ethernet.

O preâmbulo consiste de 64 bits alternando 0s e 1s, sendo utilizado para auxiliar a estação receptora na sincronização. O campo CRC (*checksum*) auxilia na detecção de erros. O campo tipo de pacote indica qual o protocolo a ser utilizado para processar o pacote, permitindo que múltiplos protocolos sejam utilizados conjuntamente numa mesma máquina, e suportados por uma mesma rede física sem interferência. O campo PAD é utilizado para garantir que o pacote tenha o tamanho mínimo de 64 bytes. Caso o número de bytes do campo de dados seja inferior a 46 bytes, o campo PAD é preenchido até que o tamanho mínimo seja alcançado [Co91a] [Ro91] [Ta89].

2.4.2 O Padrão IEEE 802.5: Token Ring

A Token Ring é provavelmente a técnica mais antiga de controle de acesso ao meio em redes configuradas em anel [St84b], tendo sido selecionada para padronização

tornando-se o padrão IEEE 802.5.

Figura 2.13 -Token Ring.

O acesso ao meio é realizado através de um *token* que circula no anel. Quando todas as estações estão desocupadas, o *token* é marcado como livre. Uma estação que deseje transmitir um pacote espera até que o *token* passe por ela, quando então altera o padrão do bit de livre para ocupado enviando o pacote logo após. O pacote realiza uma volta no anel sendo copiado pela estação destino e retirado de circulação pela estação que o transmitiu, a qual insere um novo *token* livre dado que as seguintes condições sejam satisfeitas [St84b]:

- a estação completou a transmissão do seu pacote, e
- o *token* ocupado retornou à estação.

Visto que existe apenas um *token*, uma única estação pode transmitir num dado instante, solucionando o problema de acesso ao meio de transmissão.

Existem duas situações que podem fazer com que a rede falhe: uma é a perda do *token*, e a outra é a circulação persistente de um *token* ocupado. Para lidar com esses problemas, o padrão IEEE 802 especifica que uma estação deve permanecer como monitor ativo, sendo responsável por detectar um *token* perdido através de um *timeout*, retirá-lo e fornecer um novo *token* livre [St84b].

2.4.3 FDDI

A FDDI (*Fiber Distributed Data Interface*) [Ta89] é uma rede de fibra ótica tipo Token Ring com taxa de transmissão de 100 Mbps, permitindo conexões a distâncias de até 200 km, conectando no máximo 1000 estações. Devido a sua alta capacidade de transmissão, a FDDI é geralmente utilizada como *backbone* de redes locais.

Em relação ao meio físico, a FDDI utiliza LEDs e fibras óticas formando um anel duplo, um no sentido horário e outro no anti-horário, figura 2.14. Essa configuração aumenta a confiabilidade do anel permitindo sua reconfiguração caso um dos anéis se quebre, ou quebrem no mesmo ponto.

Figura 2.14 -FDDI.

Uma diferença marcante entre essa tecnologia e o padrão 802.5, é a existência de vários pacotes circulando no anel. Isso é necessário devido ao comprimento e ao número de estações, caso contrário o atraso devido à espera do *token* seria substancial.

2.5 O Conjunto de Protocolos Internet

A história da Internet tem origem nos anos setenta através da DARPA (*Defense Advanced Research Project Agency*), agência do governo americano pioneira nas pesquisas de redes de comutação de pacotes (*packet switching*) [Ro91].

No início havia apenas uma rede, a ARPANET, conectando algumas dezenas de computadores por todo os Estados Unidos. Com o desenvolvimento do hardware e o advento das tecnologias Ethernet, Packet Radio (utiliza as ondas de rádio como meio de transmissão) e de satélites, a DARPA viu-se frente a frente com a necessidade de desenvolver uma arquitetura de rede que permitisse a interconexão dessas diversas tecnologias garantindo confiabilidade. Assim, nos anos oitenta surgiu o conjunto de protocolos Internet, também conhecido por TCP/IP [Co91a] [Ro91] [So91]. Esse conjunto de protocolos engloba uma série de protocolos que permitem a interconexão e comunicação de redes, locais ou não, independentemente da tecnologia de meio, fornecendo serviços a vários tipos diferentes de máquinas conectadas a redes heterogêneas.

A estrutura do conjunto de protocolos Internet pode ser conceituada como uma série de quatro camadas, figura 2.15.

Figura 2.15 -Estrutura em camadas do conjunto de protocolos Internet.

Cada camada é projetada para lidar com as comunicações num modo pré-determinado. Assim cada protocolo formata dados de comunicação e anexa ou remove informações, e o pacote é passado para a camada inferior de uma estação emissora, ou para a camada superior na estação receptora [Co91a].

Embora o modelo de referência ISO/OSI e o conjunto de protocolos Internet sejam utilizados para prover a comunicação entre redes heterogêneas, o conjunto de protocolos Internet é mais simples e menos formal que o modelo de referência ISO/OSI, o que o torna mais fácil de ser compreendido e implementado. Adicionalmente, existe uma diferença marcante na filosofia de suas estruturas. No modelo de referência, todas as camadas lidam com tratamento de erros, enquanto no conjunto Internet, a confiabilidade é tratada a nível de transporte e aplicação [Co91a].

2.5.1 Camada de Interface de Rede

A Camada de Interface engloba as duas primeiras camadas do modelo de arquitetura ISO/OSI, camada física e de enlace de dados, sendo portanto responsável pela transmissão dos **datagramas IP**, unidade básica de transferência nos protocolos Internet, sobre uma rede física [Ro91].

Basicamente os serviços oferecidos por essa camada são:

- mapeamento dos endereços IP sobre endereços específicos de interface; e
- encapsulamento de datagramas IP para transmissão sobre um único meio específico.

Para demonstrar como a camada de interface realiza as tarefas citadas acima, tome-se como exemplo a transmissão de datagramas IP sobre uma rede Ethernet.

Encapsulamento do datagrama IP em um pacote Ethernet

O datagrama IP é transmitido na porção de dados de um pacote Ethernet, onde o campo tipo de pacote indica o tipo de protocolo como sendo IP.

Como visto anteriormente, item 2.4.1 sobre a Ethernet, a porção de dados tem limite máximo de 1500 bytes, e mínimo de 46 bytes. Caso o tamanho do datagrama IP ultrapasse o máximo, ele deve ser fragmentado, e se for menor que o mínimo, ocorre o *padding* (preenchimento da porção de dados com *bytes* nulos após o datagrama IP). Essas atividades são realizadas transparentemente para o IP [Ro91].

Mapeamento de endereços

O mapeamento do endereço Internet destino para o endereço Ethernet destino é feito dinamicamente através do protocolo *Address Resolution Protocol* (ARP) (protocolo de resolução de endereços) [So91] [Co91a] [Ro91].

Quando a camada de interface recebe um datagrama IP para ser enviado a uma determinada máquina conectada na Ethernet, o primeiro passo é consultar o *cache* ARP ou tabela ARP, uma tabela residente em memória onde cada entrada apresenta o endereço IP de uma máquina e o seu respectivo endereço Ethernet. Essa tabela é preenchida dinamicamente pelo ARP à medida que novos endereços Ethernet são necessários [So91].

endereço IP	endereço Ethernet
223.1.2.1	08:00:39:00:2f:c3
223.1.2.3	08:00:5a:21:a7:22
223.1.2.4	08:00:10:99:ac:54

Figura 2.16 -Tabela ARP simplificada.

Caso o endereço IP se encontre mapeado, o datagrama IP é transmitido em um pacote Ethernet para o endereço físico correspondente. Caso contrário a camada de interface:

1. constrói um pacote ARP de pedido e o envia em um pacote Ethernet no modo *broadcast*; e
2. enfileira o datagrama IP.

Figura 2.17 -Formato do pacote ARP.

Embora se tenha escolhido a Ethernet como exemplo, o ARP é projetado para ser utilizado com uma variedade de tecnologias, e conseqüentemente manipula endereços físicos e de protocolos arbitrários. Portanto, há a necessidade de se ter campos indicando o comprimento desses endereços [Ro91].

Na figura 2.17 o campo hardware type indica o tipo de interface de hardware (valor 1 para a Ethernet). O campo protocol type especifica qual protocolo de rede está sendo traduzido; no caso do IP o valor utilizado é o mesmo para o pacote Ethernet (0x0800). O campo operation indica um pedido ARP (1) ou uma resposta ARP (2). Os demais campos já são especificados pelos seus nomes, sendo que o único a não ser preenchido pelo emissor é o campo target IP address.

Uma vez devidamente preenchido, o pacote ARP é encapsulado no pacote Ethernet e enviado pela rede.

Quando a camada de interface recebe um pacote ARP, verifica se o tipo de hardware e protocolo são suportados. Se não forem, o pacote é descartado, caso contrário:

1. verifica se o endereço de protocolo do emissor já está na tabela ARP local, se estiver atualizá-la;
2. verifica se o endereço do protocolo destino corresponde ao endereço IP local, se não, o pacote é descartado;
3. caso contrário

(a) se o endereço de protocolo do emissor não consta na tabela ARP, esse endereço juntamente com o endereço de hardware do emissor é adicionado à tabela local;

(b) se o campo de operação indicar um pedido ARP, a camada de interface gera um pacote de resposta ARP aproveitando-se do pacote enviado, permutando os campos fonte/destino e atualizando o campo operation para o valor 2 (resposta ARP). O pacote é enviado diretamente a quem fez o pedido, e não como *broadcast*.

Uma vez de posse do endereço físico procurado, o datagrama IP é desenfileirado, o endereço mapeado e o pacote enviado como de costume.

Nota-se que para cada interface Ethernet existe uma tabela ARP associada. Algumas implementações do IP e ARP não enfileiram o datagrama IP, ele é simplesmente descartado ficando a cargo do protocolo TCP ou da aplicação que se utiliza do UDP, lidar com a recuperação do datagrama [So91].

2.5.2 Camada Internet

A camada Internet equivale à camada de rede do modelo de referência ISO/OSI, deixando transparente para as camadas superiores a topologia da internet e o tipo de meio de transmissão utilizado em cada rede física compreendendo a Internet. Para isso, realiza os seguintes serviços [Ro91]:

- um serviço de transporte independente do meio de transmissão;
- um mecanismo de endereçamento global; e
- um esquema de roteamento para transferir dados através das redes que compõem a Internet.

Esses serviços permitem que a Internet seja visível como uma rede homogênea virtual.

Serviço de transporte

A camada Internet oferece o serviço de datagrama. Os dados recebidos da camada superior são agrupados em pacotes denominados datagramas IP e, devido ao tipo de transporte oferecido, são enviados independentemente uns dos outros ficando a cargo das camadas superiores cuidar dos possíveis erros gerados pelo fato do serviço não ser confiável.

Endereçamento

Uma estação conectada à Internet é identificada através do seu endereço IP, composto por 32 bits, sendo uma parte utilizada para identificar a rede à qual a estação está conectada (identificador de rede), e a outra parte utilizada para identificar a própria estação (identificador de estação).

O endereço Internet, ou IP, apresenta três classes primárias baseadas na divisão dos bits entre identificador de estação e identificador de rede.

Existem três classes primárias de endereçamento Internet, cada uma com um número diferente de bits alocados para o identificador de rede e o identificador de estação.

Figura 2.18 -As três classes primárias de endereços Internet.

Os endereços da classe A podem endereçar 2^7 redes da classe A, com cada uma contendo $2^{24}-2$ máquinas; os da classe B endereçam 2^{14} redes classe B com até $2^{16}-2$ máquinas; e os da classe C endereçam 2^{21} redes classe C, com cada uma possuindo no máximo 2^8-2 máquinas.

Na prática, uma rede identificada pelo identificador de rede pode ser composto por

várias sub-redes, que são endereçadas utilizando uma hierarquia de endereçamento de três níveis, onde o identificador de estação é dividido em número da sub-rede e número da estação. A identificação desses números é realizada através de uma máscara AND [Ro91].

Figura 2.19 -Exemplo de uma máscara de sub-rede.

Para facilitar a manipulação de endereços Internet por parte do usuário, adota-se uma notação decimal onde cada octeto do endereço é separado por pontos e representado por um inteiro. Exemplificando, o endereço 10000000 00001010 00000010 00011110 na notação decimal é representado por 128.10.2.30 [Co91a].

Roteamento

Basicamente existem dois tipos de roteamento: **roteamento direto** onde as estações fonte e destino estão localizadas na mesma rede IP e **roteamento indireto** onde as estações fonte e destino localizam-se em redes IP diferentes. O algoritmo de roteamento utiliza uma tabela de roteamento IP a qual armazena informações sobre possíveis destinos e como chegar a eles [Co91a].

rede	flag direto/indireto	roteador	nro. interface
223.1.2	direto	<branco>	1
223.1.3	indireto	223.1.2.4	1
223.1.4	indireto	223.1.2.4	1

Figura 2.20 -Tabela simples de roteamento.

O Protocolo IP (*Internet Protocol*)

O protocolo IP define o formato do datagrama IP, e o conjunto de regras que especificam como o datagrama deve ser processado e como os erros devem ser manipulados [Co91a].

Figura 2.21 -Formato do datagrama IP.

Ao receber dados da camada de transporte, o IP os encapsula em um datagrama IP e ao enviá-lo verifica o MTU (*Maximum Transfer Unit*) da interface associada para ver se há necessidade de fragmentação. Se houver, informações a esse respeito são colocadas no cabeçalho. Baseado no endereço destino, o IP realiza o roteamento verificando se o datagrama deve ser transportado diretamente, ou para um roteador.

Na recepção, o IP utiliza as informações do cabeçalho para verificar a validade do datagrama, identificar a transmissão e verificar se é um fragmento ou não. Se a transmissão foi fragmentada, o IP reagrupa os fragmentos no datagrama original e o passa para a camada superior, sem o cabeçalho IP.

O Protocolo ICMP (*Internet Control Message Protocol*)

Esse protocolo permite que estações e roteadores enviem mensagens de erro ou de controle a outras estações e roteadores. É utilizado juntamente com o IP para suprir a deficiência deste em relação aos mecanismos de recuperação de erros [Co91a] [Ro91].

Embora inicialmente projetado para fornecer um modo de roteadores transmitirem as causas de erros no transporte (roteador congestionado, estação receptora fora do ar, etc) à estação, o ICMP pode ser utilizado para comunicação estação a estação (ex. *ping*).

2.5.3 Camada de Transporte

Essa camada fornece serviço de comunicação entre aplicativos rodando em máquinas separadas. Tal comunicação é frequentemente denominada fim-a-fim. Dois protocolos estão neste nível: TCP e UDP.

Protocolo UDP (*User Datagram Protocol*)

O protocolo UDP permite a transferência de mensagens no modo sem conexão, e sem a garantia de transporte ou sequência [Co91a] [Ro91]. Utiliza o serviço do IP, onde o campo *protocol* do datagrama IP possui o valor 17, e a porção de dados engloba o

datagrama UDP.

Figura 2.22 -Formato do datagrama UDP.

O endereçamento da camada de aplicação é realizado através de inteiros de 16 bits denominados **portas**. A combinação de um endereço IP e um número de porta denomina-se *socket Internet*, e identifica unicamente uma entidade de aplicação [Ro91].

Protocolo TCP (*Transmission Control Protocol*)

O protocolo TCP [Co91a] [Ro91] especifica o serviço orientado a conexão do conjunto de protocolos Internet, conseqüentemente apresenta três fases: estabelecimento de conexão, transmissão dos dados e liberação da conexão.

É um protocolo confiável que aceita mensagens arbitrariamente longas do processo de aplicação e as quebra em segmentos de no máximo 64 kbytes. Esses são enviados no formato da figura 2.23.

Figura 2.23 -Formato do segmento TCP.

Esse segmento é encapsulado em um datagrama IP, onde o campo *type of service* possui o valor 6.

No TCP, uma conexão é definida como sendo um par de *sockets Internet*. A quádrupla endereço IP fonte, porta TCP fonte, endereço IP destino e porta TCP destino, identifica unicamente uma **conexão TCP** [Ro91].

2.5.4 Camada de Aplicação

A Camada de Aplicação contém os processos de aplicação que podem utilizar o protocolo TCP ou o UDP dependendo da confiabilidade requerida e das características físicas da rede sobre a qual os protocolos estão implementados [So91].

Atualmente existe uma série de aplicações desenvolvidas, dentre as quais pode-se destacar o TELNET, FTP, NFS, SNMP, SMTP e XWindow.

Figura 2.24 -Estrutura dos protocolos Internet e algumas aplicações.

2.6 Dispositivos de Interconexão de Redes

Através da implementação do modelo de referência ISO/OSI e dos protocolos TCP/IP, é possível a comunicação entre redes fisicamente heterogêneas. Nesse item são apresentados os dispositivos utilizados na conexão física a fim de se formar uma internet.

No contexto do Modelo de Referência ISO/OSI, o dispositivo que provê a interconexão entre redes, denomina-se *relay system*. O *relay system* que realiza apenas as funções alocadas para as três camadas inferiores (física, enlace de dados e rede), denomina-se *sistema intermediário (intermediate system)* [Sc92], usualmente conhecido como roteador. O *relay system* que opera a nível de MAC denomina-se *MAC bridge* [Cu90], usualmente denominado simplesmente de *bridge* ou ponte. A nível de camadas superiores, o *relay system* denomina-se *gateway*, e é responsável por mapeamentos complexos, tais como, as traduções entre os serviços oferecidos por pares diferentes de protocolos [Ro90c] por exemplo, um gateway entre FTAM e FTP.

