**Universidade Federal do Rio de Janeiro**

**COPPE - Coordenação dos Programas de Pós-Graduação em Engenharia**

**Programa de Engenharia Elétrica**

## Gerenciamento de Redes

## Conceitos Básicos sobre os Protocolos SNMP e CMIP

**Alexandre Sztajnberg**

**Disciplina: COE728 - Redes de Computadores**

**Prof.: Otto Muniz Bandeira Duarte**

**Abril de 1996**

---

# Sumário

---

# 1. A Gerência em Redes de Computadores

As redes foram concebidas, inicialmente, como um meio de compartilhar dispositivos periféricos mais caros como impressoras, modems de alta velocidade, painéis pc-fax e etc. Entretanto, à medida que as redes crescem e tornam-se integradas às organizações, o compartilhamento dos dispositivos toma aspecto secundário em comparação às outras vantagens oferecidas. As redes passaram a fazer parte do cotidiano dos usuários como uma ferramenta que oferece recursos e serviços que permitem a interação e o aumento de produtividade.

Considerando este quadro, torna-se cada vez mais necessário a Gerência do ambiente de redes de computadores para mater todo este ambiente funcionando de forma *suave*.

A gerência em redes de computadores torna-se tarefa complexa em boa parte por conseqüência do crescimento acelerado das mesmas e tanto em desempenho e suporte a um grande conjunto de serviços. Além disso os sistemas de telecomunicações, parte componente das redes também adicionam complexidade a estas redes e estarão cada vez mais presentes, mesmo em pequenas instalações.

Este conjunto de componentes (e os problemas associados) somente poderão ser gerenciados se uma estrutura bem definida for seguida. Admitindo-se que as ferramentas para gerência de redes não abrangem toda a gama de problemas de uma rede e que estas nem sempre são usadas nas organizações que possuem redes, se faz necessário que outros mecanismos de gerência sejam utilizados para suprir suas carências mais evidentes.

As informações que circulam em uma rede de computadores devem ser transportadas de modo confiável e rápido. Para que isso aconteça é importante que os dados sejam monitorados de maneira que os problemas que porventura possam existir sejam resolvidos na medida do possível. Uma rede sem mecanismos de gerência pode apresentar problemas como congestionamento do tráfego, recursos mal utilizados, recursos sobrecarregados, problemas com segurança e outros.

A gerência está associada ao **controle** de atividades e ao **monitoramento** do uso de recursos da rede. As tarefas básicas da gerência em redes, simplificadamente, são obter informações da rede, tratar estas informações, possibilitando um diagnóstico, e encaminhar as soluções dos problemas. Para cumprir estes objetivos, funções de gerência devem ser embutidas nos diversos componentes de uma rede, possibilitando descobrir, prever e reagir a problemas.

Para resolver os problemas associados a gerência em redes a ISO através do OSI/MN propôs três modelos:

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

## 1.1 O que gerenciar ?

Dependendo da ênfase atribuída aos investimentos realizados no ambiente de processamento de dados, as funções de gerência de rede podem ser centralizadas no processador central (não é este o modelo que vamos explorar neste trabalho) ou distribuídas em diversos ambientes locais. Dependendo da heterogeneidade dos ambientes - por exemplo, um mainframe IBM com rede SNA e redes locais com ambiente UNIX, Netware da Novell ou LAN

Manager da IBM/Microsoft -, a dificuldade para gerenciamento de rede é muito grande, implicando, assim, o uso de várias ferramentas inseridas em uma estrutura, de certa forma, complexa, com os limites de atuação definidos (se possível, padronizados) entre os componentes envolvidos.

Esta estrutura pode definir aspectos como: a estratégia empregada no atendimento/chamadas dos usuários, atuação do pessoal envolvido nas tarefas de gerenciamento de rede, supridores de serviços, inclusive externos etc. Portanto, o ponto-chave para as atividades de gerência de rede é a organização, e aspectos como o atendimento do usuário se caracterizam como primordial para o sucesso da estrutura. É desejável que o usuário dos serviços de rede tenha um único ponto de contato para reportar problemas e mudanças.

Os limites de atuação desta gerência devem levar em conta a amplitude desejada pelo modelo implantado na instalação que, além de operar a rede, deve envolver tarefas como:

- > > Controle de acesso à rede
- > > Disponibilidade e desempenho;
- **> >** Documentação de configuração
- **> >** Gerência de mudanças;
- **> >** Planejamento de capacidade
- **> >** Auxílio ao usuário;
- **> >** Gerência de problemas
- **> >** Controle de inventário etc.;

A ênfase relativa atribuída em cada uma dessas categorias por uma instalação depende do tamanho e complexidade da rede.

# 1.2 Como a rede é vista pelos usuários

De um ponto de vista técnico, observa-se que as redes de computadores estão em constante expansão, tanto fisicamente como em de complexidade. Entretanto, para o usuário final a rede é vista como algo muito simples, ou seja, apenas supridor de ferramentas que facilitam suas atividades cotidianas. Outro aspecto não menos relevante é o fato de que nem sempre as redes locais estão instaladas em ambientes que tem como produto final a informática (e nem deveria...).

Desta forma, para o usuário, os sistema de computação deve estar disponível o tempo todo para auxiliá-lo a atingir objetivos como vendas, qualidade, rapidez, eficiência etc., *não importando* quais os mecanismos envolvidos para tal. Mas qual o verdadeiro impacto de uma eventual parada no computador central ? E se a paralisação for apenas parcial ? Ou apenas uma linha ou estação de trabalho ? Perguntas como estas devem ser levadas em conta antes da elaboração de qualquer modelo de gerência de redes, pois a partir de respostas a questões como estas é que se pode elaborar uma estrutura mínima.

Qualquer que seja a estrutura implantada, para se obter resultado dentro de padrões aceitáveis de serviços e informações, para o usuário final, além de ferramentas, é fundamental o bom nível técnico do pessoal envolvido com as atividades administrativas e de gerência da rede.

# 1.3 Protocolos de gerenciamento

Embora existam alguns protocolos desenvolvidos para gerenciamento de redes, este trabalho será concentrado nos protocolos padronizados na Internet e pela ISO, respectivamente o SNMP e o CMIP. Com o aumento de aplicações rodando em ambientes distribuído estes protocolos têm-se firmado ainda mais como solução para gerenciamento de grandes redes.

No modelo tradicional, as redes são compostas por múltiplos componentes. Além das máquinas em que as aplicações estão efetivamente executando, roteadores, *bridges*, *gateways* e modems são componentes importantes. No tocante ao software, vários outros componentes estão envolvidos, especialmente em ambientes ***multifornecedores*** e, em alguns casos, seria extremamente confortável que componentes de software pudessem er gerenciados. A tarefa de gerenciamento deve, então, ser resolvida por uma combinação entre entidades (como será visto em detalhes nas próximas seções) chamados de gerentes e agentes. O código de um agente é constituído por uma função de

gerenciamento - contadores, rotinas de teste, temporizadores etc. - que permite o controle e gerenciamento do objeto gerenciado. Já a instrumentação de gerenciamento está tipicamente associada a uma estrutura particular de gerenciamento, que especifica as regras empregadas para definir a informação referente a um objeto referenciado, permitindo, assim, que este possa ser monitorado e gerenciado.

A **SMI (*Structure Management Information***), como é chamada esta instrumentação, é análoga à linguagem de programação usada para construir estruturas de dados e permitir operações que possam ser executadas sobre essas estruturas. A combinação de uma SMI com um protocolo particular é denominada ***framework***.

**> > Padrões do Modelo de Referência OSI da ISO**

A ISO especifica o **CMIP** (***Common Management Information Protocol***) e o **CMIS** (***Commom Management Information Services***) como protocolo e serviço de gerenciamento de rede do nível de aplicação do modelo OSI.

A utilização dos padrões da ISO para gerenciamento têm sido amplidade (além dos méritos técnicos) em boa parte pela OSF, que está comprometida, através do **OSF/DME (Open Software Foundation/Distributed Management Environment**), em suportar os padrões OSI de gerenciamento. A função do DME é fornecer facilidades que permitam integrar o gerenciamento de sistemas em ambientes heterogêneos, satisfazendo três requisitos básicos: interoperabilidade, consistência e flexibilidade.

**> > Padrões TCP**

A necessidade de mecanismos de gerenciamento nas redes baseaas em TCP/IP é atendida pelo **SNMP** ***(Simple Network Management Protocol***) em associação com o esquema de **MIB** (***Management Information Base***), que também é suportado pelo padrão OSF/DME. Uma das vantagens do SNMP é a simplicidade e facilidade de implementação, e com isso a grande maioria dos problemas de gerenciamento de rede podem ser contornados com TCP/IP.

# 1.4 Arquiteturas de gerenciamento

Vários produtos têm surgido com a finalidade de gerenciar a rede, quase que em sua totalidade baseados no padrão SNMP e CMIP. O sucesso do SNMP baseia-se no fato de ter sido ele o primeiro protocolo de gerenciamento não proprietário, público, fácil de ser implementado e que possibilita o gerenciamento efetivo de ambientes heterogêneos. Geralmente, estes produtos de gerenciamento de redes incorporam funções gráficas para o operador de centro de controle.

No gerenciamento SNMP, é adicionado um componente ao hardware (ou software) que estará sendo controlado que recebe o nome de **agente**. Este agente é encarregado de coletar os dados dos dispositivos e armazená-los em uma estrutura padrão (denominada MIB, como mencionado anteriormente e que será vista em detalhes na seção 2). Além desta base de dados, normalmente é desenvolvido um software aplicativo com a habilidade de sumarizar estas informações e exibi-las nas estações encarregadas das tarefas de monitorar a rede.

Basicamente, são definidas quatro tipos de MIBs: MIB I, MIB II, MIB experimental e MIB privada. As MIBs do tipo I e II fornecem informações gerais sobre o equipamento gerenciado, sem levar em conta as características específicas deste equipamento. A MIB II, em verdade, é uma evolução da MIB I, que introduziu novas informações além daquelas encontradas na MIB I.

Portanto, através das MIBs do tipo I e II é possível obter informações como tipo e status de interface (Ethernet, FDDI, Token-Ring), número de pacotes transmitidos, número de pacotes com erro, informações de protocolos de transmissão etc.

As MIBs experimentais são aquelas que estão em fase de testes, com a perspectiva de serem adicionadas ao padrão e que, em geral, fornecem características mais específicas sobre a tecnologia dos meios de transmissão e equipamentos empregados.

As MIBs privadas são específicas dos equipamentos gerenciados, possibilitando que detalhes peculiares a um determinado equipamento possam ser obtidos. É desta forma que é possível se obter informações sobre colisões,

configuração, swap de portas, e muitas outras, de um HUB. Também é possível fazer um teste, reinicialização ou desabilitar uma ou mais portas do HUB através de MIBs proprietárias.

Os pioneiros na implantação dos protocolos SNMP foram os fornecedores de gateways, bridges e roteadores. Normalmente, o fornecedor desenvolve o agente SNMP e posteriormente desenvolve uma interface para a estação gerente da rede. Em geral, estes produtos funcionam para vários sistemas operacionais, como VMS, SUN-OS, DOS, AIX e outros, e é muito comum que estes fornecedores incluam bibliotecas e utilitários que permitam a criação de aplicações de gerenciamento com características específicas para alguns componentes da rede.

As implementações básicas do SNMP permitem monitorar e isolar falhas, já as aplicações mais sofisticadas permitem gerenciar o desempenho e a configuração da rede. Estas aplicações, em geral, incorporam menus e alarmes para facilitar a interação com o profisional que está gerenciando a rede.

O esquema dos produtos desenvolvidos com o protocolo SNMP são um pouco diferentes dos produtos que utilizam o protocolo CMIP. Os fornecedores de produtos que utilizam o protocolo CMIP pressupõem que os fabricantes possuam algum tipo de gerenciamento em seus equipamentos, portanto estas informações podem ser disponibilizadas para um integrador via protocolo CMIP. O conceito de integrador foi definido em três níveis: o mais baixo, que contém os agentes e os elementos gerenciadores, o intermediário, que consiste em elementos do sistema de gerenciamento, e finalmente o nível mais alto, que consiste no integrador dos sistemas de gerenciamento. Produtos como o NetView da IBM, Accumaster da AT& T, Allink da Nynex e o SunNet Manager da Sun Microsiytems, dentre outros, são exemplos deste tipo de implementação.

A dificuldade maior para uma aplicação integradora é que os fornecedores não tem as mesmas variáveis de gerenciamento e tampouco as mesmas operações em seus servidores de objetos.

A escolha entre um ou outro protocolo de gerenciamento deve recair sobre o tipo de rede e dos produtos a ela agregados, sendo que podem ser mesclados os dois protocolos.

O SNMP e seu ***Internet Standard Network Management Framework*** são adequados a agentes simples e fáceis de implementar, enquanto o CMIP e o seu ***framework Network Management Forum Release 1.0*** são adequados para agentes com um ou mais servidores de objetos dentro da modalidade cliente-servidor orientado para objeto, dentre os quais incluem-se o RPC.

Nas próximas seções os conceitos relacionados com MIB, SNMP e CMIP serão abordados com maiores detalhes.

---

# 2. Estrutura e Identificação da Informação de Gerenciamento

Um dos componentes conceituais de importância nos sistemas de gerenciamento é a forma com que as informações sobre os elementos básicos que se quer monitorar/gerenciar/administrar são armazenados. estas informações precisam estar disponíveis de forma padronizada, de forma que quanquer aplicação de gerenciamento possa resgatá-las e torná-las úteis de alguma forma.

Objetos gerenciados são acessados via uma informação virtual armazenada, denominada **Base de Informação Gerencial** (***Management Information Base***) ou **MIB**. Objetos de uma MIB são especificados usando a **Notação Sintática Abstrata** (***Abstract Syntax Notation One - ASN.1***).