Na comunidade Internet, costuma-se utilizar o termo *gateway* como roteador. Essa terminologia é adotada nos próximos capítulos desta tese, utilizando os dois termos para designar o mesmo dispositivo de interconexão.

2.6.1 Pontes (Bridges)

As pontes são dispositivos físicos *store-and-forward* (armazena-e-envia) que fornecem interconexão a nível de acesso ao meio (MAC). Sua função é filtrar e transmitir pacotes entre as redes locais que conecta [Be88].

A figura a seguir apresenta a operação de uma ponte bilateral [Se88].

Figura 2.25 -Operação de uma ponte a nível de camadas OSI.

A filtragem do tráfego é realizada através de uma tabela que faz a correspondência entre os endereços das estações e os canais de comunicação [Ta89] [Se88]. Considere a figura a baixo,

Figura 2.26 -Uma configuração com quatro LANs e duas pontes.

As pontes operam no modo promíscuo, ou seja, têm acesso a todos os pacotes que circulam pelas redes conectadas a elas. Quando a ponte 1 recebe um pacote da LAN1 cuja estação destino também se localiza na mesma LAN, esse pacote é descartado uma vez que já está na LAN correta; caso contrário, se a estação destino estiver na LAN 2, 3, ou 4, a ponte 1, pela tabela sabe que o pacote deve ser transmitido para a LAN2.

O fato das pontes filtrarem o tráfego entre as redes permite que, se utilizadas adequadamente, possibilitem uma distribuição balanceada da carga nas redes, pois só passam adiante os pacotes que realmente se destinam ao outro segmento de rede; além de aumentarem a segurança de certos segmentos restringindo-lhes o acesso.

A compatibilidade entre MACs diferentes oferecida pelas pontes não é total,

existem problemas devido principalmente à inexistência de um padrão no formato de pacotes. Adicionalmente, pode haver diferenças nas taxas de transferência e nos *timers* das camadas superiores [Ta89].

A ponte é um dispositivo importante no gerenciamento de redes devido a sua capacidade de armazenar dados estatísticos, a nível de MAC, para cada uma de suas interfaces de rede. Adicionalmente, graças a sua operação promíscua, pode verificar a utilização da rede, e realizar estatísticas de erros, taxa de pacotes e de bytes por cada segmento de rede, por estação específica, ou por pacote selecionado [Be88].

2.6.2 Roteadores (*Gateways*)

Os roteadores, ou *gateways*, são dispositivos que fornecem interconexão a nível de rede [Ta89].

Figura 2.27 -Operação de um roteador a nível de camadas OSI.

Além de operarem em camadas diferentes, os roteadores diferem das pontes nos seguintes aspectos: a ponte opera transparentemente para as estações, ao passo que o roteador é explicitamente endereçado quando há a necessidade de roteamento; os roteadores podem tomar decisões de roteamento levando em conta questões como congestionamento ou número de roteadores até o destino [Se88].

O roteamento nos roteadores pode ser estático ou dinâmico dependendo da forma como a tabela de roteamento é atualizada. No roteamento estático os caminhos para as diversas redes são configurados manualmente pelo gerente da rede. Embora se tenha o controle direto do roteamento, ele não é adequado às grandes *internets* onde as topologias estão frequentemente sujeitas a mudanças. Nesse caso, o roteamento dinâmico é o mais adequado uma vez que novas rotas podem ser "aprendidas", possivelmente adicionadas às rotas estáticas e atualizadas caso ocorra alguma mudança na configuração da *internet* [Be88]. Alguns protocolos de roteamento bastante utilizados atualmente são: RIP [Co91a], IGRP e OSPF [Re93].

Os roteadores são importantes fontes de informação no gerenciamento de canais de comunicação e conseqüentemente no gerenciamento de redes. Através de estatísticas sobre

o tráfego na rede é possível ao gerente antecipar e controlar problemas de congestionamento.

CAPÍTULO 3

Gerenciamento de Redes de Computadores

3.1 Introdução

Através dos benefícios oferecidos pelas Redes de Computadores e da diminuição do custo do hardware, o número de computadores interconectados começou a crescer dentro das organizações originando problemas administrativos. As tarefas de configuração, localização de falhas e gerenciamento de dispositivos e recursos da rede passaram a consumir tempo, elevando os gastos dentro das organizações. Paralelamente, as redes começaram a ser interconectadas rapidamente; redes locais acoplavam-se a redes regionais, as quais por sua vez, interconectavam-se a *backbones* nacionais. Para se ter uma idéia da velocidade com que as interconexões ocorriam, na metade da década de oitenta o número de redes conectadas à Internet chegou quase a dobrar de ano para ano [Ro91].

Cientes desses problemas, a ISO e a IAB (*Internet Activity Board*), iniciaram pesquisas buscando soluções que permitissem aos gerentes de rede realizar tarefas tais como obter dados sobre performance e tráfego da rede em tempo real, diagnosticar problemas de comunicação e reconfigurar a rede atendendo a mudanças nas necessidades do usuário e no ambiente [Kl88]. Porém, vários obstáculos teriam que ser superados, entre eles a heterogeneidade dos equipamentos de rede (computadores, roteadores e dispositivos de meio), dos protocolos de comunicação e das tecnologias de rede. Adicionalmente, as internets englobavam redes que estavam sob diferentes domínios administrativos [Ro91].

A solução da ISO veio através do desenvolvimento de uma arquitetura de gerenciamento de redes ISO/OSI, mais conhecida pelo seu protocolo de gerenciamento CMIP (*Commom Management Information Protocol*) [Br93] [Kl88] [Wa89]. Para o gerenciamento das redes baseadas em TCP/IP, duas estratégias foram tomadas. A solução a curto prazo deu-se através do protocolo SNMP (*Simple Network Management Protocol*) [Ca90] [Ro91], que se tornou rapidamente um sucesso, sendo adotado por vários fabricantes de produtos de rede e pesquisadores. Na solução a longo prazo, a arquitetura de gerenciamento OSI foi mapeada sobre o serviço de transporte TCP, dando origem ao CMOT (*CMIP Over TCP*) [Wa89].

Pensando na transição entre as duas estratégias, foram desenvolvidos dois documentos relacionados com as informações de gerenciamento, a fim de assegurar a compatibilidade entre o SNMP e o CMOT. No documento sobre a SMI (*Structure of Management Information*) estão descritas regras para definição de objetos gerenciados de modo que fiquem independentes do protocolo de gerenciamento. Já o documento sobre a MIB (*Management Information Base*) descreve os objetos que podem ser acessados via um protocolo de gerenciamento. Embora inicialmente a compatibilidade fosse mantida, surgiram divergências entre as duas linhas de pesquisa e a necessidade de compatibilidade foi suspensa. Isso possibilitou ao SNMP responder a novas necessidades operacionais da comunidade Internet, incluindo a adição de novos itens à MIB. Atualmente, a estrutura de gerenciamento Internet consiste nos documentos sobre SMI [Ro90a], MIB [Mc90] e SNMP [Ca90].

Nesse capítulo são abordados os principais aspectos da estrutura de gerenciamento Internet e da arquitetura OSI de gerenciamento.

3.2 A Estrutura de Gerenciamento Internet

Tendo origem na segunda metade da década de oitenta como uma expansão do protocolo SGMP (*Simple Gateway Monitoring Protocol*) [Da87], utilizado no gerenciamento de roteadores IP, o SNMP tornou-se o protocolo de gerenciamento escolhido pela indústria de informática constituindo, atualmente, um padrão operacional "de facto" [Ro91].

Grande parte do sucesso do SNMP deve-se a sua simplicidade, sendo um protocolo envia/recebe (*send/receive*) com apenas quatro operações. Outro aspecto importante é sua capacidade de gerenciar redes heterogêneas constituídas de diferentes tecnologias, protocolos e sistemas operacionais. Dessa forma o SNMP pode ser utilizado, por exemplo, para gerenciar redes Ethernet, Token Ring e FDDI conectando IBM PCs, Apple Macintoshs, estações de trabalho SUN e outros tipos de computadores [Sc90].

As aplicações de gerenciamento utilizam o SNMP para [Co91c]:

- fazer *polling* nos dispositivos de rede e coletar dados estatísticos para análise em tempo real;
- receber um conjunto limitado de notificações de eventos significativos ou mensagens *trap*; e
- reconfigurar dispositivos de rede.

Essas tarefas são realizadas através das operações disponíveis nesse protocolo.

3.2.1 Conceitos

O Modelo de Arquitetura

O Modelo de Arquitetura SNMP engloba três elementos: vários elementos de rede, pelo menos uma estação de gerenciamento de rede (*Network Management System-NMS*) e o protocolo de gerenciamento de rede (SNMP) [Ca90] [Ro91].

Os elementos de rede referem-se a dispositivos que pertencem a uma das três categorias descritas abaixo:

- sistema *host*, tal qual uma estação de trabalho, servidor de terminal ou impressora;
- sistema *gateway* (roteador); ou
- dispositivo de meio como por exemplo uma ponte, repetidores ou hubs em geral.

Cada elemento de rede possui implementado um **agente de gerenciamento**, ou **agente SNMP**, responsável por realizar as funções de gerenciamento requisitadas pelas estações de gerenciamento.

As estações de gerenciamento de rede são computadores, geralmente estações de trabalho, nos quais rodam o protocolo de gerenciamento e aplicações de gerenciamento de rede responsáveis por monitorar e controlar os elementos de rede. A aplicação de gerenciamento é também conhecida como **gerente SNMP**, ou simplesmente **gerente**.

O protocolo de gerenciamento de rede é responsável por definir a comunicação entre a estação de gerenciamento e os elementos de rede.

Figura 3.1 -Sistema de gerenciamento SNMP.

Objetivos do Modelo de Arquitetura

O modelo de arquitetura Internet minimiza explicitamente o número e a complexidade das funções de gerenciamento realizadas pelos agentes de gerenciamento. Esse objetivo é atrativo por diversas razões, entre elas: o custo de desenvolvimento do software do agente necessário para suportar o protocolo é conseqüentemente reduzido, e conjuntos simplificados de funções de gerenciamento são facilmente compreendidos e utilizados pelos projetistas das ferramentas de gerenciamento de redes [Ca90]. Um outro objetivo é que a arquitetura SNMP seja, tanto quanto possível, independente da arquitetura e mecanismos de *hosts* ou *gateways* particulares

Gerenciamento Proxy

No contexto da Estrutura de Gerenciamento Internet, o protocolo de gerenciamento utilizado pelas estações de gerenciamento e pelos elementos de rede faz uso do protocolo de comunicação TCP/IP. Surge então o problema de como gerenciar dispositivos, tais como repetidores e pontes, que não conseguem suportar pelo menos um conjunto mínimo do protocolo Internet, ou *hosts* e *gateways* que utilizam outro conjunto de protocolos. A solução vem através de um agente especial denominado *agente proxy* [Ro91] [Sc90b].

Quando surge a necessidade de se gerenciar os tipos de dispositivos descritos acima, a estação de gerenciamento contata o agente *proxy* indicando qual o dispositivo a ser gerenciado. O agente *proxy* recebe as requisições do gerente em SNMP, traduz essas requisições para o protocolo nativo do dispositivo e as passa para este (figura 3.2). A interação agente-*proxy*-gerente é análoga.

Figura 3.2 -Gerenciamento através do agente *proxy*.

O *proxy* pode também ser utilizado para *caching* de informações de gerenciamento, armazenando as informações de elementos de rede que não alteram freqüentemente. Dessa forma, a estação gerenciadora interage com o agente *proxy*, o qual passa a ser o gerente-*proxy* em relação aos elementos de rede, diminuindo a carga de processamento nesses elementos [Ro91].

Representação das Informações de Gerenciamento

As informações de gerenciamento manipuladas pelo SNMP podem requerer estruturas de dados relativamente complexas, necessitando de uma forma de representação não ambígua e independente de restrições de máquina. Para isso, é utilizado um subconjunto da linguagem formal ASN.1 (*Abstract Syntax Notation One*) [IS88a] [Kn88]

[Ro91], desenvolvida pela ISO. Essa linguagem formal apresenta uma gramática bem definida, possuindo tipos primitivos de dados (*boolean*, *integer*, *octet string*, *bit string*, *NULL*), que podem ser agrupados na construção de tipos mais complexos. Inclui o conceito de módulo, um conjunto de descrições ASN.1 relacionadas a um tema comum (especificação de um protocolo, de uma base de informações de gerenciamento, etc) [Ro91]. Cada módulo possui um nome através do qual outros módulos podem referenciá-lo para utilizar os tipos nele definidos.

Particularmente no gerenciamento, a linguagem ASN.1 apresenta dois propósitos distintos [Ro91]:

- definir formalmente os formatos dos PDUs trocados pelo protocolo de gerenciamento SNMP; e
- definir os objetos que serão gerenciados.

3.2.2 Objetos Gerenciados

SMI Internet

A SMI Internet (*Structure and Identification of Management Information for TCP/IP-based internets*) [Ro90a] descreve as estruturas comuns e o esquema de identificação para a definição de informações de gerenciamento utilizadas no gerenciamento de redes TCP/IP. Adicionalmente inclui a descrição de um modelo de informação para o gerenciamento de redes, juntamente com o conjunto de tipos genéricos utilizados para modelar a informação de gerenciamento. As descrições formais das estruturas são dadas utilizando a linguagem ASN.1.

Os objetos gerenciados são recursos monitorados e controlados por um ou mais sistemas de gerenciamento [Kl88], e são especificados pela SMI através de um nome, uma sintaxe, o nível de acesso (*read-only*, *read-write*, *write-only*, *not-accessible*), e o status (*mandatory*, *optional*, ou *obsolete*) [Ca90]. O nome, ou identificador de objeto, é definido através do tipo ASN.1 OBJECT IDENTIFIER, o qual fornece um meio de se identificar unicamente o objeto. Esse tipo é uma sequência de inteiros não negativos obtidos percorrendo-se um ramo da árvore, descrita na figura 3.3, de cima para baixo. Essa árvore na verdade é um ramo de uma outra árvore maior, onde o nó raiz apresenta três nós subordinados:

- ccitt(0), administrado pelo CCITT,
- iso(1), administrada pela ISO/IEC;
- joint-iso-ccitt(2): administrada conjuntamente pela ISO/IEC e CCITT.

Figura 3.3 -Árvore de identificação ASN.1.

A sintaxe define as estruturas de dados que modelam o objeto. Essas estruturas são construídas utilizando-se tipos ASN.1 e são definidas na SMI. A codificação dessas estruturas é realizada utilizando a sintaxe de transferência que complementa a ASN.1, o BER (*Basic Encoding Rules*) [IS88b] [Ro91]. Esta sintaxe de transferência fornece um algoritmo especificando como um valor de qualquer tipo de estrutura de dados, definida utilizando ASN.1, deve ser codificado para transmissão.

MIB Internet

A MIB Internet (*Management Information Base for Network Management of TCP/IP-based Internets*), definida em [Mc90] e também conhecida como MIB-I, contém as definições dos objetos gerenciados, seguindo os mecanismos descritos na SMI. Consequentemente, na MIB os objetos já estão devidamente especificados. Por exemplo, o objeto *ifNumber* (número da interface), pertencente ao grupo *Interface*, possui o identificador 1.3.6.1.2.1.2.1, obtido percorrendo-se a árvore da figura 3.3; o tipo de dado ASN.1 que o modela é o *integer*, o nível de acesso é *read-only*, e o status é *mandatory* (todos os elementos de rede devem implementá-lo).

Todos os dispositivos que são elementos de rede devem implementar a MIB, porém, nem todos os dispositivos possuem implementadas todas as funções do conjunto de protocolos. Por essa razão, a MIB divide os objetos em oito grupos; caso um elemento de rede contenha um objeto de um grupo, ele deve implementar todos os objetos desse grupo [Ro91].

A MIB-II [Ro90b] inclui mais dois grupos e outros objetos, mantendo compatibilidade com a SMI e a MIB-I, a fim de atender a novas necessidades operacionais, e fornecer suporte para dispositivos multi-protocolos. Os grupos contidos na MIB-I, e os adicionais da MIB-II são:

System group: contém informações de configuração genéricas e deve ser implementado por todos os elementos de rede. Os objetos desse grupo armazenam informações tais como nome do dispositivo e a quanto tempo o agente está no ar.

Interface group: contém informações genéricas das entidades na camada de interface e deve ser implementado por todos os elementos de rede. São exemplos de objetos desse grupo, informações como tipo de interface, taxa de transmissão e endereço físico.

Address translation group: implementado por todos os elementos de rede, contém informações de resolução de endereços tais como número da interface, endereço físico e endereço Internet. Tornou-se obsoleto na MIB-II.

IP group: contém informações referentes ao protocolo IP, tais como datagramas descartados devido a erros, número de datagramas passados para a camada inferior, máscara IP, etc. Contém também informações de roteamento tais como a tabela IP. Deve ser implementado por todos os elementos de rede.