Cada tipo de objeto (denominado *Object Type*) tem um nome, uma sintaxe e uma codificação. O nome é representado unicamente como um **IDENTIFICADOR de OBJETO** (*Object Identifier*). Um IDENTIFICADOR de OBJETO é um nome administrativo determinado.

A sintaxe para um tipo de objeto define uma estrutura abstrata de dados correspondente para este tipo de objeto. Por exemplo, a estrutura de um dado tipo de objeto pode ser um Inteiro (INTEGER) ou uma String de Octetos (OCTET STRING) ou ainda qualquer construção ASN.1 a ser avaliada para uso na definição da sintaxe de um tipo de objeto.

Uma codificação de um tipo de objeto é simplesmente como instâncias daquele tipo de objeto e são representados

usando a sintaxe deste tipo de objeto. Implicitamente ligado a notação de uma sintaxe de objeto e codificação é como o objeto está representado quando estamos transmitindo em uma rede.

# 2.1 Nomes

Nomes são usados para identificar objetos gerenciáveis. Nomes são especificados na sua hierarquia natural. O conceito de ***Identificador de Objeto*** é usado para reproduzir esta noção. Um Identificador de Objeto pode ser usado com outras intenções além de nomear tipos de objetos gerenciáveis; por exemplo, cada padrão internacional tem um identificador de objetos designados a eles com a intenção de identificação. Desta forma, os identificadores de objetos representa um significado para identificar algum objeto, sem considerar a semântica associada com este objeto (isto é, um objeto de rede, um documento padrão, etc.).

Um ***Identificador de Objetos*** é uma seqüência de Inteiros o qual atravessa uma árvore global. Esta árvore consiste de uma raiz conectada a um número de nós rotulados lado a lado. Cada nó pode, deste modo, ter seus próprios filhos os quais são rotulados. Neste caso, nós podemos assumir este nó como uma sub-árvore. Este processo pode continuar de um nível arbitrário ao mais profundo.

Um *Identificador de Objetos* é entendido como um controle administrativo de significados designados para os nós que podem ser delegados quando se atravessa pela árvore. Um rótulo é um par de uma breve descrição textual e um inteiro.

O nó raiz não é rotulado, mas tem pelo menos três filhos diretamente abaixo dele: Um deles é administrado pela ***International Organization for Standardization***, cujo o label é iso(1); o outro é administrado pela ***International Telegraph and Telephone Consultative Committee*** (agora denominado ITU-T), rotulado ccitt(0); e o terceiro é administrado em conjunto pela ISO e CCITT, denominada joint-iso- ccitt(2).

Abaixo do nó iso(1), a ISO designou uma sub-árvore para ser usada por outras organizações (inter)nacionais, denominado org(3). Destes nós filhos, dois foram designados para o ***National Institutes of Standards and Technology*** dos EUA. Uma destas sub-árvores foi transferida pelo NIST para o departamento de defesa dos EUA, denominado dod(6).

O DoD não mostrou como ele gerencia sua própria sub-árvore de Identificadores de Objetos. Desta forma, assume-se que o DoD aloca um nó para a comunidade Internet, para ser administrada pela ***Internet Activities Board (IAB)*** como se segue:

**internet OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 }**

Com isso, a sub-árvore da Internet de Identificadores de Objetos começa com o prefixo: **1.3.6.1.**

Agora, vamos especificar a política abaixo o qual esta sub-árvore de Identificadores de Objetos é administrada. Inicialmente, quatro nós são apresentados:

- **directory** OBJECT IDENTIFIER ::= { internet 1 }
- **mgmt** OBJECT IDENTIFIER ::= { internet 2 }
- **experimental** OBJECT IDENTIFIER ::= { internet 3 }
- **private** OBJECT IDENTIFIER ::= { internet 4 }

**Directory -** A sub-árvore Directory(1) é reservada para o uso do diretório OSI na Internet.

**Mgmt -** A sub-árvore mgmt(2) é usada para identificar objetos que estão definidos em documentos aprovados pela IAB. A administração da sub-árvore mgmt(2) é delegado pela IAB para a *Internet Assigned Numbers Authority*. As RFCs que definem novas versões de modelos Internet de MIBs aprovadas é determinado um identificador de objetos pela *Internet Assigned Numbers Authority* para identificar os objetos definidos por aquela RFC.

Por exemplo, a RFC que define o modelo Internet inicial de MIB pode ser designado como um documento de gerenciamento de número 1. Esta RFC pode usar o identificador de objetos { mgmt 1 } ou 1.3.6.1.2.1 na definição do modelo Internet da MIB.

A geração de novas versões do modelo de Internet da MIB é um processo rigoroso, existindo regras que são usadas

quando uma nova versão é definida.

**Experimental -** A sub-árvore experimental(3) é usada para identificar objetos usados por experiências da Internet. A administração desta sub-árvore é delegada pela IAB para a *Internet Assigned Numbers Authority*.

Por exemplo, um experimento pode receber o número 17 e ter disponível o identificador de objetos { experimental 17 } ou 1.3.6.1.3.17 para uso.

**Private -** A sub-árvore private(4) é usada para identificar objetos definidos unilateralmente. A administração desta sub-árvore é delegada pela IAB para a Internet Assigned Numbers Authority. Inicialmente, esta sub-árvore tem, no mínimo, um filho:

**enterprises OBJECT IDENTIFIER ::= { private 1 }**

A sub-árvore enterprises(1) é usada, entre outras coisas, para permitir que prováveis facções de subsistemas de redes registrem modelos de seus produtos.

Recebendo uma sub-árvore, a empresa pode, por exemplo, definir novos objetos MIB na sua sub-árvore. Com isto, é fortemente recomendado que a empresa também registre seus subsistemas de rede abaixo desta sub- árvore, de modo a evitar uma ambigüidade no mecanismo de identificação para serem usados pelos protocolos de gerência. Por exemplo, se a empresa “Flintstones, Inc.” produz subsistemas de rede, então ela pode requisitar um nó abaixo da sub-árvore enterprises para a Internet Assigned Numbers Authority. Com isto, um nó pode ser numerado assim: 1.3.6.1.4.1.42

A empresa “Flintstones, Inc.” pode então registrar sua “Rota Fred” com o nome de: 1.3.6.1.4.1.42.1.1

A figura a seguir mostra como é feita a alocação para redes baseadas nos protocolos TCP/IP, dentro da hierarquia global de árvore.

**Figura 2.1 - Árvore de alocação para redes baseadas no protocolo TCP/IP**

Acompanhando a árvore global mostrada anteriormente, vamos mostrar como o identificador de objetos 1.3.6.1.2 é obtido pela junção de grupos de números com o seguinte significado:

- ÆO primeiro grupo define o nó administrador:
  - (1) para ISO
  - (2) para CCITT
  - (3) para uma junção da ISO com a CCITT.
- ÆO segundo grupo para o nó administrador ISO define (3) para o uso de outras organizações
- ÆO terceiro grupo define (6) para o uso do Departamento de Defesa dos EUA (DoD).
- ÆNo quarto grupo o DoD não indicou como ele gerencia seu grupo, assim a comunidade Internet assumiu o grupo (1) para ela própria.
- ÆO quinto grupo foi aprovado pela IAB para ser:
  - (1) para o uso do diretório OSI na Internet
  - (2) para identificação de objetos com proposta de gerenciamento
  - (3) para identificação de objetos com proposta experimental
  - (4) para identificação de objetos para uso privado

No exemplo { **system 1** } além de ser o nome do objeto, significa que o identificador de objetos é 1.3.6.1.2.1.1.1. ele é o primeiro objeto dentro do primeiro grupo (system) dentro da Base de Informação Gerencial (MIB).

# 2.2 Sintaxe

A Sintaxe é usada para definir a estrutura correspondente aos tipos de objetos, usando as construções da linguagem informal ASN.1. O tipo ObjectSyntax define as diferentes sintaxes que podem ser usadas na definição de um tipo de objeto.

## 2.2.1 Tipos Primitivos

Apenas os tipos primitivos da ASN.1 como INTEIRO, STRING de OCTETOS, IDENTIFICADOR de OBJETOS e NULO são permitidos. Estes são algumas vezes referidos como **tipos não agregados**.

### 2.2.2 Tipos Construídos

O tipo construído SEQUENCE da ASN.1 é permitido, podendo este ser usado para gerar listas ou tabelas. Para listas, a sintaxe é da seguinte forma:

**SEQUENCE { < type1> ,..., < typeN> }**,

onde cada < type> está relacionado com algum dos tipos primitivos da linguagem ASN.1 listadas anteriormente. Para tabelas, a sintaxe é da seguinte forma:

**SEQUENCE OF < entry>** ,

onde < entry> pode ser uma lista construída. Listas e tabelas são, algumas vezes, referidas **como tipos agregados**.

### 2.2.3 Tipos Definidos

Em geral, novos tipos de aplicação podem ser definidos, sendo assim podem estar relacionados dentro de um tipo primitivo, lista, tabela, ou alguma outra aplicação implicitamente definido pela ASN.1. A seguir, alguns tipos de aplicação serão definidos:

Æ ***NetworkAddress*** **(Endereço de Rede)**

Esta "escolha" representa um endereço entre as várias possibilidades de famílias de protocolos. Até este exato momento, apenas uma família de protocolos, a família Internet, está presente nesta "escolha".

Æ ***IpAddress*** **(Endereço de IP)**

Este tipo de aplicação representa um endereço Internet de 32 bits. Ele é representado como uma STRING de OCTETOS de tamanho 4, dentro da ordem de bytes da rede.

Æ ***Counter*** **(Contador)**

Este tipo de aplicação representa um inteiro não negativo o qual aumenta até atingir o valor máximo, quando então ele encerra e começa de novo a aumentar a partir do zero. O valor máximo para os contadores é de 2^32-1 (4294967295 em decimal).

Æ ***Gauge*** **(Medida)**

Este tipo de aplicação representa um inteiro não negativo, que pode aumentar ou decrementar, mas encerra quando atinge o valor máximo. Este valor é de 2^32-1 (4294967295 em decimal).

Æ ***Time Ticks*** **(Intervalos de Tempo)**

Este tipo de aplicação representa um inteiro não negativo que conta o tempo em centenas de segundos desde alguma época**.** Quando tipos de objetos estão definidos em uma MIB a qual usa este tipo de ASN.1, a descrição deste tipo de objeto identifica a época correspondente.

Æ ***Opaque*** **(Opaco)**

Este tipo de aplicação sustenta a capacidade de passar arbitrariamente a sintaxe ASN.1. Um valor é codificado usando as regras básicas da ASN.1 dentro de uma string de octetos. Deste modo, é codificada como uma string de octetos, tendo o efeito de "ocultar duplamente" o valor ASN.1 original.

Note que esta implementação necessita apenas de ser capaz de acessar e reconhecer dados obscuramente codificados. Não é necessário ser capaz de desvendar o dado e então interpretar o seu conteúdo.

# 2.3 Codificações

Uma vez que uma instância de um tipo de objeto tenha sido identificada, seu valor deve ser transmitido aplicando-se as regras básicas de codificação da linguagem ASN.1 na sintaxe deste tipo de objeto.

# 2.4 Objetos Gerenciáveis

Embora a proposta deste capítulo não seja definir objetos na MIB, ele especifica o formato a ser usado por outros capítulos que definem estes objetos.

Uma definição de tipo de objeto usando TCP/IP possui cinco campos:

## 2.4.1 Objeto

É um nome textual para o tipo de objeto denominado DESCRITOR de OBJETO (Object Descriptor) o qual acompanha o seu correspondente IDENTIFICADOR de OBJETO.

## 2.4.2 Sintaxe

É uma sintaxe abstrata para um tipo de objeto. Ela pode ser uma escolha entre: uma Sintaxe Simples (SimpleSintax) que pode ser um tipo Inteiro (Integer), uma String de Octetos (Octet String), um Identificador de Objeto (Object Identifier) ou Nulo (Null); pode ser também uma Sintaxe de Aplicação (ApplicationSintax) podendo esta ser um Endereço de Rede (NetworkAdddress), um Contador (Counter), uma Medida (Gauge), um Intervalo de Tempo (TimeTicks) ou Incompreensível (Opaque); e, além destes também pode ser um outro tipo de aplicação.

## 2.4.3 Definição

É uma descrição textual da semântica de um tipo de objeto. Implementações devem assegurar que as instâncias do objeto cumpram esta definição desde que esta MIB seja pretendida por uso em ambiente multi- vendedor. A definição é vital para que os objetos tenham significados consistentes através de todas as máquinas.

## 2.4.4 Acesso

Para leitura, leitura e escrita, escrita ou sem acesso.

## 2.4.5 Status

Obrigatório, opcional ou obsoleto.

Ex.:

```
OBJETO

sysDescr    { system 1 }

Sintaxe     STRING de OCTETOS

Definição   Este valor deve incluir todo o nome e identificação da
            versão do tipo de hardware do sistema, software de
            sistema operacional e software de rede. É obrigatório
            que contenha apenas caracteres ASCII imprimíveis.

Acesso      leitura

Status      obrigatório
```

Este exemplo mostra a definição de um objeto contido em uma MIB. Seu nome é **sysDescr** e faz parte do grupo do System.

Um objeto gerenciador não tem apenas que estar definido mas identificado também. Isto é feito usando o

Identificador de Objetos como um número de telefone, reservando um grupo de números para diferentes localizações. No caso do TCP/IP - baseado em gerenciamento de rede, o número alocado é 1.3.6.1.2 e a SMI usa isto como uma base para definir novos objetos.