ICMP group: contém vinte e seis contadores que podem ser resumidos em:

a) para cada tipo de mensagem ICMP, existem dois contadores, um para o número de vezes que esse tipo de mensagem foi recebido pela entidade IP local e o outro para o número de vezes que esse tipo de mensagem foi gerado pela entidade IP local;

b) quatro contadores adicionais para guardar o total de mensagens ICMP recebidas, enviadas, recebidas com erro ou não enviadas devido a erros. Deve ser implementado por todos os elementos de rede.

TCP group: contém informações referentes a entidades TCP, por exemplo, identificar qual o algoritmo de retransmissão, número de conexões, tempo máximo de *timeout* para retransmissão. Possui uma tabela que guarda informações referentes às entidades de aplicação que estão usando TCP; essa tabela contém cinco colunas: estado da conexão, endereço IP local, porta TCP local, endereço IP remoto e porta TCP remota. Deve ser implementado por todos os grupos que implementam o protocolo TCP.

UDP group: possui quatro contadores para informações referentes aos datagramas UDP (número de datagramas transmitidos para a camada acima, descartados devido a erros de formatos, recebidos da camada superior e destinados a portas desconhecidas), e uma tabela que armazena o endereço IP e a porta local das entidades de aplicação que se utilizam do protocolo UDP. Esse grupo deve ser implementado por todos os elementos de rede que implementam o protocolo UDP.

EGP group: esse grupo deve ser implementado por todos os elementos de rede que implementam o EGP (*Exterior Gateway Protocol*) [Mi84].

Transmission group: esse grupo descrito na MIB-II guarda informações sobre MIBs específicas do meio de transmissão.

SNMP group: nesse grupo, pertencente a MIB-II, estão descritos os objetos gerenciados para as entidades de aplicação SNMP, os quais armazenam informações referentes às quantidades de PDUs recebidas para cada tipo de erro, número de objetos da MIB recuperados, alterados, etc.

3.2.3 Estrutura Administrativa

A estrutura administrativa determina as políticas de autorização e autenticação utilizadas entre as entidades de aplicação SNMP [Ca90] [Ro91]. Essas entidades de aplicação possuem uma série de relações administrativas cujos conceitos são introduzidos a seguir.

Um conceito importante é o de **comunidade** (*community*), relação entre um agente SNMP e um ou mais gerentes. A comunidade é identificada através de uma cadeia (*string*) de octetos denominada **nome da comunidade** (*community name*), utilizada nos mecanismos de autenticação e autorização. Dessa forma, as mensagens SNMP trocadas entre agentes e gerentes contêm duas partes:

- nome da comunidade, além de outras informações necessárias para validar a entidade SNMP emissora como membro da comunidade identificada; e
- dados, contendo uma operação SNMP e os operandos associados [Ro91].

Autenticação

O mecanismo de autenticação disponível no SNMP é muito simples, não havendo nenhum mecanismo de criptografia para proteger o nome da comunidade. Uma entidade SNMP emissora é considerada autenticada como membro da comunidade, caso o nome da comunidade descrita na mensagem SNMP corresponda a uma comunidade reconhecida pela entidade SNMP receptora.

Autorização

Após a entidade emissora ter sido autenticada, é preciso verificar qual o nível de acesso que lhe é permitido. Além do modo de acesso definido juntamente com o objeto na MIB, existe o modo de acesso (*read-only*, *read-write*) definido pela comunidade para cada objeto pertencente a uma **visão** (*view*), um subconjunto de objetos visíveis para uma comunidade particular.

3.2.4 O Protocolo de Gerenciamento SNMP

Enfatizando o protocolo de comunicação propriamente dito, o SNMP é um protocolo *request/response* com apenas quatro operações [Ro91].

- *get*: utilizado para buscar informações de gerenciamento específicas;
- *get-next*: busca, transversalmente, informações de gerenciamento, ou seja, busca o valor da variável que sucede a especificada na operação (adequado para buscas em tabelas);
- *set*: manipula informações de gerenciamento; e
- *trap*: reporta eventos extraordinários.

Essas operações são efetuadas através de cinco primitivas (*get-request*, *get-response*, *get-next-request*, *set-request* e *trap*). Essas primitivas interagem de quatro formas [Ro91]:

1)_ O gerente busca informações de gerenciamento no agente.

Figura 3.4 -Buscando informações de gerenciamento no agente.

Através da primitiva *get-request*, o gerente pode obter as instâncias das variáveis contidas na primitiva. Caso a instância não exista, uma *get-response* é retornada assinalando o erro; caso contrário as instâncias são retornadas.

2)_ O gerente atravessa uma porção da *view* do agente.

Figura 3.5 -Buscando informações de gerenciamento transversalmente no agente.

Para cada uma das variáveis na primitiva *get-next-request*, a instância que a sucede é retornada através do *get-response*. Caso o final da visão seja alcançado, o *get-response* retorna um erro.

3)_ O gerente atualiza informações de gerenciamento através do agente.

Figura 3.6 -Atualizando informações de gerenciamento no agente.

O gerente lista na *set-request* as variáveis que devem ser atualizadas, juntamente com os seus valores. Caso não haja nenhum erro, a *get-response* retorna idêntica ao *set-request*; caso contrário retorna um erro assinalando a causa que pode ser: inexistência da instância, a instância existe mas não permite a escrita (*read-only*), a instância permite a escrita mas o valor fornecido no pedido tem erro de sintaxe ou de limite de valores.

4)_ O agente notifica um evento extraordinário.

Figura 3.7 -Reportando a ocorrência de um evento ao gerente.

Quando um evento extraordinário ocorre, o agente identifica os gerentes, determina para cada um deles uma comunidade apropriada e envia o *trap*.

Serviço de Transporte

Embora o SNMP utilize o serviço orientado a datagrama, principalmente o UDP, ele é um protocolo independente do serviço de transporte, podendo utilizar inclusive o serviço orientado a conexão, ou ser mapeado sobre o protocolo da camada de enlace de dados ou do MAC [Ro91].

3.3 A Versão 2 da Estrutura de Gerenciamento Internet - SNMPv2

A segunda versão da estrutura de gerenciamento Internet, SNMPv2 [Ca93d], veio atender a novas necessidades da comunidade Internet e suprir algumas deficiências da primeira versão. A descrição do SNMPv2 é apresentada através de uma comparação sucinta com a primeira versão.

A segunda versão da estrutura de gerenciamento Internet manteve os mesmos componentes da estrutura de gerenciamento da primeira versão: nós gerenciados que implementam o agente, pelo menos uma estação gerenciadora e o protocolo de gerenciamento. Mas apresenta alterações na estrutura da relação entre esses componentes, na estrutura de informação de gerenciamento (SMI) e nas operações do protocolo de gerenciamento. Algumas dessas alterações são apresentadas em linhas gerais.

Uma entidade SNMPv2 pode desempenhar dois papéis:

- agente: quando realiza operações de gerenciamento em resposta a mensagens SNMPv2 recebidas (que não sejam notificações) ou quando envia notificações de *traps*;
- gerente: quando inicia operações de gerenciamento SNMPv2 através da geração de mensagens SNMPv2 ou quando realiza operações de gerenciamento em resposta a notificações de informações ou *traps* recebidas.

A entidade pode desempenhar os dois papéis de acordo com a implementação e configuração, sendo que pode também atuar como um agente *proxy*.

Uma novidade em relação à versão passada é a capacidade de dois gerentes se comunicarem através da definição de uma nova primitiva, a *inform-request*. Através dessa primitiva um gerente pode, por exemplo, passar informações sobre as visões de sua MIB. Outra primitiva que foi adicionada ao conjunto definido na versão 1 é a *get-bulk-request*, permitindo que uma quantidade maior de informações sejam transmitidas por vez, suprimindo uma deficiência da primeira versão, permitindo assim uma busca rápida e eficiente em grandes tabelas [Ca93d].

O documento que descreve a SMI da versão 2, [Ca93a], vem dividido em três partes:

- definições de módulo: utilizados na descrição dos módulos de informação,
- definições de objeto: utilizados na descrição de objetos gerenciados,
- definições de notificação: utilizados na descrição de transmissões não solicitadas de informações de gerenciamento.

Foram definidas duas novas macros, MODULE-IDENTITY e NOTIFICATION-TYPE, para exprimir concisamente a sintaxe e semântica de um módulo de informação e de

uma notificação. E definidos dois novos nós filhos do nó internet: o nó *security* e o nó *snmpV2*.

O modelo administrativo do SNMPv2 abandona o modelo de estrutura baseada em comunidade apresentada na primeira versão. No novo modelo, os pares que trocam mensagens SNMPv2 são identificados distintamente, possibilitando a aplicação de um modelo de controle de acesso mais conveniente e permitindo uma utilização mais efetiva de protocolos de segurança no futuro. Os mecanismos de autenticação e segurança para a nova versão são definidos em [Ca93c].

Um aspecto importante a ser considerado, é a coexistência entre as duas versões da estrutura de gerenciamento Internet. Existem duas abordagens para se obter a coexistência a nível de protocolo. Uma delas sugere um mecanismo *proxy*, onde uma entidade SNMPv2, atuando como agente, é implementado e configurado para agir como um *proxy* entre as duas versões.

Figura 3.8 - Agente proxy entre SNMPv1 e SNMPv2.

A outra abordagem propõe que a entidade SNMPv2, agindo como gerente, dê suporte às duas versões transparentemente para a aplicação de gerenciamento. Dessa forma, antes de se comunicar com a entidade SNMPv2 que faz o papel de agente, o gerente consulta uma base de dados local para selecionar a versão correta do protocolo a ser usado.

3.4 A Arquitetura OSI de Gerenciamento

Preocupada com a questão de gerenciamento de redes, a ISO propôs em 1989 uma arquitetura de gerenciamento capaz de viabilizar o controle e monitoração de redes que utilizam protocolos OSI. Dessa forma, o modelo de referência ISO/OSI foi estendido para acomodar capacidade de gerenciamento, e foram especificados protocolos para o transporte de informações de gerenciamento entre sistemas abertos. Dentre esses destaca-se o CMIP (*Common Management Information Protocol*), utilizado tanto no gerenciamento de recursos OSI quanto no gerenciamento de recursos reais de rede [K188].

Nos itens a seguir são apresentados os modelos de gerenciamento e os

componentes da arquitetura OSI referida acima: estrutura, componentes de gerenciamento, estrutura da informação de gerenciamento e serviços e protocolos para a troca de informações de gerenciamento.

3.4.1 Os Modelos de Gerenciamento

O gerenciamento OSI é descrito tipicamente através de três modelos [Kl88] [Wa89]: modelo organizacional, modelo funcional e modelo de informação.

O Modelo Organizacional

Esse modelo introduz o conceito de **domínio** de gerenciamento, uma partição administrativa de uma rede ou internet para fins de gerenciamento. Esse conceito é importante por razões de escala, segurança e autonomia da rede. Nesse contexto, o modelo organizacional descreve os modos nos quais o gerenciamento OSI pode ser distribuído administrativamente através de vários domínios de gerenciamento e sistemas de gerenciamento dentro de um domínio.

O Modelo de Informação

O Modelo de Informação fornece diretrizes para definir os objetos gerenciados e as informações de gerenciamento associadas a eles. Esse modelo engloba uma base de dados para as informações de gerenciamento (MIB) e uma estrutura de informação de gerenciamentos (ISO SMI) [IS87] descritos no item 3.4.2.

O Modelo Funcional

A arquitetura de gerenciamento OSI define cinco áreas funcionais para prover as necessidades do usuário no gerenciamento de suas redes.

- **Gerenciamento de falhas:** fornece mecanismos para detectar, isolar e corrigir operações anormais do ambiente OSI e da rede de comunicação. Inclui também a capacidade de realizar seqüências de testes para fins de diagnóstico.
- **Gerenciamento de configuração:** permite a configuração de elementos de redes remotos. A alteração na configuração pode ser necessária para acomodar mudanças nas necessidades do usuário, aliviar congestionamentos na rede ou isolar falhas.
- **Gerenciamento de performance:** fornece mecanismos para medir, monitorar, avaliar e relatar os níveis de desempenho alcançados pela rede. Através da análise dos dados de performance, o gerente pode tomar decisões quanto a alterações na configuração de forma a manter a qualidade do serviço da rede.
- **Gerenciamento de segurança:** oferece mecanismos para o gerenciamento dos serviços de proteção (controle de acesso, autenticação, criptografia, etc)

- **Gerenciamento de contabilização:** oferece funções que permitem determinar os custos da utilização dos recursos da rede.

3.4.2 Componentes da Arquitetura

Neste item são apresentados os componentes da arquitetura OSI de gerenciamento.

A Estrutura de Gerenciamento OSI

O Gerenciamento OSI é realizado através de funções fornecidas pelo Gerenciamento de Sistemas, Gerenciamento de Camada e Operação de Camada [Br93] [Kl88].

- **Gerenciamento de Sistemas:** responsável por gerenciar o sistema local e, em conjunto com outros sistemas pares, gerenciar o meio físico de comunicação, os protocolos de comunicação e dispositivos associados.
- **Gerenciamento de Camada:** gerencia recursos relacionados com as atividades de comunicação de uma única camada, através de funções de apoio internas e de protocolos de propósito especial. Esses protocolos não prestam serviços às camadas superiores e são independentes dos protocolos de gerenciamento de outras camadas.
- **Operação de Camada:** gerencia uma única instância de comunicação (ex. uma conexão) em uma camada, utilizando o protocolo normal de cada camada para a troca de informações de gerenciamento.

Nesse contexto, o modelo de arquitetura é ilustrado como na figura 3.9.

Figura 3.9 -Modelo de arquitetura do gerenciamento OSI.

O processo de aplicação de gerenciamento de sistemas (*System Management Application Process - SMAP*) é um processo local responsável por executar funções de gerenciamento que asseguram a operação contínua do sistema como um todo, permitindo que se adapte a mudanças nas condições ambientais e nas necessidades do usuário. A interface específica para cada camada é obtida através das LMEs (*Layer Management Entity*), onde cada uma delas concentra a funcionalidade da camada de sua responsabilidade.

A entidade de aplicação de gerenciamento de sistema (*System Management Application Entity - SMAE*) concentra as funções de integração entre as LMEs e a função de interface com o gerente. Adicionalmente a SMAE providencia a interface das LMEs de um nó da rede com as LMEs pares em outro nó, através do protocolo CMIP [Br93]. O Protocolo de gerenciamento CMIP fornece um formato inteligível comum para a transferência de informação de gerenciamento entre entidades pares [Br93].

Componentes de Gerenciamento

Assim como na Estrutura de Gerenciamento Internet, a arquitetura de gerenciamento OSI apresenta três elementos: sistemas de gerenciamento, onde são executados os processos de aplicação chamados **gerentes**; sistemas gerenciados ou elementos de redes, onde residem processos de aplicação denominados **agentes** e **objetos gerenciados** que representam recursos sujeitos a gerenciamento [Wa89]. A figura 3.10 ilustra o sistema de gerenciamento CMIP.

Figura 3.10 - Sistema de gerenciamento CMIP.

De modo geral, a interação entre os componentes ocorre da seguinte forma: o gerente envia operações de gerenciamento aos agentes a fim de obter informações sobre objetos gerenciados e controlá-los; o agente por sua vez, recebe as operações e as executa sobre os objetos gerenciados. Além disso, o agente pode enviar ao gerente notificações geradas pelos objetos gerenciados, ou sobre a ocorrência de eventos extraordinários.

Sob o enfoque do Gerenciamento de Sistemas, as interações que ocorrem entre as entidades de aplicação de gerenciamento são modeladas como operações de gerenciamento e notificações, sendo que uma assume o papel de gerente e a outra o papel de agente, embora esses papéis não sejam fixos; ou seja, uma entidade que seja agente para uma entidade gerente, pode assumir o papel de gerente em relação a uma outra entidade de aplicação de gerenciamento [Sp93].

Os Serviços e o Protocolo OSI de Gerenciamento

A Estrutura de Gerenciamento OSI utiliza-se dos seguintes serviços e protocolos, para gerenciar sistemas abertos [KI88] [Wa89]:

- **ACSE** (*Association Control Service Element*): para que duas entidades de aplicação troquem informações de gerenciamento através do CMIP é necessário que se estabeleça primeiro uma associação entre eles. O ACSE é o responsável pelo estabelecimento e liberação dessas associações;
- **ROSE** (*Remote Operation Services Elements*): uma vez estabelecida a conexão, o CMIP utiliza os serviços fornecidos pelo ROSE para executar operações em um sistema remoto. O ROSE pode ser considerado o equivalente da ISO para o RPC (*Remote Procedure Call*);
- **CMISE** (*Common Management Information Service Element*): quando o gerente e o agente desejam trocar informações de gerenciamento, ambos utilizam os serviços do CMIS, implementados pelo CMISE, responsável por fornecer os serviços básicos de gerenciamento, notificação de eventos e busca e manipulação de dados de gerenciamento, com ou sem confirmação. O CMISE utiliza os serviços do ACSE e do ROSE.