Deste modo, é definido uma Estrutura de Informação de Gerenciamento *(SMI - Structure Management Information*) que especifica o modelo de informação a ser adotado. Este modelo deve incluir a definição da estrutura da informação de gerenciamento armazenada em bases de dados destinadas a esse fim, as operações que podem ser realizadas sobre a mesma e as notificações que podem ser emitidas em decorrência de alguma operação ou alteração destas informações. Assim sendo, pode-se garantir a interoperabilidade entre diferentes sistemas de gerenciamento de rede onde tais sistemas conseguem ter uma visão comum da informação de gerenciamento.

# 2.5 Base de Informação Gerencial (MIB)

Todo sistema complexo necessita armazenar as informações manipuladas em algum tipo de base de dados. A Base de Informação Gerencial (**MIB - Management Information Base**) é o nome conceitual para a informação de gerenciamento, incluindo os objetos gerenciados e seus atributos. Pode-se considerar as informações para a configuração do sistema como também pertencentes à MIB.

A SMI descreve o cenário no qual a Base de Informação Gerencial pode ser definida. A SMI, baseada na abordagem orientada a objetos, introduz os conceitos de hierarquia, herança, nomeação e registros usados na caracterização e identificação de objetos gerenciados. Além disso, ela define o conjunto de operações que pode ser realizado sobre os objetos gerenciados da MIB e o comportamento desses objetos mediante a execução destas operações.

Dentro deste contexto, a MIB é definida como um conjunto de objetos gerenciados dentro de um Sistema Aberto, na qual um objeto gerenciado é a visão abstrata de um recurso real dentro deste sistema.

## 2.5.1 A MIB da Internet (TCP/IP)

A MIB da Internet define os objetos que podem ser gerenciados por cada camada do protocolo TCP/IP. Estes objetos estão sob a guarda de um agente de gerenciamento e a comunicação entre este agente e um gerente, localizado na estação de gerenciamento é feita utilizando o protocolo SNMP.

A MIB e o protocolo SNMP utilizam uma restrição do padrão ASN.1 do OSI [SMI, ASN.1] para a definição dos objetos e dos Protocol Data Units (PDUs). Inicialmente, esta escolha foi feita para permitir uma compatibilidade com o protocolo CMIP do modelo OSI, mas este objetivo não existe mais.

Como todos os padrões da tecnologia TCP/IP, as definições usadas no gerenciamento SNMP foram publicadas na série RFC (Requests For Comments). As definições originais do protocolo SNMP, bem como dos objetos gerenciados foram publicados em 1989. Em 1990, foi feita uma revisão da MIB, que passou a se chamar de MIB-II. Em 1993, foi publicado um conjunto de padrões novos, chamado SNMPv2, com alterações ao protocolo e extensões às definições dos objetos.

A MIB divide os objetos em vários grupos. A tabel a seguir mostra a MIB-I e os grupos nela definidos.

| Grupo | Objetos para | # |
|---|---|---|
| system | informações básicas do sistema | 3 |
| interfaces | interfaces de rede | 22 |
| at | tradução de endereço | 3 |
| ip | software de protocolo IP | 33 |
| icmp | protocolo de estatíst. para contr.interno de msgs. | 26 |
| tcp | software de protocolo TCP | 17 |
| udp | software de protocolo UDP | 4 |
| egp | software de protocolo EGP | 6 |

**Tabela 2.1 - número de objetos nos grupos**

A tabela a seguir mostra a MIB-II e os grupos nela definidos.

| Grupo | Objetos para | # |
|---|---|---|

| | | |
|---|---|---|
| system | informações básicas do sistema | 7 |
| interfaces | interfaces de rede | 23 |
| at | tradução de endereço | 3 |
| ip | software de protocolo IP | 38 |
| icmp | protocolo de estat. para contr. interno de msgs. | 26 |
| tcp | software de protocolo TCP | 19 |
| udp | software de protocolo UDP | 7 |
| egp | software de protocolo EGP | 18 |
| transmiss | transmissão. Média-específica | 0 |
| snmp | aplicações snmp | 30 |

**Tabela 2.2 - número de objetos nos grupos**

A lista definida de objetos gerenciáveis foi derivada daqueles elementos considerados essenciais. Esta implementação de se pegar apenas objetos essenciais não é restrita, uma vez que a SMI proporciona mecanismos de extensão como uma nova versão de uma MIB e uma definição de um objeto privado ou que não seja padrão.

A seguir, são listados alguns exemplos de objetos de alguns grupos:

Æ **Grupo System**

- sysDescr - completa descrição do sistema (versão, hardware, sistema operacional)
- sysObjectID - objeto para identificação do vendedor
- sysUpTime - tempo desde a última reinicialização
- sysContact - nome da pessoa de contato
- - sysServices - serviços oferecidos pelo dispositivo

Æ **Grupo IP**

- ipForwarding - indica se esta entidade é um gateway IP
- ipInHdrErrors - número de datagramas recebidos descartados devido a erros em seu cabeçalho IP
- ipInAddrErrors - número de datagramas recebidos descartados devido a erros em seu endereço IP
- ipReasmOKs - número de datagramas IP remontados com sucesso.
- ipRouteMask - máscara de sub-rede para rota

Æ **Grupo TCP**

- tcpRtoAlgorithm - Algoritmo para determinar o timeout para retransmissão de um octeto desconhecido
- tcpMaxconn - limite do número de conexões TCP que a entidade pode sustentar
- tcpInSegs - Número de segmentos recebidos incluindo aqueles recebidos com erro
- tcpConnRemAddress - o endereço remoto IP para determinada conexão TCP
- tcpInErrs - número de segmentos descartados devido ao formato de erro
- tcpOutRsts - número de reinicializações geradas

Æ **Grupo UDP**

- udpInDatagrams - número de datagramas UDP entregues aos usuário UDP
- udpNoPorts - número de datagramas UDP recebidos para aqueles onde não existe aplicação para aquela porta de destino
- udpInErrors - número de datagramas UDP recebidos que não podem ser entregues por diversas razões, menos a falta de uma aplicação para a porta de destino
- udpOutDatagrams - número de datagrama UDP enviados por esta entidade

Æ **Grupo Interfaces**

- ifIndex - número da interface
- ifDescr - descrição da interface
- if Type - tipo da interface
- ifMtu - tamanho do maior datagrama IP
- ifAdminisStatus - status da interface
- ifLastChange - hora em que a interface inicializou o status corrente

Embora os grupos anteriormente citados não formem uma MIB completa, servem como exemplo para mostrar como os objetos são nela definidos.

Como ilustração, considere a tabela 2.1 mostrada anteriormente. O grupo Interfaces contém dois níveis de objetos: o número da interface representado pelo nó (ifNumber) e uma tabela contendo informações daquelas interfaces (ifTable). Cada entrada (ifEntry) nesta tabela contém os objetos para uma interface particular. Com isso, o tipo de interface (ifType) é definido na árvore MIB usando a seguinte notação ASN.1: 1.3.6.1.2.1.2.2.1.3.

Para finalizar, a MIB da Internet não inclui informações de gerenciamento para aplicações tais como: Acesso a Terminal Remoto (TELNET), Transferência de Arquivos (FTP - File Transfer Protocol) e Correio Eletrônico (Simple Mail Transfer Protocol).

Além disso, na Internet não são previstos mecanismos para definir ações e eventos associados a um objeto gerenciado. Um conjunto bem menor de operações é definido para a Internet, conforme a tabela a seguir:

| OPERAÇÕES- | FUNÇÕES |
|---|---|
| **Get-request** | Obter o valor de uma variável específica |
| **Get-next-request** | Obter o valor da próxima variável, sem conhecer o seu nome |
| **Get-response** | Responder a uma operação de Get-request ou Get-next-request |
| **Set-request** | Armazenar um valor em uma variável específica |
| **Trap** | Resposta vinculada à ocorrência de um evento |

**Tabela 2.4 - conjunto de operações para o gerenciamento na Internet**

Os eventos que geram a operação ***trap*** definida no SNMP são pré-definidos, tais como: queda e recuperação de enlace, falha de autenticação e perda de vizinho de gateway.

## 2.5.2 A MIB no modelo OSI

Existem três tipos de **hierarquias de gerenciamento** usadas pelos Sistemas de Gerenciamento OSI: Herança, Nomeação e Registro.

- Hierarquia de Herança

A hierarquia de herança, também denominada de hierarquia de classe, está relacionada às propriedades associadas aos objetos descritas através de seus atributos, comportamento, pacotes condicionais, operações e notificações. Dentro desta hierarquia, define-se, então, o conceito de classes de objeto hierarquizadas às quais pertencem objetos com propriedades similares. Existem, então, superclasses às quais estão subordinadas subclasses. Uma subclasse herda todas propriedades de sua superclasse, de maneira irrestrita, independentemente da necessidade ou não destas propriedades. A estas subclasses podem ser aglutinadas propriedades adicionais.

A superclasse do topo desta hierarquia é chamada de TOPO *(Top)*, da qual todas as outras classes são derivadas. Dentro desta organização, as classes mais gerais são definidas próximas ao TOPO.

A figura abaixo apresenta um exemplo de árvore de Classes de Objetos Gerenciados.

**Figura 2.2 - árvore de Classes de Objetos**

Opcionalmente, utiliza-se o conceito de herança múltipla, ou seja, a habilidade de uma subclasse ser derivada de mais de uma superclasse, permitindo a maior reutilização possível de definições de classes. Isto também permite melhorar a capacidade de um sistema de gerenciamento reconhecer características familiares entre classes de objeto. Este conceito de herança múltipla assegura ainda que, quando uma classe herda a mesma característica de múltiplas superclasses, então, aquela classe é definida como se aquela característica fosse herdada de uma única superclasse.

Dentro desta hierarquia, conforme foi mencionado anteriormente, as propriedades dos objetos de uma classe são descritas através dos seus atributos, comportamento, pacotes condicionais, operações e notificações.

Um atributo pode determinar ou refletir o comportamento de um objeto gerenciado. Os atributos de um objeto podem ser obrigatórios ou contidos em pacotes condicionais. Se um atributo é obrigatório, ele deve estar presente em todos as instâncias dos objetos gerenciados de uma dada classe. Todo atributo possui um valor ou conjunto de valores

(**set-valued**) do mesmo tipo. A Asserção de Valor de Atributo (AVA - Atribute Value Assertion) é uma declaração de que um atributo particular possui um valor. O valor de um atributo pode ser lido e/ou modificado através de operações sobre objetos gerenciados.

Todos os objetos gerenciados de uma classe devem possuir o mesmo comportamento, que define: a semântica dos atributos, operações e notificações; a resposta às operações de gerenciamento; as circunstâncias em que as notificações devem ser emitidas; as dependências entre valores de atributos particulares e os efeitos dos relacionamentos entre objetos gerenciados.

Um pacote condicional é uma coleção de atributos, notificações, operações e comportamentos opcionais, que está totalmente presente ou ausente num objeto gerenciado. Apenas uma instância de uma pacote condicional pode existir em um objeto gerenciado e a mesma somente pode ser acessada como parte deste objeto.

Deve-se ainda ressalvar o conceito de classes alomórficas. Uma subclasse é dita alomórfica de sua superclasse quando apresenta comportamento semelhante à sua superclasse. Para uma subclasse alomórfica, a faixa de valores de um atributo herdado deve ser a mesma ou um subconjunto da faixa de valores definida para o mesmo atributo da sua superclasse alomórfica.

- Hierarquia de Nomeação

A hierarquia de nomeação, também chamada de hierarquia de *containment*, descreve as relações entre instâncias de objetos com seus respectivos nomes.

Dentro desta hierarquia, define-se um relacionamento “estar contido em” aplicado aos objetos. Objetos de uma classe podem conter objetos da mesma classe e de classes diferentes. Um objeto gerenciado que contém outro objeto é dito superior; o objeto contido é dito subordinado. Um objeto gerenciado está contido dentro de um e somente um objeto gerenciado superior.

Este relacionamento de *containment* pode ser usado para modelar hierarquias de partes do mundo real (por exemplo, módulos, submódulos e componentes eletrônicos) ou hierarquias organizacionais (por exemplo, diretório, arquivos, registros e campos).

Isto implica que o objeto gerenciado existe somente se o seu objeto superior existir e que todo objeto gerenciado tem um nome que é derivado de um relacionamento de *containment.*

O nível mais alto desta hierarquia é chamado de RAIZ, que é um objeto nulo e sempre existe. Dentro de uma instância superior, todos os subordinados são unicamente identificados por um nome característico relativo **(RDN - *Relative Distinguished Name***). Um RDN é formado por um atributo, chamado de atributo característico (*distinguished attribute*) e mais algum valor. A combinação do atributo e do valor deve ser única para cada instância do objeto tendo o mesmo superior. Um nome completo de uma instância, chamado de “nome característico” (DN - *Distinguished Name*), consiste em uma seqüência de RDNs começando pela RAIZ e inclui o RDN da própria instância. Assim, todos os DNs são únicos e cada instância de objeto tem um único nome.

Para cada classe de objeto, uma ou mais regras devem ser definidas para identificar a classe superior e o atributo característico. Estas regras são chamadas de *name bindings*.

Uma vez que um *name bindings* foi definido para uma classe de objeto, este *name bindings* está disponível para uso em todas as classes derivadas daquela classe. Novos *name bindings* podem ser criados como resultado de especialização, entretanto, para todas as classes de objetos alomórficas, novos *name bindings* somente podem ser criados se eles forem também definidos para todas suas superclasses alomórficas.

- Hierarquia de Registro

A hierarquia de registro, por sua vez, é usada para identificar de maneira universal os objetos, independentemente das hierarquias de heranças e nomeação. Esta hierarquia é especificada segundo as regras estabelecidas pela notação ASN.1 para árvore de registros usada na atribuição de identificadores a objetos. Cada nó desta árvore está associado a uma autoridade de registro (por exemplo, ISO - *International Organization for Standardization* e CCITT - *Consultative Commitee for International Telegraph and Telephone*) que determina como são atribuídos os seus números. Desta maneira, cada objeto é identificado por uma seqüência de números, cada um correspondente a um nó.

# 2.6 Operações de Gerenciamento

As operações de gerenciamento executadas na fronteira (fronteira entre um recurso e o objeto gerenciado que o representa) do objeto gerenciado são primitivas. Para que se obtenha sucesso na realização de uma operação, o sistema de gerenciamento invocador deve ter os direitos de acesso necessários e as restrições de consistência, associadas à classe do objeto gerenciado e não devem ser violadas. O sistema de gerenciamento pode ser requisitado para executar uma operação em vários objetos gerenciados com sincronização atômica, isto é, ou esta operação é efetivamente realizada sobre todos os objetos, ou não é realizada.

A definição da classe de objetos deve especificar, para cada operação sobre o objeto gerenciado, o critério para suportar pedidos de operações de gerenciamento com sincronização atômica com outros objetos gerenciados.

A execução só está sujeita a restrições existentes na definição de classes relevantes.

Existem dois tipos básicos de operações de gerenciamento que podem ser realizados sobre os objetos gerenciados:

Æ Operações orientadas a atributos;

Æ Operações sobre objetos gerenciados como um todo.

## 2.6.1 Operações Orientadas a Atributos

O comportamento descrito a seguir é comum a todas as operações orientadas a atributos:

- Todos os atributos envolvidos como parte de uma única operação devem estar disponíveis para o objeto gerenciado;
- Todas as operações falham se e somente se o comportamento do objeto gerenciado é tal que a operação não é realizada em alguns dos atributos deste objeto;
- Quando uma operação de leitura ou modificação sobre uma lista de valores de atributos é solicitada através de um único pedido, a forma em que estas leituras ou modificações são sincronizadas entre os atributos depende da definição do comportamento da classe do objeto gerenciado e da lista de atributos especificados na operação;
- Quando uma sincronização atômica está em efeito, os estados intermediários resultantes da operação não são visíveis por outras operações de gerenciamento. A utilização de sincronização atômica pode levar a resultados inesperados, se os atributos da lista não estiverem sincronizados entre si no objeto.

As seguintes informações devem estar disponíveis para execução de operações orientadas a atributos:

  - identificador de atributo;
  - filtros;
  - listas ordenadas das classes de objetos alomórficas;

Após a execução da operação, os seguintes resultados estão disponíveis na fronteira do objeto gerenciado:

  - identificador de atributo e valores dos atributos que sofreram operações;
  - indicação de erro para os atributos que não puderam ser submetidos à operação.

Os seguintes tipos de erros devem ser identificados:

  - identificadores de atributos desconhecidos, isto é, não encapsulado dentro do objeto gerenciado;
  - classe de objeto especificada que não é parte do conjunto alomórfico do objeto gerenciado;
  - falha geral no processamento da operação.

A operação de gerenciamento sobre um atributo de objeto gerenciado pode provocar efeitos diretos e/ou indiretos.

Os efeitos diretos são definidos pela operação de gerenciamento Substituição do Valor do Atributo (*Replace Attribute Value*).

Os efeitos indiretos são resultados de relações do objeto gerenciado em questão. Como exemplos de efeitos indiretos

pode-se citar:

- modificação de um atributo dentro do mesmo objeto;
- mudança de comportamento do objeto gerenciado;
- alteração de um atributo de um objeto gerenciado relacionado;
- mudança no comportamento de um objeto gerenciado relacionado causado pela
- mudança de um ou mais atributos naquele objeto gerenciado.

As operações orientadas a atributos especificadas nas normas [ISO/IEC 10165-1] são as seguintes:

**a)** ***GET ATTRIBUTE VALUE (Obtenção de Valor do Atributo)***

Esta operação aplica-se a todos os tipos de atributos, exceto aqueles definidos como não acessíveis para leitura. Sua função é ler a lista de valores de atributos especificada, ou, se nenhuma lista é fornecida, ler todos os valores de atributos a retornar àqueles que puderem ser lidos.

Para que uma operação de GET ATTRIBUTE VALUE seja realizada, devem estar disponíveis informações relativas aos identificadores de atributos ou de grupos de atributos usadas para determinar como ou se a operação em questão deve ser executada. Se os valores de tais atributos não puderem ser lidos, há indicação de erro, identificando-se os casos de erro devido à restrição de acesso para leitura.

**b)** ***REPLACE ATTRIBUTE VALUE (Substituição do Valor do Atributo)***

Esta operação aplica-se somente a grupos de atributos ou a atributos que são acessíveis para escrita. Sua função é alterar os valores dos atributos especificados com os novos valores conhecidos.

Para que uma operação de REPLACE ATTRIBUTE VALUE seja efetuada, é necessário que sejam fornecidas informações referentes aos identificadores dos atributos a serem alterados e seus novos valores. Tais informações são usadas para determinar como e se tal operação deve ser executada. Há indicação de erro para os atributos cujos valores não puderem ser substituídos, identificando-se aqueles não substituídos por não serem acessíveis para escrita.

**c)** ***SET WITH DEFAULT VALUE (Substituição do Valor do Atributo pelo Valor Default)***

Esta operação aplica-se a todo tipo de atributo, exceto àqueles definidos como não acessíveis para escrita. Sua função e fazer com que o objeto gerenciado substitua o valor de alguns de seus atributos pelo seu valor default, definido como parte da especificação da classe de objeto em questão. Esta operação não restaura as condições iniciais do objeto quando de sua criação.

Para a realização de uma operação de SET WITH DEFAULT VALUE devem ser fornecidas informações referentes aos atributos e aos grupos de atributos usados para determinar como e se os seus valores devem ser substituídos por seus valores default. Há indicação de erro se esta operação não puder ser efetuada, identificando-se as situações em que os valores de atributos não são acessíveis para escrita em que não existem valores default definidos.

**d)** ***ADD MEMBER (Inclusão de Valores)***

Esta operação aplica-se a atributos cujos valores são conjuntos (sets) acessíveis para escrita. Para cada conjunto especificado de valores de atributo, esta operação substitui os valores de atributos existentes pelo conjunto união do conjunto existente com o conjunto especificado nesta operação.

Para que a operação ADD MEMBER seja realizada, devem estar disponíveis informações relativas aos identificadores dos atributos e aos seus valores a serem adicionados. Estas informações são utilizadas para determinar como e se esta operação deve ser executada. Há indicação de erro para aqueles atributos cujos novos valores não puderem ser adicionados, identificando-se os valores não adicionados por não serem acessíveis para escrita.

**e)** ***REMOVE MEMBER (Remoção de Valores)***

Esta operação aplica-se a atributos cujos valores são conjuntos (sets) acessíveis para escrita. Para cada conjunto especificado de valores de atributo, esta operação substitui o conjunto existente de valores de atributo pelo conjunto diferença entre o conjunto já existente e o conjunto especificado nesta operação.

Para determinar se e como esta operação deve ser executada, devem ser fornecidas informações referentes aos identificadores de atributos e seus valores a serem excluídos. Há indicação de erros para aqueles atributos dos quais não puderam ser excluídos valores, identificando-se aqueles valores não excluídos por não serem acessíveis para escrita.

## 2.6.2 Operações sobre Objetos Gerenciados como um todo

Estas operações aplicam-se a objetos gerenciados como um todo e seus efeitos geralmente não se limitam a modificar os valores dos seus atributos. São definidas as seguintes operações: CREATE, DELETE e ACTION.

Operações adicionais podem ser definidas por meio da operação ACTION. A semântica destas operações faz parte da definição da classe de objeto gerenciado.

Os objetos gerenciados são criados e removidos por meio de operações de gerenciamento ou como efeito colateral de uma outra operação.

**a) *CREATE (Criação de Objeto)***

Esta operação aplica-se a todos os objetos especificados como passíveis de criação pela definição de sua respectiva classe de objeto. Sua função é requisitar a criação e iniciação de um objeto gerenciado. É uma operação única, desde que aplicada a um objeto gerenciado que ainda não exista. A operação CREATE cria um objeto gerenciado de uma classe específica ou pertencente a uma subclasse alomórfica, dentro da hierarquia de nomeação.

Quando um objeto gerenciado é criado, são designados valores a todos seus atributos, encapsulados no objeto ou em algum de seus pacotes específicos. Estes valores são obtidos a partir de informações fornecidas na própria operação CREATE e da definição de sua classe de objeto.

Na operação CREATE podem ser especificados explícita ou implicitamente um ou mais pacotes condicionais por meio da especificação de um objeto gerenciado ou por default, como parte da especificação da classe de objeto. Esta operação falha se o sistema não puder fornecer um objeto gerenciado com pelo menos um dos pacotes condicionais requeridos.

Uma definição de objeto gerenciado deve permitir a um objeto ser criado sem a especificação do nome de suas instâncias e sem a especificação de sua localização na hierarquia de nomeação.

A localização de um objeto gerenciado na hierarquia de nomeação pode ser especificada explicitamente, através da especificação do nome do objeto gerenciado que vai conter a instância que está sendo criada, ou implicitamente, através do nome da própria instância sendo criada.

Quando é especificado somente o nome do objeto gerenciado, que vai conter a instância criada, o RDN do novo objeto gerenciado é atribuído pelo sistema gerenciado.

Para determinar como e se a operação de criação pode ser executada, devem ser fornecidas as seguintes informações:

- Identificador da classe de objeto;
- Atributo de pacotes;
- Identificadores dos atributos para os quais devem ser designados valores específicos como parte da iniciação da instância de objeto;
- Identificador da referência do objeto gerenciado, a partir do qual a informação de iniciação deste objeto está sendo obtida.

Como resultado da operação de criação, devem ser retornadas as informações referentes aos identificadores dos atributos para os quais tiverem sido atribuídos valores como parte da iniciação da instância do objeto.

Além disso, no caso de o objeto gerenciado não poder ser criado, há indicação de erro, identificando-se as situações específicas, tais como: identificadores de atributos desconhecidos, identificador inválido da referência do objeto gerenciado e especificação do *name binding* inválido.

**b) *DELETE (Remoção de Objeto)***

Esta operação aplica-se a todos os objetos gerenciados que podem ser removidos remotamente. Ela pode ser efetuada mesmo sobre objetos gerenciados que tiverem sido criados através de uma operação local.

A operação DELETE requisita que o objeto gerenciado remova a si mesmo. Esta operação remove o objeto gerenciado que representa um recurso e implica um efeito análogo sobre o próprio recurso.

Quando um objeto gerenciado recebe um pedido de remoção, ele verifica se outros objetos gerenciados estão contidos nesse objeto. Caso afirmativo, o comportamento do objeto gerenciado depende da especificação da classe do objeto gerenciado.

O objeto gerenciado pode remover todos os objetos gerenciados nele contidos para assegurar a integridade do nome, ou pode recusar-se a executar esta operação de remoção até que todos os objetos gerenciados nele contidos tenham sido removidos.

Similarmente, quando um objeto gerenciado a ser removido tem relacionamento com outros objetos gerenciados, as remoções destes objetos podem comprometer a integridade dos relacionamentos e/ou dos objetos gerenciados relacionados. Geralmente, objetos gerenciados e seus relacionamentos devem ser removidos de maneira a assegurar a integridade do sistema gerenciado a cada remoção efetuada. Se a remoção de um objeto gerenciado resultar na perda da integridade do relacionamento, então, o objeto gerenciado pode rejeitar o pedido de remoção ou iniciar operações que assegurem que esta integridade seja mantida.

A emissão ou não de uma notificação, como resultado da remoção de um objeto gerenciado, depende da definição do objeto gerenciado.

Para que seja possível determinar como e se a operação de DELETE deve ser executada, devem ser fornecidas informações referentes à lista ordenada das classes alomórficas e aos filtros associados.

No caso de os objetos gerenciados não poderem ser removidos, é retornada uma indicação de erro identificando-se aqueles erros decorrentes de identificadores de atributos desconhecidos.

Como resultado da operação de remoção de objetos gerenciados, pode ou não ser emitida uma notificação. Isto depende da definição do objeto gerenciado.

**c) *ACTION***

Essa operação pode ser executada para todas as classes de objetos gerenciados. A operação ACTION requer que o objeto gerenciado execute a ação especificada e indique o seu resultado. A ação e a informação opcional associada são parte da definição de classe do objeto gerenciado.

As seguintes informações devem estar disponíveis para ser possível determinar como e se a operação ACTION deve ser executada sobre uma instância de classe de objetos gerenciados:

- lista ordenada das classes alomórficas;
- especificação da ação a ser executada;
- identificador da classe de objeto para suportar comportamento alomórfico.

Após a execução da ação especificada, são obtidas informações sobre o seu resultado e eventuais erros. Neste caso, devem ser identificados erros relativos à ação desconhecida e à classe de objeto desconhecida.

# 2.7 Compiladores de MIBs

Além da descrição dos objetos gerenciados e suas relações, uma MIB contém informações detalhadas sobre cada objeto, como por exemplo, o tipo de acesso a um objeto, um valor *default* razoável para um objeto e um conjunto de valores que um objeto pode assumir. Estas informações possuem um valor inestimável para os fornecedores de softwares para agentes e gerentes pois elas permitem que vários fornecedores utilizem um mesmo conjunto de informações de gerenciamento, obedecendo um conjunto de características padrão. Além disso, uma MIB pode ser compilada por um compilador de MIBs, de forma que as informações presentes na MIB estejam disponíveis para aplicações como MIB *browsers* e *graphers*. Estas aplicações são consideradas aplicações genéricas. São aplicações simples que obtém toda a sua capacidade de gerenciamento através da análise de uma MIB, sem qualquer intervenção

humana.

Além de checar a sintaxe de uma MIB, um compilador de MIBs pode gerar automaticamente as estruturas de dados e o código necessários para que um agente implemente uma determinada MIB. Um compilador de MIBs também pode fazer com que as informações sobre os objetos gerenciados de MIBs proprietárias ou de novas MIBs que sejam padronizadas estejam disponíveis para uma aplicação de gerenciamento existente.