Os serviços implementados pelo CMISE podem ser divididos em quatro categorias [KI88]:

- **Serviços de associação de gerenciamento**: esses serviços controlam o estabelecimento e a liberação, normal ou anormal, das associações de gerenciamento utilizando os serviços do ACSE.
- **Serviços de manipulação de dados**: são serviços para obter, modificar, adicionar ou remover dados da MIB.
- **Serviço de notificação de eventos**: permite a notificação de eventos para os gerentes ou

sistemas pares com ou sem confirmação.

- Serviços de controle direto: são utilizados para fazer com que um sistema de gerenciamento efetue uma determinada ação, por exemplo executar um teste de diagnóstico da rede.

Estrutura de Informação

A ISO SMI [IS87] especifica o modelo de informação adotado pelos sistemas de gerenciamento OSI, descreve o cenário na qual a MIB pode ser definida e define o conjunto de operações que podem ser realizadas sobre os objetos da MIB e o comportamento dos objetos quando executadas essas operações [Br93]. Adicionalmente introduz os conceitos de hierarquias de herança, nomeação e registro utilizados na caracterização e identificação dos objetos gerenciados.

Dentro desse contexto, a MIB é definida como um conjunto de objetos gerenciados dentro de um Sistema Aberto, no qual um objeto representa um recurso real desse sistema [Br93].

3.5 Produtos Comerciais para Gerenciamento de Redes de Computadores

No mercado de informática torna-se cada vez maior o número de produtos comerciais destinados ao gerenciamento de redes. Existem basicamente quatro tipos de ferramentas com esse fim [Da91]:

- ferramentas a nível físico que detectam problemas a nível de cabos e conexões de hardware, tais como cabos abertos e mau funcionamento em hardwares de conexão;
- monitores de redes, que se conectam às redes (um por segmento) monitorando o tráfego. Através do exame de informações a nível de pacotes, o monitor consegue compilar estatísticas referentes à utilização das redes, tipo de pacotes, número de pacotes enviados e recebidos por cada nó da rede, pacotes com erros e outras variáveis importantes;
- analisadores de rede, que auxiliam no rastreamento e correção de problemas encontrados nas redes. Para isso apresentam características sofisticadas para análise do tráfego na rede, captura e decodificação de pacotes e transmissão de pacotes em tempo real;
- sistemas de gerenciamento de redes integrados (SGRI), os quais permitem a monitoração e controle de uma rede inteira a partir de um ponto central. Esses sistemas

apresentam interface gráfica com o usuário, seguem o modelo gerente-agente e realizam gerenciamento de falhas, performance, configuração, segurança e contabilização.

O sistema MultView visa encaixar-se no quarto tipo de ferramenta, e para avaliar sua funcionalidade, estudou-se dois dos mais conhecidos SGRI: o AIX NetView/6000 da IBM Corporation e o SunNet Manager da Sun Microsystems.

3.5.1 SunNet Manager

O SunNet Manager é a plataforma de gerenciamento desenvolvida pela Sun Microsystems, cuja arquitetura baseia-se no modelo gerente-agente, utilizando como protocolo de transporte o TCP/IP [Su92b].

Existem dois tipos de agentes no contexto do SunNet Manager: um que acessa diretamente os objetos gerenciados, como por exemplo informações sobre o *hard disk*, utilização de memória, estatísticas de entrada e saída de discos, CPU e *tty*, status da impressora, estatísticas sobre RPC, NFS e sobre as camadas de protocolos; e outro que acessa indiretamente os objetos gerenciados, comunicando-se com um agente SNMP. Este último denomina-se agente proxy [Su92a].

A comunicação entre o gerente SunNet e os agentes é realizada através de uma biblioteca de serviços que fornece a infraestrutura de gerenciamento e lida com o serviço de comunicação. Os processos agente e gerente acessam os serviços através de APIs (*Application Programming Interface*) [St90], as quais usam RPC/XDR (*Remote Procedure Call/External Data Representation*) [Co93]. Um esquema ilustrando a comunicação entre o gerente SunNet e os agentes é apresentado a seguir:

Figura 3.11- Interação entre o gerente SunNet e os agentes.

A interface do gerente SunNet é gráfica seguindo o padrão OPEN LOOK GUI, sendo portanto baseada em janelas, menus botões e ícones. Nesta interface se encontram as

entradas para as funções de gerenciamento, e são apresentadas as informações textuais e gráficas. O SunNet Manager é capaz de descobrir dinamicamente e mostrar graficamente o mapa das redes gerenciadas. Existem duas possibilidades de visualizar as redes e seus componentes. A primeira denomina-se visão hierárquica e mostra hierarquicamente os vários níveis do ambiente das redes (redes e sub-redes). A segunda é a visão panorâmica, onde o administrador da rede pode visualizar recursos de rede, por exemplo os roteadores gerenciados na rede, independentemente de sua localização na hierarquia de rede. O SunNet Manager apresenta um editor gráfico que possibilita ao usuário editar o desenho dos mapas das redes gerenciadas.

Ainda considerando-se a parte gráfica, o SunNet apresenta uma ferramenta capaz de gerar gráficos comparativos, bi ou tri-dimensionais, na representação de dados dinamicamente ou informações armazenadas em arquivos (*logged*).

3.5.2 AIX NetView/6000

O AIX NetView/6000 é o sistema de gerenciamento para redes TCP/IP desenvolvido pela IBM Corporation. Também segue o modelo gerente-agente, onde o gerente é o sistema rodando o AIX NetView/6000 em uma estação RISC [Ne92].

A comunicação entre o gerente e os agentes é feita através do protocolo SNMP, como mostra a figura 3.12.

Figura 3.12 -Interação entre o gerente AIX NetView e os agentes.

A interface gráfica com o usuário baseia-se no padrão OSF/Motif, também

utilizando janelas, menus, botões e ícones. O AIX NetView/6000 descobre dinamicamente as redes dentro do domínio de gerenciamento, especificado pelo administrador da rede, e as apresenta graficamente, possuindo um mecanismo de layout automático que minimiza a necessidade de modificações manuais. Operações de edição são providas para modificar o mapa das topologias das redes, a fim de possibilitar a representação de informações que não puderam ser descobertas dinamicamente pelo sistema. A atualização de informações é feita automaticamente pelo sistema.

O AIX NetView/6000 apresenta ferramentas para testar conectividade de redes, encontrar rotas e requisitar informações em agentes SNMP remotos. Possibilita o acesso a aplicações de administração de sistemas remotos, através de uma interface para o SMIT (*System Management Interface Tool*) em nós rodando a versão 3, *release 2* do AIX; e para o SAM (*System Administration Manager*) para nós rodando HP-UX. Informações sobre a performance da rede são apresentadas através de um gráfico de linhas.

Tanto o SunNet Manager quanto o AIX NetView/6000 implementam todas as operações do protocolo SNMP e realizam tratamento de *traps*.

Embora não apresente o mesmo nível de funcionalidade dos dois sistemas comerciais, o MultView mostra a sua atualização com o estado da arte dos SGRIs comerciais, através da utilização de um protocolo padrão de gerenciamento, no caso o SNMP, da utilização de uma interface gráfica com o usuário, da capacidade de descobrir dinamicamente o mapa das redes gerenciadas, além das funções de monitoração.

Está previsto como trabalhos futuros a incorporação de algumas funções que constam no AIX NetView/6000 e no SunNet Manager e que são importantes para tornar o MultView mais completo, como por exemplo o editor gráfico, apresentação gráfica de dados estatísticos e de performance, implementação da operação *set* e tratamento de *traps* originados de agentes.

Uma característica do MultView que não se encontra nos dois sistemas apresentados, e que o diferencia dos demais, é a incorporação de funções de multimídia, como por exemplo a apresentação de imagens e transmissão de sons *on-line* ou gravados. Essa característica possibilita uma visão mais efetiva e completa das informações de gerenciamento e dinamiza a interação entre o administrador e as redes gerenciadas.

CAPÍTULO 4

Projeto do Sistema Monitor Gráfico de Redes

4.1 Introdução: Uma descrição geral do sistema

O Sistema Monitor Gráfico de Redes baseia-se no modelo de arquitetura SNMP descrito no capítulo 3, levando em consideração os elementos de rede que implementam agentes SNMP e o protocolo SNMP utilizado no transporte das informações de gerenciamento. Nesse contexto, o sistema monitor encaixa-se no papel de gerente SNMP, porém sem a capacidade de controlar os elementos de rede (não implementa a operação *set-request* do protocolo SNMP).

O objetivo básico do sistema monitor é apresentar dados sobre redes, *hosts* e *gateways* (informações estatísticas, de configuração, roteamento, mapas da rede, etc), para que, a partir da análise desses dados, o gerente possa tomar as decisões necessárias para garantir a operabilidade da rede.

Considerando o fator humano no projeto do sistema, buscou-se formas de facilitar a visibilidade das informações por parte do gerente e de simplificar o acesso às funções de monitoração. Nesse sentido, a interface com o usuário é baseada em janelas, seguindo o padrão XWindow, e os dados são apresentados graficamente, sempre que possível e conveniente.

Este capítulo especifica diversos aspectos do sistema, dentre eles:

- fontes de informações sobre a rede e seus componentes
- funções do sistema,
- interface do sistema com o usuário,
- estruturas de dados, e
- estrutura modular do sistema.

4.2 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 sobre as redes gerenciadas e seus componentes. 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 a utilização da estrutura de gerenciamento Internet, a MIB-II [Ro90b] apresenta uma série de objetos que podem ser úteis em algumas funções do sistema descritas no item 4.3. Os grupos da MIB-II foram avaliados e o resultado está descrito a seguir, apresentando os grupos dos objetos relevantes no contexto.

Grupo System

- *sysDescr*: pode incluir o nome e versão do tipo de hardware do sistema, sistema operacional, e software de rede;
- *sysUpTime*: o tempo em centésimos de segundos desde a última vez que a parte de gerenciamento de rede do sistema foi reinicializada;
- *sysLocation*: localização física do dispositivo;
- *sysContact*: nome da pessoa de contato;
- *sysName*: nome do dispositivo;
- *sysServices*: serviços oferecidos pelo dispositivo.

Grupo Interface

- *ifNumber*: número da interface de rede no qual o sistema pode receber e enviar datagramas IP;
- *ifDescr*: contém informações sobre a interface, incluindo nome do fabricante, nome do produto, versão da interface de hardware;
- *ifType*: tipo da interface, levando-se em consideração o(s) protocolo(s) de rede/enlace de dados/físico imediatamente abaixo do IP na pilha de protocolos (ex. ethernet-csma cd, iso-88025-token ring, fddi, rfc877-x25, T1-carrier, hyperchannel, etc);
- *ifMtu*: tamanho em octetos do maior datagrama IP que pode ser enviado/recebido na interface;
- *ifSpeed*: uma estimativa da *bandwidth* em bits por segundo;
- *ifPhysAddress*: endereço da interface na camada de protocolo imediatamente abaixo do IP na pilha de protocolos;
- *ifOperStatus*: estado operacional corrente da interface (*up*, *down*, *testing*);
- *ifInOctets*: número de octetos recebidos na interface desde o tempo em que o agente SNMP foi inicializado;
- *ifInDiscards*: número de pacotes recebidos e descartados, mesmo que nenhum erro tenha sido detectado, para prevenir a sua entrega para as camadas superiores (ex. para liberar espaço no buffer);
- *ifInErrors*: número de pacotes recebidos com erros;

- *ifOutOctets*: número total de octetos transmitidos pela interface desde o tempo em que o agente SNMP foi inicializado;
- *ifOutDiscards*: número de pacotes descartados, mesmo que nenhum erro tenha sido detectado, para prevenir a sua transmissão (ex. liberar espaço no buffer);
- *ifOutErrors*: número de pacotes não transmitidos por motivo de erros;
- *ifOutQlen*: comprimento da fila de saída de pacotes (em número de pacotes).

Grupo IP

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

Em relação à tabela de endereços IP, considerando que cada entrada na tabela corresponde a um endereço IP:

- *ipAdEntAddr*: o endereço IP;
- *ipAdEntNetmask*: a máscara de sub-rede associada ao endereço IP da entrada;

Em relação 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 unicamente a interface local através do qual o *next hop* para esta rota poderia ser alcançado;
- *ipRouteMetric1*: métrica de roteamento primária para a rota;
- *ipRouteMetric2*, *ipRouteMetric3*, *ipRouteMetric4*: métricas alternativas;
- *ipRouteNextHop*: endereço IP do próximo *hop* para a rota;
- *ipRouteType*: tipo de rota (direto, remoto, inválido, outros);
- *ipRouteProto*: mecanismo de roteamento a partir do qual a rota foi aprendida (local, icmp, egp, ggp, hello, rip, etc);
- *ipRouteMaks*: *netmask* para a rota;

Em relação à tabela ARP:

- *ipNetToMediaPhysAddress*: endereço físico do dispositivo;
- *ipNetToMediaNetAddress*: endereço Internet do dispositivo.

Grupo TCP

- *tcpMaxConn*: número máximo de conexões TCP suportadas pela entidade;

Em relação à tabela contendo informações específicas de conexões:

- *tcpConnState*: estado da conexão;
- *tcpConnLocalAddress*: endereço ip local da conexão;
- *tcpConnLocalPort*: número da porta local da conexão;
- *tcpConnRemAddress*: endereço IP remoto da conexão;
- *tcpConnRemPort*: número da porta remota da conexão;

4.3 Funções do Sistema Monitor

Considerando o objetivo básico do sistema monitor, definiu-se um conjunto mínimo de funções que fornecem dados sobre as redes gerenciadas e seus componentes.

- apresentação gráfica das redes gerenciadas (mapa das redes),
- apresentação gráfica dos componentes de uma rede,
- apresentação de informações sobre um *host*, *gateway* ou rede,
- verificação do status de *hosts* e *gateways*,
- apresentação de dados estatísticos,
- apresentação de tabelas de roteamento,
- apresentação de conexões TCP e UDP ativas entre a máquina onde está implementado o agente SNMP e outras máquinas, e
- entrada de informações complementares.

Estas funções envolvem outras funções responsáveis por obter dados sobre a rede e equipamentos, sempre dentro de um domínio de gerenciamento definido pelo gerente da rede.

4.3.1 Apresentação Gráfica das Redes Gerenciadas

A apresentação gráfica das redes dentro do domínio de gerenciamento (mapa das redes gerenciadas), mostrando as conexões através de *gateways*, e os endereços das sub-redes interconectadas é importante para dar ao gerente uma noção espacial das redes, como ilustra a figura 4.1.

A descoberta das redes que estão dentro do domínio e dos *gateways* que fazem as conexões é dinâmica, ou seja, o próprio sistema se encarrega de obter essas informações e fazer seu desenho da tela, sem a necessidade de interação com o gerente (usuário do sistema).

Na apresentação das redes deve-se permitir a entrada de eventos para a ativação de menus ou seleção de um determinado *gateway* ou rede, sobre a qual podem ser aplicadas outras funções (apresentação da tabela de roteamento, apresentação dos componentes, etc) dependendo do tipo de objeto selecionado.

Figura 4.1 - Apresentação do mapa das redes gerenciadas.

4.3.2 Apresentação Gráfica dos Componentes de uma Rede

Esta função é ativada selecionando-se, através do *mouse*, uma rede no desenho das redes gerenciadas, apresentando graficamente, em tela, os *hosts* e *gateways* conectados, juntamente com seus nomes e endereços Internet. Por exemplo, selecionando-se a rede 143.107.231.0, o desenho dos seu componentes seria como ilustrado abaixo.

Figura 4.2 - Apresentação dos componentes de uma rede.

A função de apresentação gráfica dos componentes também permite a entrada de eventos, nesse caso a seleção de um determinado *host* ou *gateway*. As funções disponíveis são as de apresentação de informações sobre o objeto selecionado: conexões ativas, tabela de roteamento, status do objeto e entrada de informações complementares.

4.3.3 Apresentação de Informações

Esta função na verdade engloba outras funções que apresentam informações sobre *hosts*, *gateways* e redes.

- apresentação de informações sobre *host* e *gateways* (nome, *alias*, endereço Internet, endereço físico, localização geográfica, tipo de equipamento, indicador sobre a presença de agente SNMP) e redes (nome, *alias*, endereço Internet, tipo de tecnologia -TCP/IP, Novell etc, número de *hosts* e *gateways* conectados). Essa função também deve permitir a entrada de dados pelo gerente (usuário) que não foram obtidas pelo sistema.
- verificação de status de *hosts* e *gateways*: através desta função, o gerente pode verificar, por exemplo, se *hosts* servidores (impressora, arquivos, etc) estão operacionais, e se determinada rede está incomunicável por motivo de *gateway* desligado. Essas informações devem ser apresentadas de forma adequada para uma identificação imediata por parte do gerente. Assim, são apresentadas graficamente no desenho das redes ou no desenho dos componentes de uma rede, por exemplo, pintando de vermelho ou fazendo piscar o ícone que representa o *hosts* ou *gateway* desligado.
- coleta e apresentação, textual ou gráfica (gráfico de barras, pizza ou linhas) de dados estatísticos.
- apresentação de tabelas de roteamento.
- apresentação de conexões ativas.

4.4 Interface do Sistema com o Usuário

Cartão de visitas de qualquer sistema, a interface com o usuário deve ser amigável, apresentando fácil manuseio e compreensão. Buscando essa premissa, na elaboração da interface optou-se pela utilização de janelas, menus e botões, como meio de entrada e apresentação de dados, os quais são acessados via *mouse*. Dessa forma, foram definidas algumas janelas básicas, descritas a seguir:

- janela principal cobrindo toda a tela e apresentando o nome e versão do sistema, e nome da instituição. Apresenta na parte lateral direita, ícones que dão acesso às funções do sistema monitor gráfico de redes apresentados no item 4.3, e a outros recursos previstos para o sistema de gerenciamento MultView (video, som e zoom de regiões do desenho do mapa das redes).

- janela para apresentação do desenho das redes gerenciadas. Contém menu para acesso às funções de estatística, status de *gateways*, tabelas de roteamento e apresentação de informações sobre redes e *gateways*.
- janela para apresentação dos componentes de uma determinada rede. Apresenta menu para acessar as funções de informação sobre a rede ou *gateway* e *host* selecionado, tabela de roteamento, conexões TCP abertas e dados estatísticos.
- janela para entrada de dados complementares contendo campos a serem preenchidos pelo usuário como por exemplo a localização geográfica do *host*, o nome da rede, tipo de equipamento, etc.
- janela para apresentação de dados em forma de texto, como por exemplo a tabela de roteamento de um determinado *gateway*,
- janela de *help* descrevendo uma determinada janela,
- janela para dados multimídia.

4.5 Estruturas de Dados

Considerando a natureza dinâmica do ambiente de redes onde *hosts* são adicionados e removidos, novas sub-redes 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 capte 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 sub-redes 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 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 4.3 - 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 4.4 ilustra a lista de *gateways*, e sua relação com a lista de interfaces.

Figura 4.4 - Lista de gateways.

4.6 Diagrama de Módulos do Sistema

A fim de se estabelecer as relações entre as funções do sistema, incluindo as estruturas de dados (E.Ds), foram definidos os módulos principais e esquematizado um diagrama de módulos do sistema monitor gráfico de redes, incluindo os arquivos consultados. O esquema geral é apresentado na figura 4.5, seguido do detalhamento dos módulos.

Figura 4.5 - Diagrama de Módulos do Sistema Monitor.

4.6.1 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 sub-redes dessas redes.

Figura 4.6 - 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, é 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 sub-redes (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 sub-rede 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.

4.6.2 Módulo Gráfico

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

Figura 4.7 - 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 4.8 - 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 desenhar 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 desenhar a 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 desenhar 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 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

desenha a conexão entre G1 e a rede

busca posição adequada na tela, em relação a rede, para desenhar 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 desenhar a 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 desenhar 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

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 4.2 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.

4.6.3 Módulo de Funções

O módulo de funções abriga as funções de coleta de dados estatísticos, verificação de status de *hosts* e *gateways*, apresentação de tabelas de roteamento, conexões ativas e entrada de dados complementares, onde essas funções são selecionadas através de menus. Os dados são mostrados textualmente através de janelas.

Figura 4.9 - Diagrama de Módulo de Funções.

Os dados estatísticos, obtidos através do módulo **coleta de dados estatísticos**, referem-se aos objetos contadores no grupo *iface*, descritos no item 4.2, nos *gateways* que possuem o agente SNMP implementado, para se ter uma noção do tráfego nas suas interfaces.

O status dos *hosts* é verificado através do comando *ping*. O módulo **status de hosts e gateways** aplica o comando *ping* a cada *host* ou *gateway* selecionado.

A tabela de roteamento é obtida pelo módulo **busca da tabela de roteamento**, obtendo-se via SNMP, os objetos do grupo IP referentes à tabela de roteamento: *ipRouteDest*, *ipRouteMetric*, *ipRouteNextRoute*, *ipRouteType* e *ipRouteProto*.

As conexões ativas em um determinado *host* são obtidas pelo módulo **conexões ativas**, através dos objetos relacionados ao grupo UDP e às conexões do grupo TCP: *udpLocalAddress*, *udpLocalPort*, *tcpMaxConn*, *tcpConnState*, *tcpConnLocalAddress*, *tcpConnLocalPort*, *tcpConnRemAddress* e *tcpConnRemPort*.

O módulo **apresentação de informações** deve estar associado ao objeto cujas informações serão mostradas. Assim o objeto deve ser primeiro selecionado, e o nó da lista

que contém as informações a seu respeito passado para o módulo de apresentação, o qual as coloca em uma janela e as apresenta ao usuário.

As informações que não puderem ser obtidas pelo sistema, tais como a localização geográfica, tipo de equipamento, nome, etc, de alguns *hosts*, *gateways* ou redes, podem ser entrados manualmente pelo gerente através do módulo **entrada de informações**.

CAPÍTULO 5

Implementação do Sistema

5.1 Introdução

A fase de implementação do sistema monitor gráfico de redes envolve duas partes:

- a definição do ambiente de desenvolvimento, e
- a implementação propriamente dita do sistema, seguindo o projeto descrito no capítulo 4.

O capítulo de implementação aborda aspectos dessas duas fases, ressaltando os pontos mais relevantes.

5.2 Ambiente de Desenvolvimento

O ambiente de desenvolvimento envolve basicamente dois itens: equipamentos e softwares. Os equipamentos utilizados foram SUNs SparcStations e SLCs com sistema operacional SunOS 4.3, interligadas em rede. Essa sub-rede faz parte da rede do ICMSC, a qual interconecta, além de SUNs, microcomputadores PC com sistema operacional DOS e Unix. A rede do ICMSC está conectada à rede do campus através de uma SParcStation com interfaces ethernet e FDDI (o *backbone* do campus é uma rede FDDI). Em relação aos softwares, identificou-se dois pontos do sistema onde a utilização de pacotes tornaria a implementação mais rápida e eficiente: a interface gráfica e a comunicação SNMP.

A escolha da biblioteca gráfica e de softwares para a implementação da interface gráfica foi direcionada pelos equipamentos. As SUNs utilizam o sistema XWindow e apresentam ferramentas para desenvolvimento de interface gráfica com o usuário: o gerador de interfaces DevGuide [Su90a] e o Toolkit XView [He90], ambos seguindo as especificações do OPEN LOOK *Graphical User Interface*. O XWindow apresenta primitivas gráficas que podem ser utilizadas na representação das redes gerenciadas e seus componentes.

Existem pacotes de domínio público que implementam a comunicação SNMP, por exemplo a distribuição CMU SNMP, o kit de desenvolvimento MIT SNMP e o

pacote ISODE. Na implementação do sistema, optou-se pelo ISODE 8.0, pela familiaridade através do livro *The Simple Book* [Ro91]. Adicionalmente, esse pacote pode ser de utilidade na implementação de um futuro módulo CMIP, uma vez que traz suporte para o desenvolvimento de aplicações OSI.

O sistema monitor foi implementado utilizando a linguagem de programação C por ser suportada por ambos os pacotes.

5.2.1 O Sistema XWindow

O Sistema XWindow [Sc90a], [He90], ou simplesmente X, é um sistema de janelas desenvolvido pelo MIT (*Massachusetts Institute of Technology*) capaz de controlar um terminal de vídeo bit-mapeado (matriz de *pixels* endereçáveis). Através do X, a tela pode ser dividida em áreas retangulares de E/S, denominadas **janelas**, que se comportam como terminais virtuais. Nessas janelas podem ser executadas aplicações normais baseadas em textos, ou aplicações gráficas.

A principal característica que faz do X um padrão, é a utilização de um protocolo de rede como base, que o torna independente da plataforma, uma vez que o protocolo X pode ser implementado para diferentes arquiteturas de computadores e sistemas operacionais. Outra vantagem decorrente do fato do sistema ser desenvolvido em cima de uma rede, consiste na capacidade de se dividir tarefas, ou seja, enquanto os dados são processados em um computador mais potente, os resultados podem ser mostrados em um outro que tenha uma resolução gráfica mais apurada.

O Sistema XWindow segue o paradigma cliente-servidor. No caso, o servidor é um programa de aplicação que controla recursos (ex. terminal de vídeo, teclado e dispositivo apontador), tornando-os disponíveis para as aplicações do usuário. Ou seja, o servidor isola a aplicação, do código específico do dispositivo. O processo cliente é o consumidor desses recursos. O cliente utiliza o protocolo X para transportar os pedidos para desenhar gráficos, ou pedir informações e o servidor para transportar até o cliente, as entradas do usuário ou as respostas de pedidos.

O Conjunto de Ferramentas XView

O desenvolvimento de aplicações X implica na interação com o protocolo X, ou seja, a aplicação deve ser capaz de receber e enviar mensagens X. Essa interação é facilitada pela biblioteca XLib, a qual realiza a interface com o Protocolo X. Porém, a programação utilizando as funções da biblioteca não é trivial; existe uma lista enorme de funções que devem ser manipuladas individualmente aumentando o código da aplicação, conseqüentemente, o número de erros e a degradação da performance.

Os conjuntos de ferramentas (*toolkits*), visam facilitar a implementação modularizando as funções mais comuns relacionadas à interface do usuário: criação de

janelas, menus, *scrollbars*, etc. Dentro deste contexto, o XView (*XWindow-System-based Visual/Integrated Environment for Workstations*) é um conjunto de ferramentas de interface orientado a objetos desenvolvido pela Sun Microsystems, Inc.

Mais especificamente, o XView fornece um conjunto de janelas que incluem:

- *Canvases* onde os programas podem desenhar;
- *Text subwindows* com capacidade de edição embutida;
- *Panels* contendo botões, itens de escolha e *sliders*; e
- *TTY subwindows* que emulam terminais baseados em caracteres.

O Gerador de Interfaces DEVGUIDE

O *OpenWindows Developer's Guide* [Su90a], ou simplesmente Devguide, é uma ferramenta de desenvolvimento projetada para facilitar a tarefa do programador de interface. Opera no ambiente de aplicação do OpenWindows, seguindo o padrão OPEN LOOK User Interface [Su90a]. Através do Devguide é possível desenvolver o *layout* da interface gráfica utilizando janelas, menus, botões, e ícones sem se escrever uma única linha de código. As modificações são efetuadas facilmente, sobrando mais tempo para o projeto e os testes da interface.

Ao ser executado, o Devguide apresenta a janela abaixo com as informações e controles necessários para operá-lo e os ícones representando os elementos que serão adicionados à interface do usuário.

Figura 5.1 -Janela de controle do Devguide.

O programador pode editar uma interface criando uma janela base, escolhendo o ícone "Base" e adicionando os elementos (botões, mensagens, *sliders*, *canvases*, etc) necessários, escolhendo os ícones correspondentes. Uma vez estando o layout da interface de acordo com os requisitos, o Devguide permite que se realize um teste simulando a

operação da interface, porém, ainda sem conexão entre a aplicação e a interface. Após o teste, o programador pode voltar à edição para realizar alterações, ou salvar a interface em um arquivo .GIL para futuras modificações, ou geração do código fonte.

Para gerar o código fonte em C com as chamadas do XView, existe o Gxv, um software que acompanha o Devguide. Além do código, o Gxv gera um Makefile, arquivos *headers* e arquivos *stubs* para auxiliar o programador da aplicação a fazer o *link* entre o código da interface e o código da aplicação.

5.2.2 O pacote ISODE

O ISODE (*ISO Development Environment*) é um pacote de domínio público contendo 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].

Os módulos do ISODE, figura 5.2, incluem:

- serviço de transporte OSI (TP0 sobre TCP e X.25; TP4 para OSI SunLink);
- serviços OSI de sessão, apresentação e controle de associação;
- ferramentas para verificação de sintaxe e transferência da linguagem abstrata ASN.1, incluindo:
 - geração de *stubs* de operações remotas (*front end* para operações remotas),
 - gerador de estruturas (ASN.1 para C),
 - *element parser* (BER - *Basic Encoding Rules*);
- serviços OSI de transporte confiável e operações remotas;
- gerenciamento, acesso e transferência OSI de arquivos;
- *gateway* FTAM/FTP;
- serviços OSI de diretório;
- terminal virtual OSI.

Figura 5.2 - Estrutura dos módulos do ISODE.

Apesar do objetivo do ISODE ser o desenvolvimento de aplicações OSI, o pacote traz embutido o 4BSD/ISODE SNMP, um pacote que implementa a estrutura de gerenciamento para sistemas Berkeley Unix. Inclui a implementação completa de um agente SNMP e ferramentas para a implementação de protótipos de aplicações de gerenciamento de redes. A versão 8.0 do ISODE pode ser obtida através de *anonymous FTP*, por exemplo, no endereço *ftp.psi.com* (38.145.211.6) dentro do subdiretório */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 manipular PDUs e mensagens SNMP (codificar, decodificar, liberar espaço de memória, montar, etc). O 4BSD/ISODE SNMP define estruturas de dados, em C, representando os tipos ASN.1 definidos em [Ro90a] ([Ro91] apresenta exemplos dessas estruturas) e as definições de mensagens, PDUs e *traps* apresentadas no RFC 1157 [Ca90]. Nesse pacote existe um programa de aplicação, o *snmpi*, capaz de buscar objetos em MIBs remotas dado o endereço e a comunidade do agente SNMP. O código fonte desse programa faz uso das primitivas em C constituindo-se num modelo para a implementação da parte de comunicação SNMP do sistema monitor. As primitivas são utilizadas através da inclusão de bibliotecas.

A segunda opção é o uso da linguagem *Gawk SNMP-capable*. A linguagem *gawk* é a implementação GNU da linguagem *awk* [Ke84], com algumas extensões. O pacote ISODE não traz os fontes a partir do qual é gerado o *gawk*, mas os arquivos utilizados para capacitá-lo a obter informações sobre objetos definidos em MIBs, remotas ou não, através do protocolo SNMP. Esses arquivos encontram-se no subdiretório *snmp/gawk-2.11*, gerado pelo 4BSD/ISODE SNMP, juntamente com programas exemplos escritos em *gawk SNMP-capable*. Os fontes da versão 2.11 do *gawk* podem ser obtidos através de *anonymous FTP*, por exemplo, no endereço *gumby.dsd.trw.com* (129.193.72.50), dentro do subdiretório */pub/networking/isode/isode-7*. O nome do arquivo é *gawk-2.11.tar.Z*.

Os arquivos utilizados para torná-lo capaz de obter objetos definidos em MIBs são basicamente dois: o programa *snmp.c* (utiliza as primitivas e estruturas em C descritas anteriormente) responsável pelas operações SNMP do *gawk*, e o arquivo *GAWK-PATCHES*. Esse arquivo contém trechos de *makefile* (inclusive a inclusão do *snmp.c* e de bibliotecas) e de códigos fontes que devem ser adicionados ao *Makefile* a aos códigos fontes originais do *gawk*. Essas modificações são aplicadas através do comando *patch* (*patch -p < GAWK-PATCHES*). Após a aplicação do patch, gera-se o programa executável através do comando *make*.

A linguagem *gawk SNMP-capable* 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*. No item 5.4.2 são apresentados alguns programas implementados utilizando *gawk SNMP-capable*.

Testes foram realizados para verificar a capacidade de comunicação do protocolo SNMP presente no pacote. Através do *snmpi* e de programas em *gawk*, foi possível obter

objetos de agentes SNMP disponíveis nas redes em servidores Novell, RISC rodando AIX Netview/6000, estação HP e roteadores CISCO (AGS e IGS).

5.3 Implementação das Estruturas de Dados

A implementação das estruturas de dados seguiu a especificação do projeto (capítulo 4). Os campos *anthost*, *nroiface* e *mark* foram adicionados para facilitar a implementação de rotinas do sistema monitor.

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

/* Estrutura de dados que contem as informacoes sobre as subnets */
struct subnet {
    char snetname [30]; /* nome da sub-rede */
    char snetaddr [16]; /* endereco Internet da sub-rede */
    char snetmask [16]; /* netmask da sub-rede */
    char snetecnol [50]; /* tecnologia da sub-rede */
    int nro; /* nro. de hosts na sub-rede */
    int x; /* posicao (x,y) da rede na tela de apresentacao
    int y; do mapa das redes */
    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 sub-rede */
};

/* 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 (x,y) do host na tela de apresentacao
    int y; dos componentes de uma rede */
    struct host *nexthost; /* ponteiro para o proximo host na lista */
    struct host *anthost; /* 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 sub-rede 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 (x,y) 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 5.3 - Implementação em C das E.Ds.

5.4 Implementação dos Módulos

Neste item são mostrados como os recursos do ambiente de desenvolvimento foram utilizados na implementação dos módulos descritos no capítulo de projeto.

5.4.1 Módulo Gráfico

A interface com o usuário foi implementada utilizando-se o DevGuide e XView, incluindo toda a parte de criação de janelas, menus, botões e outros itens de interface. Através do Gxv gerou-se o Makefile, onde foram inseridos os programas referentes às funções de montagem das estruturas de dados, desenho das redes gerenciadas, desenho dos componentes da rede e das demais funções do sistema monitor descritas no item 4.3. Através do Devguide, reduziu-se o tempo gasto na implementação da interface, porém o código gerado ficou maior se comparado à utilização direta do XView.

Na parte dos desenhos do mapas das redes e de seus componentes, figuras 5.4 e 5.5, foram utilizadas primitivas gráficas do XWindow para desenhar retas, retângulos, preencher polígonos, etc. Foram implementadas funções para desenho de *hosts*, *gateways*, redes e conexões.