A entrada para um compilador de MIBs é uma coleção de módulos de MIBs escritos em um subconjunto de linguagem ASN.1. Estes módulos contêm definições de objetos gerenciados que correspondem às informações sobre os dispositivos da rede que podem ser manipulados através do protocolo SNMP. Os compiladores de MIBs podem gerar várias representações das definições dos objetos gerenciados contidos nas MIBs usadas como entrada. Estas representações podem ser processadas mais facilmente pelos agentes e aplicações de gerenciamento do que a representação ASN.1

Algumas destas representações são declarações de estruturas de dados em linguagens de programação de alto nível, como C, que podem ser compiladas e ligadas em uma aplicação de gerenciamento ou agente. Outras são arquivos de dados contendo representações das definições dos objetos gerenciados que podem ser lidas para a memória por uma aplicação de gerenciamento ou agente em tempo de execução. Em alguns casos, o compilador de MIBs gera um código de saída que auxilia na implementação das MIBs de entrada. Por exemplo, um compilador de MIBs pode gerar esqueletos de rotinas para a recuperação ou alteração do valor de um objeto gerenciado, ou rotinas para a geração de **Trap-PDUs** específicas.

A habilidade de reconhecer as descrições presentes em uma MIB mecanicamente é muito atraente, principalmente para os fabricantes de aplicações genéricas, pois estas podem cobrir uma grande variedade de agentes de MIBs. Com o grande número de MIBs padronizadas e MIBs proprietárias disponíveis atualmente, os compiladores de MIBs reduzem o esforço dos fornecedores para manterem suas aplicações atualizadas.

Assim, muitos esforços estão concentrados em facilitar a forma pela qual diversas MIBs possam ser compiladas em cada produto de gerenciamento. Tentativas estão sendo feitas para que mais informações possam ser reconhecidas dinamicamente possibilitando que uma aplicação gerencie eficientemente um dispositivo completamente desconhecido para o fornecedor do produto de gerenciamento e para o usuário. Embora os compiladores de MIB já tenham provado sua utilidade, o gerenciamento de redes inteligente não poderá ser alcançado pela simples utilização desta tecnologia.

Infelizmente, a informação mais importante da MIB, ou seja, o texto que descreve detalhadamente um objeto, não pode ser compreendido por um compilador de MIB (com a tecnologia disponível atualmente). Por exemplo, um compilador pode ler a descrição de um objeto da MIB-II e aprender que este objeto é um inteiro que pode assumir os valores um e dois, o valor deste objeto pode ser lido e alterado e a implementação deste objeto é obrigatória. Mas somente um ser humano pode compreender a partir da descrição em linguagem natural do objeto *ipForwarding*, que se o valor deste objeto for igual a um então o sistema descrito por este objeto está atuando como um gateway, senão o sistema é apenas um nó da rede. Além disso, existem informações conhecidas por um administrador de rede experiente que não são descritas em nenhuma MIB, como por exemplo, o fato de que em algumas circunstâncias, pode ser perigoso para um sistema atuar como um gateway. Para a construção de um sistema de gerenciamento de redes inteligente, as aplicações devem conter todo este conhecimento. Como este conhecimento não pode ser fornecido no formato da MIB, os fornecedores de aplicações de gerenciamento devem desenvolver outras formas de incluir estes conhecimentos em suas aplicações.

Sem esta inteligência, muitas aplicações genéricas ficam limitadas à coleta, formatação e exibição das informações de gerenciamento. Estas informações são apresentadas para o usuário, que aplica sua inteligência humana para analisá-las. Esta carga só poderá ser retirada das mãos do administrador de rede se as aplicações se tornarem mais inteligentes. Sem esta inteligência será muito difícil que uma aplicação possa coletar informações suficientes de uma MIB desconhecida para gerenciar eficientemente um dispositivo desconhecido.

# 2.8 Interface com o Usuário

A indústria de gerenciamento de redes reconhece que, embora os protocolos e MIBs de gerenciamento tenham progredido muito, as aplicações de gerenciamento ainda deixam muito a desejar. Alguns membros da comunidade de padronização acreditam que a solução é a inclusão de novos tipos de informação ao formato da MIB padrão. As

informações orientadas para a aplicação deveriam ser adicionadas às definições dos objetos da MIB. Além das informações existentes, como tipo e descrição, as informações orientadas para a aplicação consistem de labels para tabelas ou gráficos, informações para formatação, valores-limiares (*thresholds*), texto de ajuda (*help*), e outras. Estas informações seriam lidas pela estação de gerenciamento, que as utilizaria para produzir uma melhor interface com o usuário para os objetos de gerenciamento SNMP.

Como exemplo, o objeto *etherStatsCollisions* da MIB RMON (veja seção 5.1) seria da seguinte forma:

```
etherStatsCollisions APPLICATION-INFO
                - - suitable for column header
        SHORT-LABEL     “Collisions”
        LONG-LABEL      “Ethernet CSMA/CD Collisions”
        PRINT-FORMAT    “decimal”
        USEFUL-STAT     “per etherStatsPackets”
                - - i. e., also useful per second
        USEFUL-STAT     “per sysUpTime * 100”
                - - absolute value isn’t interesting
        THRESHOLD       “.03 per etherStatsPackets”
                - - polling is useful
        VOLATILITY volatile
        ICON-BIT-MAP
                “120A9847E48C92A001F28437B5900E12
                8A7712C890203D487565D6080C4E7478
                3B921C983A782C08314E78213C019C28
                3E774C182A001B2438A7565D6080437B”
        HELP-TEXT               “us-english”
“A collision is an event on an Ethernet network that is a part of
everyday operation, but when excessive, can signal that the network
is overloaded or is too long (especially when the average packet size
is small). To avoid excessive collisions, use Token Ring.”
```

## 2.8.1 A Utilização das Informações Orientadas para a Aplicação

As informações orientadas para a aplicação são lidas pela estação de gerenciamento. Algumas destas informações podem controlar como a aplicação vai exibir os dados coletados através do protocolo SNMP, enquanto outras informações podem ser exibidas para ajudar o usuário a entender o significado destes dados. No entanto, as aplicações que utilizam estas novas facilidades não podem ser consideradas “inteligentes”, pois não são capazes de fazer recomendações para o usuário com base nos dados recebidos através do protocolo SNMP. Esta inteligência, tão desejada pelos gerentes de rede, só poderá ser fornecida adequadamente pelas aplicações desenvolvidas para uma MIB particular.

Como as MIBs proprietárias multiplicam o número de MIBs padrão por um fator de dez, é muito difícil para os fabricantes de aplicações de gerenciamento de rede suportar todas as MIBs proprietárias. Para estas MIBs, as informações adicionadas orientadas para a aplicação são muito importantes.

## 2.8.2 Como Divulgar as Informações da Interface com o Usuário

Foi sugerido que as informações orientadas para a aplicação fossem adicionadas ao formato padrão das MIBs, pela extensão da macro OBJECT-TYPE. Por várias razões, este não é o lugar ideal para definir estas informações. Uma MIB é um contrato entre os projetistas de agentes e os projetistas de aplicações de gerenciamento, realizado por uma entidade de padronização ou por um fabricante. Uma MIB descreve os objetos de gerenciamento para garantir que a implementação dos agentes e gerentes utilizem as mesmas definições. Os autores de MIBs em geral e as entidades de padronização em particular não podem assumir a responsabilidade adicional de projetar a interface para os usuários. Os grupos que trabalham na definição de novas MIBs possuem muitos outros detalhes para se preocupar.

Além disso, os grupos de padronização são internacionais. Claramente, não seria apropriado que um padrão definisse uma interface em inglês. Por outro lado, seria igualmente inadequado que um destes grupos de trabalho gaste um tempo enorme traduzido as informações de uma interface em 20 línguas diferentes, garantindo que todos os labels possam ser exibidos em uma tela de 80 colunas!

## 2.8.3 O Caminho Certo

A forma correta de adicionar estas informações nas aplicações seria a criação de uma nova macro, ligada ao objeto da MIB ao qual se refere. Arquivos de macros APPLICATION-INFO poderiam ser fornecidos pelos vendedores de

agentes para definir partes da interface para o usuário das estações de gerenciamento. Esta estratégia seria mais apropriada para MIBs proprietárias, que, de outra forma não poderia esperar suporte à aplicação de todas as estações de gerenciamento de redes.

É importante que as aplicações de gerenciamento se tornem mais eficientes. Porém, a adição de novas funções a estas aplicações deve se concentrar nas áreas corretas. As informações sobre a interface com o usuário não pertencem às MIBs padrão, mas serão muito úteis se estiverem disponíveis a partir de outras fontes.

## 2.9 Conclusão

Como foi mencionado anteriormente, a MIB é caracterizada pela sua estrutura (organização dos itens e forma de identificá-los) e pelas operações realizadas sobre os mesmos, segundo definição da SMI.

As MIBs da ISO e da Internet são modeladas através das técnicas de programação por objeto. Dentro deste contexto, os recursos a serem gerenciados são representados através de objetos gerenciados.

A grande diferença entre estas MIBs reside nas hierarquias usadas para representar tais objetos. A hierarquia de registros é usada para identificar de maneira universal os objetos tanto nos casos da ISO como no caso da Internet. Em ambos, esta hierarquia é especificada seguindo as regras definidas pela notação ASN.1 usada na atribuição de identificadores de objetos.

Embora a arquitetura de gerenciamento SNMP tenha possibilitado o monitoramento dos nós gerenciadores, ela não provocou a produção de aplicações de gerenciamento “inteligentes”. A principal causa desta situação é que as informações de gerenciamento foram definidas em um nível muito baixo. Foram produzidas diversas MIBs contendo vários objetos gerenciados, mas não foram produzidos documentos descrevendo como estes objetos podem ser usados no gerenciamento eficiente de uma rede. O resultado disto é que a maioria das aplicações de gerenciamento é de *browsers*, que não possuem nenhuma inteligência. Por outro lado, alguns dos produtos desenvolvidos nos últimos anos possuem características mais inteligentes e portanto úteis do que estas ferramentas mais simples.

---

# 3. O protocolo SNMP

O protocolo **SNMP (*Simple Network Management Protocol*)** é a solução adotada na Internet para permitir que gerentes de redes possam localizar e corrigir problemas. Geralmente, é utilizado um processo na máquina do administrador chamado de cliente (uma *workstation* ou um *gateway*, por exemplo) que se conecta a um ou mais servidores SNMP localizados em máquinas remotas, para executar operações sobre os objetos gerenciados (por exemplo, para obter informações sobre estes objetos). O SNMP utiliza o protocolo UDP na comunicação entre cliente e servidor. Para o cliente da rede, o SNMP executa as operações sobre os objetos de forma transparente, o que permite a interface do software de gerenciamento da rede criar comandos imperativos para executar operações sobre os objetos gerenciados. Esta é a grande diferença entre gerenciar uma rede usando o protocolo SNMP e gerenciar a mesma rede usando outros protocolos.

No protocolo SNMP são definidas tanto a sintaxe (forma e a representação dos nomes e do valores) como o significado das mensagens trocadas entre os clientes e os servidores. O formato das mensagens e dos objetos gerenciados de uma MIB são especificados com a linguagem ASN.1 e ao contrário de outros protocolos usados nas redes TCP/IP, suas mensagens não apresentam campos fixos, e portanto, não pode-se representar as mensagens simplesmente com o uso de estruturas fixas.

O SNMP também define as relações administrativas entre os vários *gateways* que estão sendo gerenciados, determinando a autenticação necessária para os clientes acessarem os objetos gerenciados.

Ao contrário dos outros protocolos de gerenciamento que apresentam muitos comandos (operações), o SNMP apresenta somente um conjunto limitado de comandos, baseado num simples mecanismo de busca/alteração. Portanto, é muito mais simples de ser implementado do que um protocolo com muitas operações, em que cada operação sobre um objeto necessita de um comando diferente para implementá-la.

O mecanismo de **busca/alteração** conceitualmente só apresenta duas operações: uma que permite ao cliente alterar

atributos de um objeto de uma MIB (**SET**), e outra para obter os valores dos atributos de um objeto (**GET**). Somente estão disponíveis estas operações (e suas variações) para o gerenciamento da rede, que serão aplicadas sobre os objetos de uma MIB. A principal vantagem de um mecanismo como este é a simplicidade e flexibilidade que este mecanismo dá ao protocolo, o que permite ao SNMP ser um protocolo bem estável porque a sua estrutura básica continuará fixa, mesmo que novos objetos sejam adicionados na MIB, ou que novas operações sejam definidas sobre estes objetos (elas serão constituídas por estas operações básicas).

A MIB define o conjunto e a semântica dos objetos que os servidores SNMP devem controlar, ou seja, define o conjunto conceitual de objetos que um servidor SNMP controla. A MIB é usada para armazenar em seus objetos os estados internos das entidades de uma rede.

Na maioria dos casos, usamos as variáveis convencionais para o armazenamento dos objetos de uma MIB, mas em alguns casos, em que a estrutura interna do TCP/IP não é exatamente compatível com a estrutura de um objeto de uma MIB, é necessário que o SNMP seja capaz de computar os objetos de uma MIB a partir das estruturas de dados disponíveis (simulação deste conjunto conceitual de objetos). Como exemplo, o que um *gateway* deve fazer para saber por quanto tempo um sistema está operacional ? A maioria dos sistemas simplesmente subtrai a hora corrente daquela em que o sistema iniciou mas neste caso, o software poderia simular um "objeto" que contenha o tempo decorrido desde o último start-up deste sistema.

Ao receber e enviar mensagens no protocolo SNMP, os nomes dos objetos não devem ser armazenados na forma textual, e sim na forma numérica definida pela sintaxe ASN.1, que representa o objeto univocamente, com o objetivo de tornar o pacote SNMP mais compacto. Quando a forma numérica que representa um objeto terminar com um zero (como em 1.3.6.1.2.1.4.3.0), representa que o objeto é a única instância existente. Por exemplo, o objeto gerenciavel **iso.org.dod.internet.mgmt.mib.ip.ipInReceives** será representado na mensagem SNMP como **1.3.6.1.2.1.4.3**.

Para minimizar o espaço interno necessário para representar um objeto, e considerando que todos os objetos em uma MIB apresentam o mesmo prefixo no seu nome, podemos retirar o prefixo após a mensagem chegar na máquina, e recolocá-lo imediatamente antes de enviar a mensagem para outra máquina.

Podemos, resumidamente, dizer que os principais objetivos do protocolo SNMP, devido ao protocolo desejar ser flexível e simples, são:

- Reduzir o custo da construção de um agente que suporte o protocolo;
- Reduzir o tráfego de mensagens de gerenciamento pela rede necessárias para gerenciar dos recursos da rede;
- Reduzir o número de restrições impostas as ferramentas de gerenciamento da rede, devido ao uso de operações complexas e pouco flexíveis;
- Apresentar operações simples de serem entendidas, sendo facilmente usadas pelos desenvolvedores de ferramentas de gerenciamento;
- Permitir facilmente a introdução de novas características e novos objetos não previstos ao se definir o protocolo;
- Construir uma arquitetura que seja independente de detalhes relevantes à somente a algumas implementações particulares.

A versão atual do protocolo SNMP é a 2.0 (SNMPv2). A principal diferença entre esta versão e a anterior é a existência de um mecanismo de comunidade melhorado, que apresenta uma identificação inambígua tanto da origem, como do formato da mensagem SNMPv2, permitindo utilizar métodos de acesso mais convencionais aos objetos gerenciados, além de permitir o uso futuro de protocolos assimétricos de segurança,com o uso de chaves públicas.

O SNMP tem como base a técnica "***fetch-store***", ou seja, todas as suas operações previstas são derivadas de operações básicas de busca e armazenamento. Estas operações são:

```
get-request         leitura de uma variável
get-next-request    leitura da próxima variável
get response        resposta a uma operação de leitura
set request         gravação de um campo variável
trap                notificação da ocorrência de um evento
```

Um gerente interage com um agente de acordo com as regras estabelecidas pelo ***framework*** de gerenciamento. Em geral, o gerenciamento da rede impõe ***overheads*** significativos, pois cada nó apenas produz algumas variáveis que serão lidas e usadas para sua monitoração.

# 3.1 Operações disponíveis no protocolo SNMP

Após a definição de como são armazenadas as informações em uma MIB pelas entidades do protocolo SNMP, é importante saber o que deve ser feito com estas informações. O que deve ser feito com os objetos num ambiente de gerenciamento, é definido através das operações aplicadas nos objetos, que são enviadas ao servidor pelo cliente. Duas operações (comandos) básicas no protocolo SNMP, são:

- A **operação SET** é usada por um cliente para alterar um ou mais atributos de um objeto gerenciado (*set-request*);
- A **operação GET** é usada por um cliente para obter o valor(es) de um ou mais atributos de um objeto gerenciado (*get-request* para o pedido e *get-response* para obter o retorno deste pedido).

Uma operação GET ou SET somente se refere a uma **única instância** de um objeto representada através de seu nome. No protocolo SNMP, as operações são **atômicas**, isto é, todas as operações de um pedido devem ser executadas. Não existem execuções parciais de um pedido (no caso, operações aplicadas a múltiplos objetos). Se ocorrer algum erro durante a execução de uma operação, os resultados produzidos por esta operação devem ser ignorados.

Antes de executar um pedido, o servidor deve mapear apropriadamente os nomes dos objetos codificados em ASN.1 nos objetos internos que armazenam as características das entidades da rede (através dos atributos do objeto).

Além das operações padrões, existem mais outras duas operações:

- Numa **operação GET-NEXT** o nome do objeto não só especifica o objeto a acessar (para obter seus atributos, como na operação GET normal), como também é usado para descobrir qual o próximo objeto na sequência léxica. Como retorno, a operação informa o nome do próximo objeto na hierarquia da MIB, e os valores dos seus atributos (obtidos através da execução de uma operação GET normal sobre o objeto).
- Uma **TRAP** que é usada para informar a ocorrência de eventos, permitindo aos servidores SNMP enviarem informações aos clientes sempre que ocorrer algum evento que informa a ocorrência de alterações nos objetos (no protocolo, foram definidas somente algumas *traps*).

A operação GET-NEXT é útil para obter os atributos dos objetos de uma tabela de tamanho desconhecido, pois um cliente pode enviar continuamente requisições GET-NEXT a um servidor que se encarregará de enviar os atributos do objeto e o nome do próximo objeto. Cada novo pedido deve especificar o nome do objeto retornado pelo pedido anterior, o que permite varrermos a tabela sem saber qual o próximo objeto desta tabela. Este processo é chamado de **caminhamento na tabela**. Devido ao ASN.1 não apresentar nenhum mecanismo para implementar tabelas ou para indexá-las, denotamos os elementos individuais (objetos) de uma tabela através de um sufixo.

Para facilitar o uso do comando GET-NEXT em tabelas, alguns nomes de objetos na MIB correspondem tabelas completas ao invés de objetos individuais, não podendo ser usados em uma operação GET (pois esta falhará), mas podem ser usados como parâmetro para a operação GET-NEXT, indicando o primeiro objeto da tabela. Não será necessário conhecer o nome do próximo objeto, pois cada comando GET-NEXT retornará o nome do próximo item da tabela. Executando este processo sucessivamente até que todos os itens da tabela tenham sido acessados, teremos varrido toda a tabela.

A implementação de uma estrutura de dados que suporte o comando GET-NEXT pode ser complicada devido a esta operação poder pular o próximo objeto simples (na ordem lexicográfica) devido a existência de objetos vazios. Como consequência, não pode-se usar simplesmente a ordem lexicográfica presente na árvore para determinar quais objetos satisfazem a um comando GET-NEXT, devendo também existir um programa que examine os objetos, pule aqueles objetos que estejam vazios e descubra o primeiro objeto simples pertencente a um objeto não vazio.

Como exemplo, a tabela ipAddrTable usa o endereço IP do objeto para identificar uma entrada particular na tabela. Se um cliente não souber este endereço IP, não poderá completar o identificador para a realização do pedido, a menos que utilize o prefixo **iso.org.dod.internet.mgmt.mib.ip.ipAddrTable.ipAddrEntry.ipAdEntNetMask** (forma numérica **1.3.6.1.2.1.4.20.1.3**) com a operação GET-NEXT, fazendo o servidor retornar a máscara da primeira entrada da tabela, junto com o nome do próximo objeto desta tabela. O cliente então, usa este nome no próximo pedido enviado ao servidor.

Para o suporte das funções GET, SET e GET-NEXT sobre tabelas, ao contrário do que acontece com objetos simples mapeados em memória, é necessário um software adicional para mapear a tabela numa estrutura interna de dados. No caso das tabelas MIB, o servidor SNMP deve providenciar algum mecanismo que permita a cada tabela ter três funções para implementar as operações GET, SET e GET-NEXT. Para o servidor descobrir qual função deve ser usada, o software que implementa o servidor deve usar a tabela para escolher a função correta, através do uso de um ponteiro para uma tabela que conterá ponteiros para cada uma das operações.

As entradas em uma tabela apontam para outras tabelas que não contém o identificador completo do objeto, mas somente o prefixo deste identificador, porque o identificador completo do objeto para um item da tabela é formado pelo prefixo que identifica a tabela, mais um sulfixo que identifica uma entrada particular na tabela em que o objeto está armazenado.

Uma vez determinado o prefixo correspondente ao objeto e formado o nome do objeto, a função de acesso correspondente a operação pedida é invocada. No caso das tabelas, a função de acesso obtém o sufixo do identificador do objeto, e o usa para selecionar uma das entrada da tabela. Para a maioria das tabelas, é usado o endereço IP para selecionar uma entrada. O endereço IP é codificado no identificador do objeto usando-se a representação decimal com pontos.

Apresentamos, mais uma vez, as operações disponíveis no protocolo SNMP na tabela 3.1 abaixo:

| Comando | Significado |
|---|---|
| **get-request** | Obter atributos de um objeto gerenciado. |
| **get-next-request** | Obter os atributos sem conhecer o nome exato do objeto. |
| **get-response** | A resposta de uma operação de busca de atributos de um objeto. |
| **set-request** | Alterar os atributos de um objeto gerenciado. |
| **trap** | Resposta acionada devido a ocorrência de um evento. |

**Tabela 3.1 - Conjunto de Operações do protocolo SNMP**

# 3.2 Mensagens no protocolo SNMP

Ao contrário de muitos outros protocolos TCP/IP, as mensagens no protocolo SNMP além de não apresentarem campos fixos, são codificadas usando a sintaxe ASN.1 (tanto a mensagem de pedido, como a de resposta) o que dificulta o entendimento e a decodificação das mensagens.

As partes mais importantes de uma mensagem são: as operações (GET, SET e GET-NEXT) e a identificação, no formato ASN.1, dos objetos em que as operações devem ser aplicadas.

Deve existir um cabeçalho que informe o tamanho da mensagem, que só será conhecido após a representação de cada campo ter sido computada. Na verdade, o tamanho da mensagem depende do tamanho de sua parte remanescente (que contém os dados), portanto o tamanho só poderá ser computado após a construção da mensagem. Uma maneira de evitar este problema é construir a mensagem de trás para frente.

Uma mensagem SNMP deve definir o servidor do qual obtemos ou alteramos os atributos dos objetos, e que será responsável por converter as operações requisitadas em operações sobre as estruturas de dados locais. Após verificar os campos de uma mensagem, o servidor deve usar as estruturas internas disponíveis para interpretar a mensagem e enviar a resposta da operação ao cliente que requisitou o pedido.

Uma mensagem é constituída por três partes principais:

- A **versão** do protocolo;
- A identificação da **comunidade**, usada para permitir que um cliente acesse os objetos gerenciados através de um servidor SNMP;
- A **área de dados**, que é dividida em unidades de dados de protocolo *(Protocol Data Units - PDUs*). Cada PDU é constituída ou por um pedido do cliente, ou por uma resposta de um pedido (enviada pelo servidor).

O primeiro campo de uma mensagem SNMP é um operador sequencial, seguido por um campo com o tamanho total da mensagem (se este tamanho não for igual ao do datagrama, será retornado um código de erro). O próximo campo é um número inteiro que identifica a versão do protocolo SNMP, seguido por um campo usado para a autentificação,

indicando a comunidade que o cliente pertence (a comunidade ***public*** permite a qualquer cliente acessar os objetos, não precisando o servidor verificar se o cliente pode ou não acessar o objeto). O quarto campo contem a operação que será executada, devendo ser um GET, SET ou GET- NEXT pois a operação de TRAP só é gerada pelo servidor. O quinto campo é usado para o servidor ter certeza de que o valor deste campo é igual ao tamanho da parte da mensagem que contém os dados. O sexto campo é uma identificação para o pedido, e o sétimo e o oitavo campos são flags que indicam erros quando estão setadas (campos de *status* e de índice de erro).

Na definição de uma mensagem, cada uma das PDUs são constituídas ou por um dos cinco tipos de PDUs para as operações ou por uma PDU para a resposta. Na definição da mensagem SNMP, deve-se ter uma sintaxe individual para cada um das cinco operações da PDU. Alguns termos encontrados nas sintaxes das PDUs das operações são:

- O campo **RequestID** é um inteiro de 4 bytes (usado para identificar as respostas);
- Os campos **ErrorStatus** e **ErrorLevel** são inteiros de um byte (sendo nulos em um pedido de um cliente);
- O campo **VarBindList** é uma lista de identificadores de objetos na qual o servidor procura os nomes dos objetos, sendo definida como uma sequência de pares contendo os nomes dos objetos (em ASN.1 este par é representado como uma sequência de dois itens). Na sua forma mais simples (com um objeto) apresenta dois itens: o nome do objeto e um ponteiro nulo.

Podemos um exemplo de uma mensagem SNMP e de uma PDU na tabela 3.2 abaixo:

**Mensagem SNMP**

```
SNMP-Message ::=
         SEQUENCE {
              version INTEGER {
           version-1 (0)
              },
  community
         OCTET STRING,
  data
         ANY
         }
```

**PDU para cada um dos tipos de operação**

```
SNMP-PDUs ::=
         CHOICE {
                 get-request
                           GetRequest-PDU,
      get-next-request
           GetNextRequest-PDU,
                 get-response
          GetResponse-PDU,
      set-request
          SetRequest-PDU,
      trap
          Trap-PDU
         }
```

**Tabela 3.2 - Exemplo do uso da notação ASN.1**

# 3.3 Servidores e Clientes SNMP

Um servidor SNMP deve ser capaz de aceitar pedidos de operações sobre os objetos gerenciados, executá-los e retornar o resultado das operações após sua execução.

A figura 3.1 ilustra como uma mensagem percorre um servidor SNMP, mostrando que primeiramente a mensagem é interpretada, indicando qual objeto da MIB deve ser mapeado num item de dados local sobre o qual a operação será aplicada. Numa operação de busca, as informações sobre o(s) objetos(s) são retornados na mensagem SNMP de resposta ao pedido, que depois de ser convertida para o formato de uma mensagem SNMP, será enviada ao cliente que solicitou a operação. Se forem solicitadas operações sobre múltiplos objetos (representados na mensagem por seus identificadores), a operação será aplicada a cada um dos objetos presentes na mensagem (passos 3 e 4 na figura).