As janelas do sistema monitor gráfico de redes podem ser visualizadas nas figuras seguintes, as quais foram obtidas através de *dumps* de tela.

A janela de apresentação do mapa das redes, figura 5.4, mostra as redes encontradas pelo sistema dentro do domínio de gerenciamento, no caso dado pelo arquivo *conf* contendo os seguintes endereços: 143.107.224, 143.107.230, 143.107.231 e 143.107.232. Os *gateways* que se encontram sem nome, são aqueles para os quais o sistema não conseguiu obter dinamicamente os nomes, visto que não estavam definidos no arquivo *hosts*.

Figura 5.4 - Janela de apresentação do mapa das redes gerenciadas.

A janela de apresentação do mapa das redes contém o menu principal, figura 5.5, o qual é ativado através do botão direito do mouse.

Figura 5.5 - O menu principal.

A opção **Quit** finaliza o sistema, e a opção **Symbols** apresenta a janela, figura 5.6, contendo os símbolos gráficos utilizados na representação das redes e seus componentes.

Ao se "clique" o mouse sobre uma rede, a janela apresentando os componentes dessa rede aparece em tela. A figura 5.7 apresenta a janela que surge em tela ao se selecionar a rede 143.107.231.0.

Figura 5.6 - Janela de símbolos gráficos.

Figura 5.7 - Janela de apresentação dos componetes de uma rede.

Embora não dê para se perceber pela figura, o sistema faz o desenho dos componentes da rede, pintando de vermelho os *hosts* não ativos e de verde os *hosts* ativos, detectados através do comando *ping*.

Quando um *host* ou *gateway* é selecionado através do mouse na janela de apresentação dos componentes de uma rede, aparece em tela a janela de informações. No exemplo da figura 5.8, a janela se refere às informações do *host* ianomami.

The screenshot shows a window titled 'Information' with a small icon in the top-left corner. It contains the following fields:

- Name: ianomami
- Alias: _____
- Internet Address: 143.107.231.2
- Physical Address: 08:00:20:0e:01:88
- Equipment: Sun Sparc 2
- Localization: Sala 27-j LDCC
- SNMP Agent: supported

Figura 5.8 - Janela de apresentação das informações sobre o *host* ou *gateway*.

As funções relacionadas à apresentação das conexões ativas, tabela de roteamento e informações sobre a interface de rede, são acionadas através de opções contidas no menu da figura 5.9. O menu apresenta outras funções que foram incorporadas ao sistema, devido à disponibilidade dos códigos fontes juntamente com o pacote 4BSD/ISODE SNMP. Essas funções referem-se ao sub-menu acionado através da opção **System**, e à opção **Device Information**.

Figura 5.9 - Menu de acesso às funções.

A opção **Interface Information** (informações sobre interface) mostra informações sobre as interfaces de rede, onde os valores são dados em número de pacotes. Os campos apresentados referem-se a:

Name: nome da interface;

MTU: unidade de transferência máxima;

Net/Dest: endereço Internet da rede;

Address: endereço Internet da interface;

Inpkts: número total de pacotes recebidos pela interface;

Ierrs: número de pacotes recebidos com erros;

Opkts: número total de pacotes transmitidos pela interface;

Oerrs: número de pacotes não transmitidos devido a erros;

Drop: número de pacotes para transmissão descartados por outros motivos que não erros, por exemplo, liberar espaço de *buffer*.

Interface Information									
Agent 143.107.231.2									
Name	Mtu	Net/Dest	Address	Ipkts	Ierrs	Opkts	Oerrs	Drop	Queue
1e0	1500	143.107.231.0	143.107.231.2	9275256	168	12908044	0	65	0
1o0	1536	127.0.0.0	127.0.0.1	47102	0	47102	0	0	0

Figura 5.10 - Janela de apresentação de informações sobre a interface de rede.

A opção **Active Connections** (conexões ativas) apresenta informações sobre os estados das conexões na máquina onde se localiza o agente SNMP, incluindo o estado das filas de recepção e transmissão. Os dados apresentados são:

Proto: tipo de protocolo;

Recv-Q: número de octetos enfileirados para leitura no *socket*;

Send-Q: número de octetos enfileirados para transmissão no *socket*;

Local address: endereço local da conexão (endereço Internet e o número da porta);

Foreign address: endereço remoto da conexão (endereço Internet e o número da porta remota);

State: estado da conexão.

The screenshot shows a window titled "Active Connections" with a subtitle "Active connections - 143.107.231.2". It displays a table with the following data:

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	(state)
tcp	0	0	127.0.0.1.sMUX	127.0.0.1.1031	ESTABLISHED
tcp	0	0	127.0.0.1.1031	127.0.0.1.sMUX	ESTABLISHED
tcp	0	0	143.107.231.2.1027	*.*	LISTEN
tcp	0	0	143.107.231.2.1700	*.*	LISTEN

Figura 5.11 - Janela de apresentação das conexões ativas.

A opção **Protocols Information** (informações sobre protocolos), apresenta informações relacionadas a estatísticas sobre os protocolos TCP/IP.

Figura 5.12 - Janela de apresentação de informações sobre protocolos TCP/IP.

A opção **Routing Table** (tabela de roteamento), apresenta informações relacionadas à tabela de roteamento, onde cada entrada na tabela apresenta dados sobre uma rota. Os campos *Refcnt* e *Use* são preenchidos apenas por agentes que implementam a MIB Unix.

Destination: endereço destino para esta rota;

Gateway: endereço do próximo gateway para esta rota;

Tipo: tipo de roteamento;

Refcnt: número de *sockets* utilizando a entrada de roteamento;

Use: número de vezes que a entrada foi utilizada;

Interface: interface de saída para esta rota.

Routing Table					
Routing table - 143.107.231.65					
Destination	Gateway	Type	Refcnt	Use	Interface
default	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.224.0	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.225.0	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.225.61	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.225.128	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.225.192	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.227.0	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.227.64	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.227.130	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.227.192	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.228.0	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.228.64	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.228.128	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.228.192	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.228.198	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.230.128	143.107.231.66	Remote			NetWare NE1000
Ethernet Driver (#3)					
143.107.231.0	143.107.231.66	Remote			NetWare NE1000

Figura 5.13 - Janela de apresentação da tabela de roteamento.

As funções acionadas através das opções do sub-menu **System** só operam em *hosts* cujos agentes implementam a MIB Unix. As janelas de informações apresentadas pelas opções **mbufs**, **print**, **users** e **Device Information** são ilustradas a seguir.

A opção **mbuf**, mostra informações sobre a alocação de mbufs no sistema onde roda o agente SNMP.

Figura 5.14 - Janela de apresentação de informação sobre a alocação de mbufs.

A opção **print**, apresenta informações sobre a tabela de impressão, onde os dados apresentados referem-se a:

Name: nome da fila na tabela de impressão;

Status: indica o estado corrente da fila na tabela de impressão;

Q: enfileiramento abilitado

P: impressão abilitada

D: daemon presente

Jobs: número de *jobs* na fila, incluindo o *job* corrente, se houver;

Remarks: estado corrente da fila;

Rank: posição do *job* na fila de impressão;

Owner: dono (*owner*) do *job*;

Job: nome do *job* na fila de impressão;

Files: conteúdo do *job*;

Total: tamanho do *job* em octetos.

Printing Table Information				
Agent 143.107.231.2				
Name	Status	Jobs	Remarks	
lp	Q+P+D-	0	lp is ready and printing	
lp1	Q+P+D+	1	sparc is ready and printing	
Rank	Owner	Job	Files	Total Size
active	castelo	054		133874 bytes

Figura 5.15 - Janela de apresentação de informações sobre a tabela de impressão.

A opção **system users** (usuários de sistema), apresenta informações sobre os usuários administrativos do sistema. Os campos referem-se a:

User

Name: nome do usuário;

ID: ID do usuário no sistema;

Group: Group ID do usuário;

Full name: nome completo do usuário;

Home: diretório *home* do usuário;

Shell: shell do usuário.

Groups:

Name: nome do grupo;

ID: ID do grupo no sistema;

Members: membros do grupo.

System User Information					
Users:					
Name	ID	Group	Full Name	Home	Shell
+	0	+			
audit	9	audit		/etc/security/audit	/bin/csh
bin	3	bin		/bin	
daemon	1	daemon		/	
ingres	7	7		/usr/ingres	/bin/csh
news	6	news		/var/spool/news	/bin/csh
nobody	65534	nogroup		/	
root	0	daemon	Operator	/root	/bin/csh
sunddiag	0	daemon	System Diagnostic	/usr/diag/sunddiag	/bin/csh
/usr/diag/sunddiag/sunddiag					
sync	1	daemon		/	
sys	2	kmem		/	/bin/sync
sysdiag	0	daemon	Old System Diagnostic	/usr/diag/sysdiag	/bin/csh
/usr/diag/sysdiag/sysdiag					
uucp	4	uucp		/var/spool/uucppublic	
Groups:					
Name	ID	Members			
+	0				
audit	9				
bin	3				
daemon	1				
kmem	2				
news	6				
nogroup	65534				
operator	5				
other	20				
staff	10				
tty	4				
uucp	8				
wheel	0				

Figura 5.16 - Janela de apresentação de informações sobre os systems users.

A opção **Device Information** (informações do dispositivo) apresenta informações sobre o dispositivo que implementa o agente SNMP. Essas informações referem-se aos objetos do grupo *system* da MIB I ou II:

- descrição do dispositivo;
- serviços oferecidos pelo dispositivo;
- nome da pessoa para contato; localização física do dispositivo;
- a quanto tempo o agente SNMP está no ar.

Device Information	
Agent 143.107.231.2	
running:	4BSD/ISODE SNMP
	(1.3.6.1.4.1.4.1.2.5)
services:	applications, end-to-end
location:	Sala 27-j LDCC
contact:	Edson Moreira <edsmorei@icmsc.usp.br>
uptime:	8 days, 2 hours, 10 minutes, 20.73 seconds

Figura 5.17 - Janela de apresentação dos objetos do grupo *system*.

5.4.2 Módulo de Montagem das Estruturas de Dados

A implementação do módulo de montagem das estruturas de dados seguiu o projeto, sendo desenvolvido um conjunto de funções para realizar os passos especificados na descrição de cada sub-módulo. A ênfase será em apresentar as funções escritas em *gawk SNMP-capable* para ilustrar como os objetos de MIBs remotas foram acessados através do protocolo SNMP. Os programas em *gawk* são ativados pelas funções e os resultados direcionados para um arquivo, o qual é processado pela função que chamou o programa. Em geral a linha de comando tem o seguinte formato: *gawk -f <nome do arquivo> <endereço Internet da máquina>*.

Determina-se que uma máquina possui o agente SNMP implementado se ela responder a uma requisição SNMP dentro de um período de tempo. Caso contrário assume-se que provavelmente não possui. O programa utilizado para esse fim é o TSNMP, o qual busca o objeto *ipForwarding*. O endereço Internet da máquina é passado como argumento (*argv[1]*).

```
BEGIN {
    # passando o endereço Internet do host através de argv
    AGENT = ARGV[1];
    RETRIES = 1;
    TIMEOUT = 1;
    if (ipForwarding == 1){ #e' gateway
        didone = 1;
        printf "1";
    }
    if (ipForwarding == 2){ #e' host
        didone = 1;
        printf "2";
    }
}
```

Figura 5.18 - Programa TSNMP.

O módulo de montagem inicial utiliza o programa TSNMP para descobrir quais as máquinas que possuem agente SNMP, e se possuírem, se são *hosts* ou *gateways*. A figura 5.19 mostra a relação entre o módulo de montagem inicial e o programa TSNMP.

Figura 5.19 - O módulo de montagem inicial juntamente com o programa TSNMP.

A obtenção de informações (endereço Internet, *netmask* e endereço físico) das interfaces dos *gateways* é feita através do programa TGATE, buscando-se os objetos *ipAdEntAddr*, *ipAdEntNetMask*, *ifPhysAddress* respectivamente. Como parâmetro passa-se o endereço internet 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 5.20 - Programa TGATE.

A figura 5.21 ilustra como o módulo de montagem da lista de *gateways* interage com o programa TGATE.

Figura 5.21 - O módulo de montagem da lista de *gateways* e o programa TGATE.

A obtenção de outros *hosts*, que não os especificados no arquivo *hosts*, é feito buscando-os nas tabelas ARP de *gateways* com agente SNMP através do seguinte programa em gawk.

```
BEGIN {
    # Obtem outros hosts através da tabela ARP do gateway que possui
    # o agente SNMP
    AGENT = ARGV[1];
    RETRIES = 1;
    TIMEOUT = 1;
    for (i in ipNetToMediaIndex) { # percorre a tabela ARP
        # Obtem o endereço Internet e físico do host
        printf "%s %s\n", ipNetToMediaNetAddress, ipNetToMediaPhysAddress
    } #for
}
```

Figura 5.22 - Programa THOSTS.

O programa TLOCAL busca a localização geográfica e o endereço físico de uma máquina, através dos objetos *sysLocation* e *ifPhysAddress*.

```
BEGIN {
    # gawk -f <arq> <end. agente> [s] [s]
    # argv[1] = endereço Internet do agente SNMP
    # argv[2] = s (busca a localização), n (não busca)
    # argv[3] = s (busca o endereço físico), n (não busca)

    AGENT = ARGV[1];
    RETRIES = 1;
```



```

TIMEOUT = 1;

if (ARGV[2] == "s"){
  # Obtem a localizacao do agente SNMP
  printf "%s ", sysLocation;
}

if (ARGV[3] == "s"){
  # Obtem o endereco fisico do host onde esta o agente
  for (i in ifIndex) {
    for (j in ipAdEntAddr) {
      if (ipAdEntIfIndex == ifIndex) {
        if(ipAdEntAddr == AGENT){
          printf "%s ",ifPhysAddress;
          break;
        }
      }
    }
  }
}
}

```

Figura 5.23 - Programa TLOCAL.

O módulo busca de novos *hosts* interage com os quatro programas como mostrado na figura 5.24, cuja sequência de execução ocorre da esquerda para a direita.

Figura 5.24 - O módulo busca de novos *hosts* e os programas em gawk SNMP-capable.