Um servidor SNMP deve ter um eficiente mapeamento de nomes, pois quando um nome de um objeto na sintaxe ASN.1 chegar ao servidor num pedido, o servidor deverá ser capaz de reconhecer o nome, para chamar o

procedimento correto para executar a operação solicitada no pedido.

Ao invés de manter todas as informações necessárias para atender ao pedido, podemos chamar um procedimento que irá mapear o nome do objeto para a sua representação interna correspondente. A maioria destes procedimentos são rápidos e diretos, pois simplesmente convertem o formato de uma mensagem SNMP (no formato ASN.1) para a representação interna, mas se não existir uma representação para algum objeto no servidor, os pedidos que executem operações sobre este objeto irão requerer mais computação por parte do servidor, e não somente a computação necessária para uma simples tradução de um nome de um objeto MIB para a estrutura de dados local usada para armazenar os dados.

**Figura 3.1 - O fluxo de uma mensagem SNMP dentro de um servidor. Os passos 3 e 4 são repetidos para cada objeto especificado na mensagem (passo com *).**

Após a conversão dos campos da mensagem para a forma interna usada pelo servidor, o pedido será armazenado numa estrutura descritora que contem um ponteiro para uma lista ligada com os nomes de todos os objetos sobre os quais a operação deve ser aplicada. Após serem geradas as respostas, estas devem ser convertidas para que possam ser adicionadas na mensagem de resposta, que será enviada ao cliente que solicitou o pedido. As operações presentes nos pedidos geralmente são executadas por funções no processo servidor.

Na prática, existem vários detalhes que complicam o código do servidor, como o fato de a mensagem SNMP usar a representação ASN.1 em seus campos. Por isso, o servidor não pode usar uma estrutura fixa para descrever o formato da mensagem, mas ao invés de usar uma estrutura não fixa (variável), o servidor pode percorrer a mensagem, analisar cada campo como pode, traduzindo cada um dos campos do formato ASN.1 para o seu formato interno, e traduzindo a resposta para o formato ASN.1 antes de enviá-la ao cliente.

Um cliente SNMP deve construir e enviar o seu pedido ao servidor, esperar pela resposta de seu pedido, e verificar se a resposta concorda com a resposta do que foi pedido. Devido ao protocolo UDP não garantir a entrega dos pacotes, o cliente deve implementar estratégias para *time out* e retransmissão das mensagens que contém os pedidos.

Um cliente só pode obter ou alterar os atributos de um objeto gerenciado somente se tiver permissão para acessar o objeto. Esta permissão é definida através de uma política de acesso. Esta política usa o mecanismo de comunidades (***community***), em que definimos para cada comunidade, um grupo de objetos e de operações que podem ser realizadas sobre estes objetos. Se um cliente não pertencer a comunidade autorizada para acessar o objeto, ou se não tiver autoridade para executar a operação sobre o objeto presente em seu pedido, o pedido será recusado, e será retornada uma mensagem de erro ao cliente, informando que ele não tem direito de acesso ao objeto, ou que ele não pode executar a operação pedida sobre os atributos do objeto. Este mecanismo permite a definição de relações administrativas entre os servidores e os clientes SNMP de uma rede

---

# 4. Gerenciamento no modelo OSI

## 4.1 Os serviços do CMIS e o protocolo CMIP

**> > Protocolo CMIP**

Assim como o protocolo SNMP, o a ISO propõe como solução para gerenciamento de redes o protocolo **CMIP** (**Common Management Information Protocol**), que também define em seu escopo os papéis de gerente e agente que trocam informações sobre os recursos gerenciados e que são armazenados em MIBs.

O protocolo CMIP engloba vários tipos de PDUs, que são mapeadas em operações análogas ao SNMP. São elas:

```
M-Action        execução de alguma ação sobre um objeto gerenciado
M-Create              criação de uma instância de um objeto gerenciado
M-Delete              remoção de uma instância de um objeto gerenciado
M-Get           leitura de atributos dos objetos gerenciados
M-Set           modificação de atributos de objetos gerenciados
MEVENT-REPORT   notificação de um evento associado a um objeto gerenciado
```

Também são definidos recursos adicionais que permitem selecionar o grupo de objetos sobre os quais se aplica uma determinada operação. O ***scoping***, como é chamado este recurso, permite selecionar um grupo instância de objeto sobre os quais se realizará uma única operação.

Por meio dos recursos de **filtro**, outra facilidade do CMIP, é possível definir um conjunto de testes aplicáveis a um grupo de instâncias de objetos, que fora anteriormente selecionado através do ***scoping***. Assim sendo, é possível reduzir significativamente a extração sobre a qual se desenrolará uma operação de gerenciamento.

Além destes, existe o recurso de sincronização, que permite sincronizar diversas operações de gerenciamento realizadas sobre instâncias de objetos selecionados através de recursos de *scoping* e filtro.

Existe uma terceira proposta chamada de CMOT (*CMIP Over TCP/IP*) cujo objetivo é permitir o uso do CMIP em redes com o protocolo TCP/IP.

Se fizermos uma comparação entre o SNMP e o CMIP, veremos que o SNMP é excessivamente simples quando usado em aplicações que não foram previstas quando foi definido, e que apresenta deficiências em relação a segurança ao ser usado em aplicações mais críticas. Já o CMIP é um protocolo poderoso e abrangente, que já foi concebido com o objetivo de adequar-se à complexidade das redes. Mas apesar desta característica, ainda não alcançou um grau de estável de aceitação pela comunidade.

As projeções de mercado demonstram que o SNMP continuará sendo muito usado em pequenas redes, enquanto que o CMIP deve dominar o mercado composto pelas grandes redes corporativas e redes públicas de telecomunicações.

Os serviços do CMIS e o protocolo CMIP são oferecidos na camada de aplicação, sendo usados para implementar sistemas desenvolvidos para vários propósitos, como gerenciamento de desempenho, do nível de falhas, de segurança, de configuração e de contabilidade, usando os recursos de uma rede baseada no modelo de comunicação OSI.

O Serviço de Informação de Gerenciamento (***CMIS - Common Management Information Service***) são os serviços prestados na camada de aplicação. São orientados a conexão, necessitando de um canal virtual (uma associação) para troca de informações. Permite definir os objetos através da seleção de sub-árvores (*scopping*), ou através do uso de predicados (filtragem). O CMIS apresenta os seguintes serviços as aplicações de gerenciamento:

```
GET          Para obter informações (atributos) de um objeto gerenciado;
SET          Para alterar os atributos de um objeto gerenciado;
ACTION       Para executar um comando (ação ou operação) sobre um objeto gerenciado;
CREATE       Para criar uma nova instância de um objeto;
DELETE       Para descartar uma instância de um objeto (removê-la);
EVENT-REPORT Para o relato de ocorrências excepcionais (notificações sobre um evento associado a u
```

Alem das funções apresentadas pelo CMIS no protocolo CMIP, o CMIS apresenta facilidades adicionais que permitem selecionar um conjunto de objetos sobre o qual pode-se aplicar a mesma operação, e também a existência de respostas múltiplas para cada requisição (uma para cada objeto gerenciado). São três facilidades adicionais:

- O ***scoping*** permite que selecionemos um grupo de instâncias de objetos gerenciados sobre o qual aplicaremos uma única operação;
- O **filtro** nos dá a possibilidade de definir um conjunto de testes que serão aplicados a um grupo de instâncias de um objeto, selecionado por uma operação de *scoping* anterior, permitindo formar um grupo menor a partir deste, sobre o qual as operações de gerenciamento devem ser aplicadas;
- A **sincronização** permite que sincronizemos várias operações de gerenciamento a serem aplicadas a instâncias de objetos gerenciados, obtidas através do uso das operações de *scopping* e de filtragem.

O CMISE (***Common Management Information Service Element***) implementa os serviços definidos pelo CMIS, executando o protocolo CMIP. É correspondente ao mecanismo **SASE (*Special Application Service Element***) da camada de aplicação, e utiliza os elementos ACSE (*Association Control Service Element*) e ROSE (***Remote Operations Service Element***) que juntos correspondem ao mecanismo de CASE (***Common Application Service Element***) também da camada de aplicação.

O protocolo CMIP apresenta uma forma inteligível comum utilizada para transferir as informações de gerenciamento entre as entidades pares na comunicação de gerenciamento, sendo que uma destas atua como um gerente, enquanto a outra atua como agente, sendo que as informações são armazenadas em MIBs descritas através da linguagem ASN.1.

Um *framework* que utilize o protocolo CMIP tende a usar a modelagem da orientação a objetos, que encapsula as operações associadas a uma estrutura de dados na própria estrutura. Aqui um agente tem um servidor de objetos gerenciados que pode executar operações de gerenciamento nas variáveis relacionadas a um nó gerenciado. Se este agente for executado em outra máquina (separadamente ao resto do código de gerência), tem-se então o gerenciamento distribuído de rede.

Os serviços oferecidos pelo CMISE ao protocolo CMIP, podem ser **confirmados** ou **não confirmados**. A tabela 4.1 mostra a relação entre os serviços CMISE e as classes de operação do protocolo CMIP. Estes serviços serão mapeados em operações aplicadas sobre os objetos gerenciados, que representam os recursos da rede a serem gerenciados.

| SERVIÇO | TIPO | CLASSE DE OPERAÇÃO |
|---|---|---|
| **M-EVENT-REPORT** | confirmado/não-confirmado | 2 ou 1/5 |
| **M-GET** | confirmado | 2 ou 1 |
| **M-CANCEL-GET** | confirmado | 2 ou 1 |
| **M-SET** | confirmado/não-confirmado | 2 ou 1/5 |
| **M-ACTION** | confirmado/não-confirmado | 2 ou 1/5 |
| **M-CREATE** | confirmado | 2 ou 1 |
| **M-DELETE** | confirmado | 2 ou 1 |

**Tabela 4.1 - Serviço de Informação de gerenciamento comum e Suas Classes deOperação.**

Os serviços oferecidos pelo CMISE e usados pelas aplicações de gerenciamento e para o informe de notificações, são:

```
M-EVENT-REPORT  Reporta um evento de um objeto gerenciado;
M-GET           Solicita a busca de informações de gerenciamento;
M-CANCEL-GET    Solicita o cancelamento de um serviço M-GET previamente requisitado e ainda penden
M-SET             Solicita a modificação da informação de gerenciamento;
M-ACTION        Solicita a execução de uma ou mais ações sobre os objetos gerenciados;
M-CREATE        Solicita a criação de uma instância de um objeto gerenciado;
M-DELETE            Solicita a deleção de uma ou mais instâncias de objetos gerenciados.
```

Define-se as seguintes classes de operação, das quais algumas estão presentes na tabela 4.1:

1. Síncrona relatando sucesso ou falha;
2. Assíncrona relatando sucesso ou falha;
3. Assíncrona relatando somente falhas (erro);
4. Assíncrona relatando somente sucesso;
5. Assíncrona com o resultado não relatado.

E define-se também as seguintes classes de associação:

1. Somente a entidade que iniciou a associação pode invocar operações;
2. Somente a entidade que responde na associação pode invocar operações;
3. Ambas as entidades, a que iniciou e a que responde em uma associação podem invocar operações.

O Protocolo CMIP é executado usando uma classe de associação 3, e as classes de operação 1,2 e 5. O CMIP assume que:

1. O Elemento de Serviço de Controle de Associação (*ACSE - Association Control Service Element*) e que os Protocolos de Unidades de Dados de Aplicação associados ao ACSE (*APDUs - Application Protocol Data Units*) apresentem os serviços da tabela 5.2.
2. O Elemento de Serviço de Operações Remotas (*ROSE - Remote Operations Service Element*) e suas APDUs disponibilizem os serviços da tabela 4.3.
3. Que estejam disponíveis as seguintes PPDUs (*Presentation Protocol Data Units*) usadas pelo ACSE e pelo ROSE para o transporte dos pedidos, para estabelecer e liberar associações de aplicação, e para o envio das notificações, dadas na tabela 4.4.

| APDUs | Serviços | Primitivas de Serviço |
|---|---|---|
| AARQ | A-ASSOCIATE | A-ASSOCIATE-request e indication |
| AARE | | A-ASSOCIATE-response e confirmation |

| | | |
|---|---|---|
| RLRQ | A-RELEASE | A-RELEASE-request e indication |
| RLRE | A-RELEASE | A-RELEASE-response e confirmation |
| ABRT | A-ABORT | A-ABORT-request e indication |

**Tabela 4.2 - Serviços assumidos pelo CMIP em relação ao ACSE e suas APDUs.**

| APDUs | Serviços | Primitivas de Serviço |
|---|---|---|
| ROIV | RO-INVOKE | RO-INVOKE-request e indication |
| RORS | RO-RESULT | RO-RESULT-request e indication |
| ROER | RO-ERROR | RO-ERROR-request e indication |
| RORJ | RO-REJECT | RO-REJECT-request e indication |

**Tabela 4.3 - Serviços assumidos pelo CMIP em relação ao ROSE e suas APDUs.**

| PPDUs | Nomes das PDUs | Primitivas de Serviço |
|---|---|---|
| CP-PPDU | P-Connect Presentation PPDU | P-CONNECT-request e indication |
| CPA-PPDU | P-Connect Presentation Accept PPDU | P-CONNECT-response e confirmation |
| CPR-PPDU | P-Connect Presentation Reject PPDU | P-CONNECT-response e confirmation |
| PD-PPDU | P-Presentation Data PPDU | P-DATA-request e indication |

**Tabela 4.4 - PPDUs utilizadas pelo ACSE e pelo ROSE.**

# 4.2 Conceitos básicos

O gerenciamento no modelo OSI da ISO baseia-se na teoria da orientação a objetos. O sistema representa os recursos gerenciados através de entidades lógicas chamadas de **objetos gerenciados**. Ao desenvolver uma aplicação de gerenciamento, usamos processos distribuídos conhecidos como gerentes (os quais gerenciam) **e agentes** (os que realizam as ações).