Outros *gateways* são obtidos buscando-se o *next hop* na tabela de roteamento dos *gateways* já obtidos, através do programa TOGATE.

```
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 ipRouteIndex) { # 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 5.25 - Programa TOGATE.

Na implementação do módulo busca de novos *gateways*, os programas TOGATE, TSNMP, TLOCAL E TGATE são utilizados como ilustra a figura 5.26.

Figura 5.26 - O módulo busca de novos gateways com os programas gawk SNMP-capable.

5.4.3 Módulo de Funções

Os módulos de funções relacionados à captura da tabela de roteamento, conexões ativas e informações sobre a interface de rede (incorpora alguns dados estatísticos), fazem uso, respectivamente, dos programas TROTA, TCONN e TIFACE (os códigos em gawk *SNMP-capable* encontram-se no apêndice A), para obter os objetos necessários via SNMP. Os programas são executados pelas funções e os resultados direcionados para arquivos, os quais são mostrados através de janelas texto, figuras 5.13, 5.11 e 5.10. Esses programas são basicamente os programas exemplos que acompanham o pacote 4BSD/ISODE SNMP, com pequenas adaptações para serem utilizados pelo sistema monitor.

CAPÍTULO 6

Conclusões e Trabalhos Futuros

A importância crescente do gerenciamento no contexto de redes de computadores gerou a necessidade de se dominar essa tecnologia, e o passo inicial dado nessa direção foi o desenvolvimento de um sistema monitor gráfico de redes, baseado na estrutura Internet, capaz de realizar funções básicas de monitoração (levantamento de informações estatísticas, verificação de status de dispositivos, apresentação de mapas das redes gerenciadas, apresentação de tabelas de roteamento e conexões ativas). As informações são apresentadas graficamente, sempre que possível, facilitando a identificação e agilizando a análise das informações por parte do usuário. Esse tipo de apresentação é bem mais amigável que as apresentações textuais.

O sistema monitor gráfico de redes foi desenvolvido com o intuito de adquirir tecnologia e conhecimentos, e expandi-lo futuramente, transformando-o em um sistema gerenciador de redes com recursos de multimídia, o MultView. Por essa razão, algumas das partes do sistema monitor foram projetadas para acomodar a inclusão de novas funções. Estas partes referem-se à definição das estruturas de dados e à interface gráfica.

O trabalho de mestrado dividiu-se em duas partes. Primeiro fez-se um estudo teórico envolvendo conceitos sobre redes de computadores e sobre a estrutura de gerenciamento Internet e a arquitetura OSI de gerenciamento, juntamente com seus protocolos de gerenciamento SNMP e CMIP. A segunda parte constituiu-se no projeto e implementação do sistema monitor gráfico de redes. A escolha do protocolo SNMP foi direcionada basicamente por três motivos: as redes existentes no campus da USP São Carlos são Internet, a simplicidade do protocolo, e a existência de equipamentos com agentes SNMP implementados, formando um campo de testes para o sistema monitor.

Os pontos fortes do sistema monitor desenvolvido centram-se:

- na capacidade de descobrir dinamicamente as redes que constam no domínio de gerenciamento, utilizando o protocolo SNMP e arquivos do Unix (*hosts* e *networks*),
- apresentação gráfica das redes e suas interconexões através de *gateways*,
- apresentação gráfica dos componentes de uma rede, e
- especificação de estruturas de dados flexíveis e adequadas, capazes de armazenar além das informações de configuração pertinentes às redes, *gateways* e *hosts*, informações

gráficas referentes ao posicionamento do objeto em tela. Essas informações gráficas constituem um suporte para a apresentação de informações de gerenciamento, utilizando recursos de multimídia, permitindo que se identifique um objeto pela posição na tela, sobre o qual se deseja por exemplo, mostrar uma imagem digitalizada ou abrir conexões para a passagem de sons ou imagens.

A interface com o usuário recebeu atenção especial, sendo projetada através de janelas seguindo o padrão XWindow. No projeto, teve-se o cuidado de implementar os ícones, botões e menus das funções a serem desenvolvidas para o sistema de gerenciamento, além dos elementos de interface necessários para o sistema monitor. A entrada de eventos que ativam outras funções do sistema pode ser realizada, através do *mouse*, diretamente sobre o desenho que representa um *host*, rede ou *gateway* sobre o qual se deseja obter informações, e não apenas através de menus ou botões, dinamizando a interação com o usuário.

Existem alguns pontos onde podem ser feitas alterações que melhoram a performance e a funcionalidade do sistema monitor.

- A busca de objetos via SNMP para a montagem da lista geral de hosts, lista de redes e de *gateways*, a partir das quais é montado o mapa das redes, demanda um tempo considerável se muitas máquinas tiverem que ser consultadas. Isso resulta do fato de que a maioria das máquinas não possuem agente SNMP implementado, consumindo todo o tempo do *timeout* para a resposta (no caso um segundo). Essa situação poderia ser remediada se inicialmente apenas algumas máquinas forem pesquisadas, e a partir destas, montadas as listas de redes e *gateways*, mostrando o mapa das redes referente às redes obtidas. As buscas de novos *hosts* e *gateways* seriam executados em processos separados e a comunicação entre eles e o sistema monitor realizada através de mecanismos de IPC (*Interprocess communication* - por exemplo, *named pipes*, *sockets*, etc). Assim, os novos objetos de redes descobertos seriam inseridos nas estruturas e os desenhos alterados dinamicamente, às vistas do gerente (usuário do sistema). Esse mecanismo possibilitaria a atualização constante do mapa das redes, sem a necessidade de se inicializar todo o procedimento de busca sempre que se desejasse efetuar alguma atualização.
- A utilização das primitivas e estruturas em C, disponíveis no 4BSD/ISODE SNMP para comunicação SNMP, embora mais complexas, permitiria um controle maior das operações de *get-request* e *get-next-request*, ao contrário da utilização do *gawk SNMP-capable*. Um exemplo é a possibilidade de se implementar um mecanismo de *timeout* na própria função que faz as buscas nas MIBs. O *gawk SNMP-capable* permite que se determine o intervalo de tempo, mas o mínimo é um segundo, tempo razoavelmente longo.
- A definição do domínio de gerenciamento poderia ser feito estabelecendo-se o número máximo de *hops* na busca de informações remotas para a montagem da lista de *gateways*. Uma possibilidade interessante é a presença de filtros para domínio de gerenciamento na apresentação do mapa das redes, ou seja, o gerente pode filtrar o domínio, por exemplo, definindo um intervalo de endereços de forma que só apareçam

as redes que lhe interessam no mapa. No desenho das redes poderia constar o intervalo em questão.

- O controle das áreas da janela onde é feito o desenho do mapa das redes poderia ser melhorado se fosse adotado um algoritmo mais adequado e abrangente. Existem algoritmos para desenho dinâmico de grafos que podem ser aplicados, uma vez que os problemas são semelhantes. Atualmente o algoritmo que faz o controle das áreas é restrito, permitindo a sobreposição de linhas, se a quantidade de conexões nos *gateways* ultrapassar um determinado limite.
- A função de levantamento de dados estatísticos poderia ser complementada, permitindo por exemplo que se captasse o tráfego diretamente nas redes. Isso exigiria a implementação de mecanismos específicos utilizando recursos do próprio Unix (ex. *etherfind*), uma vez que a MIBII não apresenta nenhum objeto que armazene tal tipo de informação. Ou implementando a RMON (*Remote Network Monitoring Management Information Base*), a qual define objetos para gerenciamento de dispositivos de monitoração remotos [Wa91]. Os dados estatísticos poderiam ser expressos na forma de gráficos.
- A definição de uma estratégia de armazenamento e recuperação, em disco, das estruturas de dados facilitaria a atualização e recuperação das informações das redes, contidas nas listas de *gateways* e de redes.
- A funcionalidade do sistema monitor poderia ser ampliada através da inclusão de novas funções tais como um editor de requisições. Através do editor o usuário poderia definir o agente, a comunidade e o objeto, selecionando-os em menus que listam os agentes SNMP encontrados, as comunidades desses agentes e os grupos de objetos da MIB I e II (para cada grupo é definido um submenu contendo os objetos do grupo). À medida que as seleções iam sendo realizadas, a requisição seria montada textualmente. O tratamento de *traps* e a implementação da operação *set-request*, possibilitaria a extensão do sistema monitor transformando-o em um sistema de gerenciamento, segundo a estrutura Internet. Um editor gráfico de topologias de redes seria interessante para permitir que o usuário rearranje a disposição das redes e *gateways* no desenho dos mapas das redes e edite o mapa de forma a inserir redes ou *gateways* que não puderam ser obtidos dinamicamente pelo sistema.
- A migração do sistema monitor gráfico para a versão 2 do protocolo de gerenciamento SNMP (SNMPv2) poderia ser realizada uma vez que já está disponível através de *anonymous FTP*.

Apesar de não incorporar ainda todas as funções propostas como trabalhos futuros, o desenvolvimento do sistema monitor gráfico de redes demonstrou a viabilidade de se desenvolver sistemas de gerenciamento, utilizando ferramentas simples tais como pacotes de domínio público e recursos do sistema Unix, e a importância de se ter tais tipos de sistemas no gerenciamento de ambientes computacionais modernos. A parte teórica, o projeto e os códigos fontes abrem caminhos e indicam atalhos que servem de ponto de

partida para o desenvolvimento de outros sistemas, que venham dar a sua contribuição para a área de gerenciamento de redes.

Apêndice A

Exemplos de programas em gawk SNMP-capable

Este apêndice apresenta os programas exemplos do pacote 4BSD/ISODE SNMP, escritos em gawk *SNMP-capable*, que foram utilizados na implementação de funções do sistema monitor gráfico de redes.

Programa TCONN

Obtém informações sobre as conexões ativas em um dispositivo que implemente o agente SNMP. Caso o dispositivo implemente a Unix MIB com objetos do Unix, esta também é consultada.

```
function socket(t,a,p, s1,s2) {
    s1 = a == "0.0.0.0" ? "*" : a;
    if (p == 0)
        s2 = "*";
    else
        if ((p, t) in services)
            s2 = services[p, t];
        else
            s2 = p;
    return (s1 "." s2);
}

function tt_type(f) {
    if (f in types)
        return types[f];
    return (f ? f : "unknown");
}

BEGIN {
    AGENT = ARGV[1];
    types[1] = "CLOSED";
    types[2] = "LISTEN";
    types[3] = "SYN_SENT";
    types[4] = "SYN_RCVD";
    types[5] = "ESTABLISHED";
    types[6] = "FIN_WAIT_1";
    types[7] = "FIN_WAIT_2";
    types[8] = "CLOSE_WAIT";
    types[9] = "LAST_ACK";
    types[10] = "CLOSING";
    types[11] = "TIME_WAIT";

    services[7, "udp"] = "echo";
    services[9, "udp"] = "discard";
    services[11, "tcp"] = "systat";
    services[13, "tcp"] = "daytime";
    services[15, "tcp"] = "netstat";
    services[20, "tcp"] = "ftp-data";
    services[21, "tcp"] = "ftp";
    services[23, "tcp"] = "telnet";
    services[25, "tcp"] = "smtp";
    services[37, "tcp"] = "time";
    services[37, "udp"] = "time";
    services[42, "udp"] = "name";
    services[43, "tcp"] = "whois";
    services[53, "udp"] = "domain";
    services[53, "tcp"] = "domain";
}
```

```

services[101, "tcp"] = "hostnames";
services[102, "tcp"] = "tsap";
services[109, "tcp"] = "pop";
services[111, "udp"] = "sunrpc";
services[111, "tcp"] = "sunrpc";
services[69, "udp"] = "tftp";
services[77, "tcp"] = "rje";
services[79, "tcp"] = "finger";
services[87, "tcp"] = "link";
services[95, "tcp"] = "supdup";
services[105, "tcp"] = "csnet-ns";
services[117, "tcp"] = "uucp-path";
services[119, "tcp"] = "untp";
services[123, "tcp"] = "ntp";
services[161, "udp"] = "snmp";
services[162, "udp"] = "snmp-trap";
services[199, "tcp"] = "smux";
services[1524, "tcp"] = "ingreslock";
services[512, "tcp"] = "exec";
services[513, "tcp"] = "login";
services[514, "tcp"] = "shell";
services[515, "tcp"] = "printer";
services[530, "tcp"] = "courier";
services[512, "udp"] = "biff";
services[513, "udp"] = "who";
services[514, "udp"] = "syslog";
services[517, "udp"] = "talk";
services[520, "udp"] = "route";
services[550, "udp"] = "new-rwho";
services[560, "udp"] = "rmonitor";
services[561, "udp"] = "monitor";

printf "Active connections%s\n\n", aflag ? " (including servers)" : "";
printf "%-5s %-6s %-6s %-20s %-20s %s\n",
    "Proto",
    "Recv-Q",
    "Send-Q",
    "Local Address",
    "Foreign Address",
    "(state)";

hasunix = unixNetstat == 1;
didone = 0;
for (i in tcpConnState) {
    didone = 1;
    if (!aflag && tcpConnLocalAddress == "0.0.0.0")
        continue;
    printf "%-5s %-6s %-6s %-20s %-20s %s\n",
        "tcp",
        hasunix ? unixTcpConnRecvQ[i] : "",
        hasunix ? unixTcpConnSendQ[i] : "",
        socket("tcp", tcpConnLocalAddress, tcpConnLocalPort),
        tt_type(tcpConnState);
}

```

```

if (!done && DIAGNOSTIC)
    printf "tcpConnTable: %s\n", DIAGNOSTIC;

done = 0;
for (i in udpLocalAddress) {
    done = 1;
    if (!flag && udpLocalAddress == "0.0.0.0")
        continue;
    printf "%-5s %-6s %-6s %-20s %-20s %s\n",
        "udp",
        hasunix ? unixUdpRecvQ[i] : "",
        hasunix ? unixUdpSendQ[i] : "",
        socket("udp",udpLocalAddress,udpLocalPort),
        hasunix ?
socket("udp",unixUdpRemAddress[i],unixUdpRemPort[i]) : "",
        "";
}
if (!done && DIAGNOSTIC)
    printf "udpTable: %s\n", DIAGNOSTIC;
}

```

Programa TROTA

Obtém informações sobre a tabela de roteamento de um dispositivo que implemente o agente SNMP. Caso o dispositivo implemente a Unix MIB, esta também é consultada.

```
function rt_type(f) {
    if (f in types)
        return types[f];
    return (f ? f : "unknown");
}

function rt_flags(f) {
    s = "";
    if (bit_and(f, 1))
        s = (s "U");
    if (bit_and(f, 2))
        s = (s "G");
    if (bit_and(f, 4))
        s = (s "H");
    if (bit_and(f, 8))
        s = (s "R");
    if (bit_and(f, 16))
        s = (s "D");
    if (bit_and(f, 32))
        s = (s "M");
    if (bit_and(f, 64))
        s = (s "Do");
    if (bit_and(f, 128))
        s = (s "C");
    if (bit_and(f, 256))
        s = (s "X");
    return s;
}

function do_stats() {
    if (!hasunix) {
        printf "routing statistics not implemented\n";
        exit(1);
    }

    didone = 0;
    for (i in unixRouteBadRedirects) {
        didone = 1;
        printf "routing:\n";
        printf "%9s bad routing redirects\n", unixRouteBadRedirects;
        printf "%9s dynamic created routes\n", unixRouteCreatedByRedirects;
        printf "%9s new gateways due to redirects\n", unixRouteModifiedByRedirects;
        printf "%9s destinations found unreachable\n", unixRouteLookupFails;
        printf "%9s uses of a wildcard route\n", unixRouteWildcardUses;
    }
    if (!didone && DIAGNOSTIC)
```

```

    printf "routing statistics: %s\n\n", DIAGNOSTIC;
}

BEGIN {
    AGENT = ARGV[1];
    RETRIES = 1;
    TIMEOUT = 3;
    hasunix = unixNetstat == 1;
    if (sflag) {
        do_stats();
        exit(0);
    }
    types[1] = "Other";
    types[2] = "Invalid";
    types[3] = "Direct";
    types[4] = "Remote";

    #printf "\n          Routing tables\n\n";
    printf "          %-15s %-15s %-8s %-6s %-10s %s\n",
        "Destination",
        "Gateway",
        hasunix ? "Flags" : "Type",
        "Refcnt",
        "Use",
        "Interface\n";

    didone = 0;

    for (i in ipRouteDest) {

        didone = 1;
        printf "          %-15s %-15s %-8s %-6s %-10s %s (%d)\n",
            ipRouteDest == "0.0.0.0" ? "default" : ipRouteDest,
            ipRouteNextHop,
            hasunix ? rt_flags(unixIpRouteFlags[i]) \
                : rt_type(ipRouteType),
            hasunix ? unixIpRouteRefCnt[i] : "",
            hasunix ? unixIpRouteUses[i] : "",
            ifDescr[ipRouteIfIndex],
            ipRouteIfIndex;
    }

    if (!didone && DIAGNOSTIC)
        printf "ipRoutingTable: %s\n", DIAGNOSTIC;

    if (!oflag)
        exit(0);

    didone = 0;
    for (i in cInpRouteDest) {
        didone = 1;

        printf "NS + %-28s %-8s %-6s %-10s %s (%d)\n",
            cInpRouteDest == "0" ? "default" : cInpRouteDest,
            hasunix ? rt_flags(unixcInpRouteFlags[i]) \
                : rt_type(cInpRouteType),

```

```
        hasunix ? unixclnRouteRefCnt[i] : "",
        hasunix ? unixclnRouteUses[i] : "",
        ifDescr[clnRouteIfIndex],
        clnRouteIfIndex;
    printf "    %-15s NS + %s\n",
        "", clnRouteNextHop;
}
if (!done && DIAGNOSTIC)
    printf "clnRoutingTable: %s\n", DIAGNOSTIC;
}
```

Programa TIFACE

Obtém informações sobre a interface de rede de um dispositivo que implemente o agente SNMP. Caso o dispositivo implemente a MIB do Unix, esta também é consultada.

```
BEGIN {
    AGENT = ARGV[1];
    TIMEOUT = 4;
    RETRIES = 2;
    printf "\n%-4s %-4s %-14s %-15s %-7s %-5s %-7s %-5s %-4s %-5s\n",
        "Name",
        "Mtu",
        "Net/Dest",
        "Address",
        "Ipkts",
        "Ierrs",
        "Opkts",
        "Oerrs",
        "Drop",
        "Queue";

    didone = 0;
    for (i in ifIndex) {
        didone = 1;
        dest = "";
        addr = "";
        for (j in ipAdEntAddr) {
            if (ipAdEntIfIndex == ifIndex) {
                split(addr = ipAdEntAddr, a, ".");
                split(ipAdEntNetMask, b, ".");
                dest = bit_and(a[1],b[1]) "." \
                    bit_and(a[2],b[2]) "." \
                    bit_and(a[3],b[3]) "." \
                    bit_and(a[4],b[4]);
                break;
            }
        }
        printf (length(ifDescr) <= 4 ? "%-4s " : "%s\n  "),
            ifDescr;
        printf "%-4d %-14s %-15s %-7d %-5d %-7d %-5d %-4d %-5d\n",
            ifMtu,
            dest,
            addr,
            ifInUcastPkts + ifInNUcastPkts,
            ifInErrors,
            ifOutUcastPkts + ifOutNUcastPkts,
            ifOutErrors,
            ifOutDiscards,
            ifOutQLen;

        if (oflag)
            for (j in clnpAdEntAddr) {
                if (clnpAdEntIfIndex == ifIndex) {
                    printf "%-4s %-5s %-14s NS+ %s\n",
```

```
        "", "", "", cInpAdEntAddr;  
        break;  
    }  
}  
if (!done && DIAGNOSTIC)  
    printf "ifTable: %s\n", DIAGNOSTIC;
```


Glossário

Abstract Syntax Notation One (ASN.1): a linguagem OSI para descrever sintaxe abstrata.

Association Control Service element (ACSE): elemento do serviço de aplicação responsável pelo estabelecimento e liberação de conexões.

Address Resolution Protocol (ARP): o protocolo do conjunto de protocolos Internet utilizado para mapear endereços IP em endereços físicos.

Agente de gerenciamento de rede: implementação de um protocolo de gerenciamento de rede que troca informações de gerenciamento com uma estação de gerenciamento de rede.

Applications Programmer's Interface (API): um conjunto de convenções de chamadas definindo como um serviço é invocado através de um pacote de software.

Basic Encoding Rules (BER): a linguagem OSI para descrição da sintaxe de transferência.

Checksum: soma aritmética utilizada para se verificar a integridade de dados.

CMIP over TCP/IP (CMOT): mapeamento da arquitetura OSI de gerenciamento para gerenciar redes baseadas no conjunto de protocolos Internet.

Common Management Information Protocol (CMIP): o protocolo OSI para gerenciamento de redes.

Common Management Information Service (CMIS): serviço oferecido pelo CMIP.

Common Management Information Service Element (CMISE): elemento de serviço de aplicação responsável pela troca de informações de gerenciamento de redes.

Comunidade: relação administrativa entre entidades SNMP.

Conjunto de protocolos Internet: uma coleção de protocolos de comunicação originalmente desenvolvidos sob responsabilidade da DARPA. O conjunto de protocolos Internet, ou TCP/IP, é atualmente a solução *de facto* para redes abertas.

Endereço *broadcast*: endereço IP ou específico do meio referindo-se a todas as estações no meio.

Entidade SNMPv2: processo real que realiza operações de gerenciamento de rede gerando e/ou respondendo a mensagens do protocolo SNMPv2.

Exterior Gateway Protocol (EGP): protocolo de alcançabilidade utilizado por roteadores em uma internet de dois níveis.

File Transfer, Access and Management (FTAM): serviço de arquivo do OSI.

Identificador de instância: um meio de se identificar uma instância de um tipo de objeto particular.

Institute of Electrical and Electronics Engineers (IEEE): organização profissional que, como parte de seus serviços para a comunidade, realiza alguns trabalhos de pré-padronizações para o OSI.

International Organization for Standardization (ISO): também conhecido por ISO/IEC (International Organization for Standardization and International Electrotechnical Committee), é uma organização que produz muitos dos padrões mundiais. O OSI é somente uma das muitas áreas padronizadas pelo ISO/IEC.

International Telephone and Telegraph Consultative Committee (CCITT): organização compreendendo o Post, Telephone, and Telegraph (PTT).

Internet: uma grande coleção de redes conectadas, inicialmente nos Estados Unidos, rodando o conjunto de protocolos Internet. Às vezes referido como DARPA Internet, NSF/DARPA Internet, ou Federal Research Internet.

internet: fisicamente, uma coleção de redes de comutação de pacotes interconectadas por *gateways* juntamente com protocolos que permitem que funcionem logicamente como uma única rede virtual.

Internet Activities Board (IAB): organização técnica que controla o desenvolvimento do conjunto de protocolos Internet.

Internet Control Message Protocol (ICMP): protocolo que faz parte integral do protocolo Internet (IP), e que lida com mensagens de erro e controle.

Internet Protocol (IP): o protocolo de rede do conjunto de protocolos Internet que oferece o serviço de transporte não orientado a conexão.

Layer Management Entity (LME): interface específica para cada camada do modelo de referência ISO/OSI, e que concentra a funcionalidade da camada de sua responsabilidade.

Local Area Network (LAN): uma rede de computadores que fornece alta velocidade, transferência com baixa latência e é limitada geograficamente.

Logical Link Control (LLC): executa as funções de enlace de dados independentes do meio e do controle de acesso ao meio.

Management Information Base (MIB): uma coleção de objetos que podem se acessados via um protocolo de gerenciamento de rede.

Maximum Transmission Unit (MTU): a maior quantidade de dados do usuário (ex. o tamanho máximo de um datagrama IP) que pode ser enviado em um único *frame* sobre um meio particular.

Medium Access Control (MAC): define o método de acesso ao meio, comunicando-se com as camadas superiores através do LLC.

MultView: sistema de gerenciamento de redes com recursos de multimídia a ser desenvolvido no ICMSC a partir do sistema monitor gráfico de redes.

Network File System (NFS): protocolo desenvolvido pela SUN Microsystems que utiliza o IP para permitir que um conjunto de computadores interligados em rede acessem os sistemas de arquivos uns dos outros. A vantagem principal do NFS sobre protocolos de transferência de arquivos convencionais é que o NFS esconde as diferenças entre arquivos remotos e locais colocando-os no mesmo espaço de nomes.

Network Information Service (NIS): auxilia o administrador de sistemas na configuração de um sistema baseado em redes.

Nó gerenciado: dispositivo de rede que contém uma implementação de um agente de gerenciamento de rede.

Open System Interconnection (OSI): um esforço internacional para facilitar a comunicação entre computadores de diferentes fabricantes e tecnologias.

Open Software Foundation (OSF): um consórcio de vários fabricantes de hardwares e softwares.

OSF/Motif: um padrão de interface gráfica que contém um conjunto de ferramentas, uma linguagem de descrição de apresentação, um gerente de janelas (*window manager*), e diretrizes (*guidelines*) de estilos.

Ping: programa utilizado para testar conectividade, a nível de IP, de um endereço Internet para outro.

Protocol Data Unit (PDU): um objeto de dado trocado entre camadas contendo dados do usuário e informações de controle do protocolo.

Protocolo de gerenciamento de redes: o protocolo utilizado para transportar informações de gerenciamento.

Remote Operation Service Element (ROSE): elemento do serviço de aplicação responsável pelas interações de *request/replay*.

Remote Procedure Call (RPC): mecanismo utilizado para modelar comunicação entre processos. Através do RPC, é possível chamar procedimentos em máquinas remotas, sem que a aplicação perceba, ou seja, do ponto de vista da aplicação, a chamada é local.

Routing Information Protocol (RIP): protocolo usado pelos sistemas 4BSD Berkeley UNIX para trocar informações de roteamento entre um conjunto de computadores.

Simple Mail Transfer Protocol (SMTP): protocolo padrão Internet para transferência de mensagens de correio eletrônico, de uma máquina para outra. O SMTP especifica como dois sistemas de correio eletrônico interagem e o formato das mensagens de controle trocadas na transferência de mensagens (*mails*).

Simple Network Management Protocol (SNMP): protocolo de aplicação do conjunto de protocolos Internet que oferece serviço de gerenciamento de rede

Sintaxe abstrata: descrição de um tipo de dado que é independente de restrições e estruturas de máquinas.

Sintaxe de transferência: descrição de uma instância de um tipo de dado que é expressado como uma string de bits.

SNMP Multiplexing Protocol (SMUX): mecanismo local utilizado para a comunicação entre um agente SNMP e um processo de usuário arbitrário que deseja exportar módulos de MIB para o agente.

System Management Application Process (SMAP): processo de aplicação de gerenciamento dentro da arquitetura OSI de gerenciamento.

Structure of Management Information (SMI): as regras usadas para definir os objetos que podem ser acessados via um protocolo de gerenciamento de rede.

System Management Application-Entity (SMAE): em um sistema OSI, o processo responsável pela coordenação entre as LMEs e o protocolo de gerenciamento.

Transmission Control Protocol (TCP): o protocolo do conjunto de protocolos Internet que oferece serviço de transporte orientado a conexão.

User Datagram Protocol (UDP): protocolo do conjunto de protocolos Internet que oferece serviço de transporte não orientado a conexão

Referências Bibliográficas

- [Be90] BEN_ARTZ, A.; CHANDNA, A.; WARRIER, U. Network Management of TCP/IP Networks: Present and Future. *IEEE Networks*, v.4, n.4, p.35-43, 1990.
- [Be88] BENHAMOU, E. Integrating Bridges and Routers in a Large Internetwork. *IEEE Network*, v.2, n.1, p. 69-71, 1988.
- [Bi84] BIRREL, A. D.; NELSON, B. J. Implementing Remote Procedure Calls. *ACM Transactions on Computer Systems*, v. 2, n. 1, p.39-59 , 1984.
- [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.
- [Ca93a] CASE, J. et. al. Structure of management Information for version 2 of the Simple Network Management Protocol (SNMPv2). Network Working Group, *Request for Comments 1442*, (DDN Network Information Center, SRI International), abril de 1993.
- [Ca93b] CASE, J. et. al. Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2). Network Working Group, *Request for Comments 1445*, (DDN Network Information Center, SRI International), abril de 1993.
- [Ca93c] CASE, J. et. al. Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2). Network Working Group, *Request for Comments 1446* (DDN Network Information Center, SRI International), abril de 1993.
- [Ca93d] CASE, J. et. al. Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2). Network Working Group, *Request for Comments 1448*, (DDN Network Information Center, SRI International), abril de 1993.
- [Ca93e] CASE, J. et. al. Coexistence between version 1 and 2 of the Internet-standard Network Management Framework. Network Working Group, *Request for Comments 1452*, (DDN Network Information Center, SRI International), abril de 1993.
- [Cl89] User Documentation for the Packet Driver Collection. Version 1.09, US Clarkson University, 1989.
- [Co91a] COMER, D. E. *Internetworking with TCP/IP -Principles, Protocols, and Architecture*. 2.ed. Englewood Cliffs, Prentice-Hall, 1991. v.1.

- [Co91b] COMER, D. E.; STEVENS, D. L. *Internetworking with TCP/IP -Design, Implementation, and Internals*. Englewood Cliffs, Prentice-Hall, 1991. v.2.
- [Co91c] COX, T. A; PISCITELLO, D. M.; TESINK, K. SNMP Agent Support for SMDS. *IEEE Network*, v. 5, n. 5, p.33-40, 1991.
- [Co93] COMER, D. E.; STEVENS, D. L. *Internetworking with TCP/IP -Client-Server Programming and Applications*, BSD socket version. Englewood Cliffs, Prentice-Hall, 1993. v.3.
- [Cu90] CURRIE, W. S. Interconnecting Local Area Networks. *Computer Networks and ISDN Systems*, v.19, p. 308-311, 1990.
- [Da87] DAVIN, J. R. et al. A Simple Gateway Monitoring Protocol. *Request for Comments 1028*, (DDN Network Information Center, SRI International), 1987.
- [Da91] DAUBER, S. M. Finding Fault. *BYTE*, v.16, n.3, p.207-214, 1991.
- [Fi91] FISHER, S. Dueling Protocols. *BYTE*, v.16, n.3, p.183-190, 1991.
- [Gi86] GIOZZA, W. F. et. al. *Redes Locais de Computadores -Tecnologia e Aplicações*. São Paulo, McGraw-Hill, 1986.
- [He90] HELLER, D. *XView Programming Manual - An OPEN LOOK Toolkit for X11*. O'Reilly&Associate, 1990. vol.7. v.2.
- [IS87] ISO TC97/SC21/WG4 N2066. *The Structure of Management Information*. Working Draft, 1987.
- [IS88a] ISO 8824. *Information processing systems-Open Systems Interconnection, Specification of Abstract Syntax Notation One (ASN.1)*. 1988.
- [IS88b] ISO 8825. *Information processing systems-Open Systems Interconnection, Specification of Basic Encoding Rules for Abstract Notation One (ASN.1)*. 1988.
- [Ke84] KERNIGHAN, B. W.; PIKE, R. *The UNIX Programming Environment*. Englewood Cliffs, Prentice-Hall, 1984.
- [Kl88] KLERER, S. M. The OSI Management Architecture: an Overview. *IEEE Networks*, v.2, n. 2, p. 20-29, 1988.
- [Kn88] KNIGHTSON , K. G.; KNOWLES, T.; LARMOUTH, J. *Standards for Open Systems Interconnections*. McGraw-Hill, 1988.
- [Le89] LEFLER, S. J. et. al. *The Design and Implementation of the 4.3 BSD UNIX Operating System*. Addison-Wesley Publishing Company, 1989.

- [Mc90] MCCLOGHRIE, K.; ROSE, M. *Management Information Base for TCP/IP-based internets. Request for Comments 1156*, (DDN Network Information Center, SRI International), 1990.
- [Me76] METCALFE, R. M.; BOGGS, D. R.. Distributed Packet Switching for Local Computer Networks. *Communications of ACM*, v. 19, n. 7, p. 395-404, 1976.
- [Mo93] MOREIRA, E. S.; ODA, C. S.; CICILINI, R.; FORTES, R. P. M. Desenvolvimento de um Gerenciador com Recursos de Multimídia para Redes Heterogêneas. /Artigo aceito para apresentação à SUCEsu/Nacional, Brasília, 1993/.
- [Ne92] NETVIEW MANUAL. *AIX NetView/6000 - Administration Reference*. IBM Corporation, v.1, 1992.
- [Od91] ODA, C. S. Um ambiente Gráfico para Monitoração de Redes Locais. In: REUNIÃO ANUAL DA SBPC, 43., Rio de Janeiro, 1991. *Anais*. v. 43, n.7, p.121-22.
- [On89] ONIONS, J. ISODE: In Support of Migration. *Computer Networks and ISDN Systems*, v.17, p. 362-366, 1989.
- [OI92] OLIVEIRA, M. C. Introdução ao X-Window e XVIEW. (Notas de aula referentes à disciplina Computação Gráfica), 1991.
- [Re93] REISS, L.; CADIN, J. *Unix System Administration Guide*. Osborne McGraw-Hill, 1993.
- [Ro90a] ROSE, M.; MCCLOGHRIE, K. Structure and Identification of Management Information for TCP/IP-based internets. *Request for Comments 1065*, (DDN Network Information Center, SRI International), 1990.
- [Ro90b] ROSE, M. T. Management Information Base Network Management of TCP/IP-based internets: MIB-II. *Request for Comments 1158*, (DDN Network Information Center, SRI International), 1990.
- [Ro90c] ROSE, M. T. *The Open Book-A Practical Perspective on OSI*. Englewood Cliffs, New Jersey, Prentice-Hall, 1990.
- [Ro91] ROSE, M. T. *The Simple Book-An Introduction to Management of TCP/IP-based Internet*. Englewood Cliffs, Prentice-Hall, 1991.
- [Sc88] SCHWARTZ, M. *Telecommunication Networks-Protocols, Modeling and Analysis*. Addison-Wesley Publishing Company, 1988.
- [Sc90a] SCHEIFLER, R. W.; GETTYS, J. The X Window System. *Software-Practice and Experience*, vol 20(S2), S2/5-S2/34, 1990.
- [Sc90b] SCHNAIDT, P. Keep It Simple. *Lan Magazine*, p.82-92, julho 1990.

- [Sc91] SCHILDT, H. *C Completo e Total*. São Paulo, Makron Books do Brasil Editora Ltda., 1991.
- [Sc92] SCHEPERS, H. J. H.; RIKKERT, K. O. B. P.; HAVERMANS, G. M. J.; HAMER, D. K. Naming, Addressing, Routing and Relaying in the OSI Environmet. *Computer Networks and ISDN Systems*, v.23, 1992.
- [Se88] SEIFERT, W. M. Bridges and Routers. *IEEE Network Magazine*, v.2, n.1, p. 57-64, 1988.
- [So91] SOCOLOFSKY, T.; KALE, C. A TCP/IP Tutorial. *Request for Comments 1180*, (DDN Network Information Center, SRI International), 1991.
- [Sp93] SPECIALSKI, E. S. Arquitetura OSI de Gerenciamento. /Apresentado à XI SBRC, Campinas, 1993/.
- [St84 a] STALLINGS, W. *Local Networks -An Introduction*. New York, Macmillan Publishing Company, 1984.
- [St84 b] STALLINGS, W. *Local Networks*. *Computing Surveys*, v. 16, n.1, p.3-27, 1984.
- [St90] STEVENS, W. R. *UNIX Network Programming*. Englewood Cliffs, Prentice-Hall, 1990.
- [Su90a] SUN MANUAL. *OpenWindows Developer's Guide 1.1 -Installation Manual*. Sun Microsystems, 1990.
- [Su90b] SUN MANUAL. *Networking Programming Guide*. Sun Microsystems, 1990.
- [Su90c] SUN MANUAL. *SunOS Reference Manual-Maintenance commands*. Sun Microsystems, 1990. v. 3.
- [Su92a] SUNNET MANAGER. *SunNet Manager-Cooperative Management Environmet*. SunConnect, março 1992.
- [Su92b] SUNNET MANAGER MANUAL. *SunNet Manager 2.0-References*. SunConnect, junho 1992.
- [Ta89] TANENBAUM, A. S. *Computer Networks*. 2.ed. Englewood Cliffs, Prentice-Hall, 1989.
- [Ta92] TANENBAUM, A. S. *Modern Operating Systems*. Englewood Cliffs, Prentice-Hall, 1992.

[Wa89] WARRIER, V.; BESAW, L. The Common Management Information Services and Protocol over TCP/IP (CMOT). *Request for Comments 1095*, (DDN Network Information Center, SRI International), 1989.

[Wa91] WALDBUSSER, S. Remote Network Monitoring Management Information Base. *Request for Comments 1271*, Carnegie Mellon University (DDN Network Information Center, SRI International), novembro 1991.

[Wi87] WILBUR, S.; BACARISSE, B. Building Distributed Computer Systems with Remote Procedure Call. *Software Engineering Journal*, p.148-59, setembro 1987.