Além de definir um modelo informacional, define-se também um modelo funcional em que para cada área é definida um conjunto de funções, que ao serem implementadas, serão usadas para gerenciar a rede.

Existem cinco áreas funcionais no gerenciamento num ambiente OSI:

- Gerência de configuração (estado da rede);
- Gerência de desempenho (vazão e taxa de erros);
- Gerência de falhas (comportamento anormal);
- Gerência de contabilidade (consumo de recursos);
- Gerência de segurança (acesso).

## 4.2.1 Gerentes, agentes e objetos gerenciados

A função de um **processo gerente** é a da coordenação das atividades a serem realizadas, através do envio de solicitações aos processos agentes. Cabe aos **processos agentes** a execução das operações sobre os objetos gerenciados, o envio das respostas as solicitações feitas pelos gerentes, e a emissão de notificações aos gerentes que relatem qualquer alteração ocorrida no estado dos objetos gerenciados. Ao estabelecer uma associação com os processos de aplicação, é possível que o gerente realize operações sobre o objeto ou sobre seus atributos. Este relacionamento entre gerente, agente e objeto gerenciado pode ser visto na figura 4.1.

**Figura 4.1 - Relacionamento Gerente-Agente**

Um **objeto gerenciado** é uma representação lógica de um ou mais recursos de comunicação ou de processamento de dados. Pode-se ter objetos específicos para uma camada, chamados de **objetos gerenciados da camada N,** e objetos usados por mais de uma camada, **chamados de objetos gerenciados do sistema**. Uma MIB é composta por um conjunto contendo estes objetos e seus atributos (contendo informações de gerência).

Definimos um objeto gerenciado através de:

- Seus atributos ou propriedades, que contém informações importantes para representar o recurso que este se relaciona;
- Sua reação às operações que recebe;
- Através de uma notificação, que indica a ocorrência de algum evento;
- As ações (operações) que podemos executar sobre este objeto;
- Seu relacionamento com outros objetos gerenciados.

Para se definir um objeto, usamos a linguagem ASN.1, que descreve os princípios necessários para se especificar os objetos (***GDMO - Guidelines for the Definition of Managed Objects***).

Num ambiente de gerenciamento OSI, usa-se o protocolo CMIP para definir as regras de comunicação entre os processos gerente e agente. O protocolo CMIP implementa as primitivas oferecidas pelo serviço de informação de gerenciamento CMIS.

Este ambiente também propõe uma estrutura de gerenciamento para permitir a definição dos conceitos necessários à construção de classes de objetos gerenciados, os princípios necessários à nomeação dos objetos e dos seus componentes, e como é definido o inter-relacionamento entre os objetos.

Para descrevermos a estrutura, usamos a **Hierarquia de Herança**, **a Hierarquia de Nomeação e a Hierarquia de Registro**.

Na **Hierarquia de Herança** a modelagem é realizada com base nas classes de objetos. Para se obter subclasses com um comportamento mais particular, deve-se detalhar uma superclasse, gerando a partir desta subclasses para um propósito mais particular do que esta classe.

Na **Hierarquia de Nomeação** é descrita a relação de composição entre os objetos, ou seja, a relação subordinado-superior entre estes objetos, além de serem definidas as regras usadas para nomear os objetos (*name binding*), de forma que este seja univocamente determinado.

Na **Hierarquia de Registro** é registrada as definições das classes dos objetos, os atributos dos objetos, as ações que podem ser aplicadas, as notificações geradas e os pacotes, seguindo as regras definidas pela notação ASN.1.

Um problema que este modelo apresenta é a existência de mais complexidade ao se construir os agentes, mas apesar desta desvantagem, e de os agentes consumirem mais recursos da rede, o uso da rede é otimizado, devido a minimização dos pedidos de informação (*polling*) necessários para obter dados sobre o objeto gerenciado, além de deixar que o gerente realize as tarefas mais específicas.

Devido a hierarquia introduzida por este modelo, é possível que um mesmo processo tenha ao mesmo tempo, a função de gerente e de agente, sendo chamado de **gerente intermediário**. Assim pode-se distribuir as tarefas entre os gerentes intermediários, de forma que cada um seja responsável por gerenciar um certo domínio da rede.

## 4.2.2 Modelo de Gerenciamento OSI

O modelo de gerenciamento OSI é definido com base em três conceitos:

- Considerando-se a estrutura de gerenciamento;
- Considerando-se as MIBs;
- Usando-se alem destes, outros conceitos.

Na estrutura de gerenciamento, temos três tipos de gerenciamento:

- **Gerenciamento de sistemas**: É um protocolo executado na camada de aplicação, que é responsável pelo gerenciamento dos sistemas. Pode-se gerenciar aqui quaisquer objetos associados a um sistema aberto. Este gerenciamento necessita do apoio das funções de todas as sete camadas do modelo OSI para poder realizar o gerenciamento;
- **Gerenciamento de camada**: Este gerenciamento é realizado sobre os objetos gerenciados relacionados as atividades de uma camada particular, e usa os protocolos de gerenciamento específicos para a camada, e as funções de apoio internas desta camada. Os protocolos de gerenciamento de propósito especial não prestam serviços a camadas superiores, e são independentes dos protocolos de gerenciamento das outras camadas.

- **Operação de camada**: É usada no gerenciamento de uma única instância de comunicação em uma camada. É um tipo de gerência que exige menores requisitos das funções de apoio, por não ser necessário um protocolo particular para a troca de informações de gerenciamento, pois utiliza-se do protocolo normal da camada para trocar estas informações.

Uma MIB é usada para armazenar as informações transferidas ou modificadas quando são usados os protocolos de gerenciamento OSI. As informações podem ou ser fornecidas ou por agentes administrativos locais ou por sistemas abertos remotos. É disponibilizada uma interface MIB para cada uma das sete camadas, que oferece as operações necessárias ao gerenciamento da rede em cada uma das camadas.

Uma interface específica para cada camada é obtida através das **Entidades de Gerenciamento de Camadas (*LME - Layer Management Entities*)**. Cada LME contém a funcionalidade da camada a que está relacionada. A integração destas entidades e a execução da função de interfacamento com o gerente é executada pela **Entidade de Aplicação de Gerenciamento de Sistema (*SMAE - System Management Application Entity*)**. Esta entidade também providencia a interface entre as LMEs de um nó da rede com as suas correspondentes no outro nó, usando o Protocolo de Informação de Gerenciamento (*CMIP)*, como pode ser visto na figura 4.2.

Agrupamos em **unidades funcionais** todos os serviços fornecidos por alguma função de gerenciamento de sistema. Estas unidades são básicas para a negociação entre os **Usuários do Serviço de Informação de Gerenciamento (*MIS-Users - Management Information Service-Users*)**, que são aplicações que utilizam os serviços de gerenciamento, e que podem desempenhar tanto a função de um agente, como a de um gerente. Quando tem a função de agente, o MIS-User é parte de alguma aplicação distribuída que controla os objetos gerenciados no seu domínio (ambiente local), e realiza operações sobre os objetos gerenciados em função dos comandos enviados pelo gerente, podendo também enviar notificações dos objetos gerenciados aos gerentes. Os papéis designados ao MIS-Users não são permanentes, podendo este, dependendo do contexto, ter a função de um agente, de um gerente, ou ambas (o papel que o processo tem será definido com base no processo com o qual este processo interage).

**Figura 4.2 - Modelo de Gerenciamento OSI**

# 4.3 Componentes do Modelo de Gerenciamento OSI

Como o ambiente a ser gerenciado é distribuído, as atividades de gerência também devem ser distribuídas. Uma instância de uma aplicação distribuída pode ser formada por uma associação de duas ou mais aplicações de gerenciamento do sistema.

As interações entre os sistemas são feitas através das operações de gerenciamento e das notificações, sendo que uma entidade tem a função de um gerente, solicitando ações de gerenciamento a outra entidade que tem a função de agente, executando as operações e enviando as notificações emitidas pelos objetos gerenciados (ver figura 4.3).

**Figura 4.3 -Suporte de Comunicação para Notificações e Operações de Gerenciamento.**

Um pedido de operação que chega a um agente é rejeitado, a menos que os mecanismos que controlam os acessos aos objetos gerenciados permitam ao gerente realizar as operações solicitadas sobre estes objetos. Sempre que existem notificações enviadas pelos objetos gerenciados, o sistema gerenciado (agente) envia as notificações aos gerentes.

Para a execução das atividades acima, dois aspectos são necessários na comunicação:

Suporte para a transferência dos pedidos de operações de gerenciamento e para o envio de notificações entre MIS-Users;

Suporte para controlar o acesso aos objetos gerenciados, e para a distribuição das informações das notificação.

## 4.3.1 Aspectos das Comunicações

Os sistemas abertos gerenciados utilizam o protocolo OSI para se comunicarem. O modelo OSI apresenta serviços gerais para gerenciamento chamados de CMIS Além disso, os MIS-Users podem utilizar outros serviços além dos fornecidos pelo modelo OSI.

Uma associação entre dois SMAEs é realizada através de um contexto de aplicação que define o conhecimento inicial de gerenciamento compartilhado, e os vários ASEs (*Applicaton Service Elements*) que podem ser usados. Uma SMAE é formada por:

- Elemento de Serviço de Aplicação de Gerenciamento de Sistema (*SMASE - Systems Management Application Service Element*), que especifica a semântica e a sintaxe abstrata da informação que é transferida pelas Unidades de Dados do Protocolo de Aplicação de Gerenciamento (*MAPDUs - Management Application Protocol Data Units*), além de especificar as informações de gerenciamento que devem ser trocadas entre duas SMAEs.
- Elemento de Serviço de Controle de Associação (*ACSE - Association Control Service Element*);
- E por Elementos de Serviço de Aplicação (*ASEs - Application Service Elements*).

Os serviços de comunicação necessários a uma SMASE podem ser prestados por um Elemento de Serviço de Informação de Gerenciamento Comum (*CMISE)* ou por vários ASEs, como o de Transferência, Acesso e Gerenciamento de Arquivos (*FTAM - File Transfer, Access and Management*), ou pelo Processamento de Transações (*TP - Transaction Processing*).

O CMISE define os serviços e procedimentos necessários para transferir as Unidades de Dados do Protocolo Comum de Informação de Gerenciamento (*CMIPDUs - Commom Management Information Protocol Data Units*) e fornece um meio para a troca de informações usadas pelas operações de gerenciamento. Para usá- lo é necessário usar um Elemento de Serviço de Operações Remotas (*ROSE - Remote Operations Service Element*).

### 4.3.2 Conhecimentos de Gerenciamento

As Informações de gerenciamento que são compartilhadas entre os SMAEs são chamadas genericamente de **conhecimento de gerenciamento compartilhado (*SMK - Shared Management Knowledge*)**. Este conhecimento pode ser estabelecido em qualquer momento, em especial, antes de se estabelecer uma associação, durante o seu estabelecimento, ou durante o seu tempo de vida. Pode-se também definir ou alterar o conhecimento de gerenciamento ao se estabelecer a associação. Esta visão de compartilhamento de informações pode ser vista na figura 5.6.

O conhecimento de gerenciamento compartilhado deve conter, dentre outras coisas, os seguintes elementos:

- O protocolo utilizado;
- As funções e unidades funcionais suportadas;
- As informações sobre os objetos gerenciados;
- As restrições nas funções suportadas, e as relações entre as funções e os objetos gerenciados.

### 4.3.3 Domínios Gerenciais

Um Domínio Gerencial é uma forma de organizar o ambiente de gerenciamento OSI e ocorre quando organizamos os objetos em conjuntos, de forma que o ambiente seja divido de acordo com as regras abaixo (veja melhor o conceito na figura 4.4):

- Divide-se o ambiente de gerenciamento OSI em partes que tenham um mesmo propósito funcional (como falha, segurança, contabilização, desempenho ou configuração), ou o mesmo propósito de gerenciamento (estrutura geográfica, tecnológica ou organizacional);
- Definir temporariamente e, provavelmente, alterar os papéis dos gerentes e agentes para cada um dos propósitos definidos, dentro de cada um dos conjuntos de objetos gerenciados definidos;
- Executar as regras de controle de tal forma que estas sejam consistentes (por exemplo, política de segurança).

**Figura 4.4 - Visão do Conhecimento de Gerenciamento Compartilhado.**

# 4.4 Áreas Funcionais no Gerenciamento OSI

O objetivo do gerenciamento OSI é o de resolver os problemas relativos a configuração de uma rede, as falhas que possam ocorrer nos componentes, aos níveis de desempenho que a rede apresenta, a segurança que esta apresenta e a

contabilização de sua utilização. Estas diferentes partes que ocorrem num problema de gerenciamento de redes são chamadas de **Áreas Funcionais de Gerência**.

Estas áreas funcionais são constituídas por processos de aplicação de gerenciamento residentes na camada OSI de aplicação. Os dados necessários para o funcionamento das diversas áreas funcionais estão em uma MIB que inclui os objetos gerenciados, seus atributos, as operações que podem ser executadas e as notificações que estes podem enviar. Estes relacionamentos podem ser vistos na figura 4.5.

A ISO define como deve ser o formato para se representar as informações de gerenciamento, e as ferramentas para coletar as informações e controlar os objetos gerenciados (que são definidos como estruturas de dados especificadas com a linguagem ASN.1), mas como os dados são tratados, e como os resultados devem ser apresentados não são padronizados pela ISO.

Para serem atendidos os requisitos necessários as áreas funcionais, foram definidas as seguintes funções de gerenciamento:

**Função de Gerenciamento de Objeto (*OMF - Object Management Function*) [ISO10164-1]**

Objetiva gerenciar a criação e a remoção de um objeto gerenciado, e o exame ou alterações nos atributos de um objeto gerenciado. Apresenta funções para gerar relatórios de criação/remoção de objetos, e relatórios de mudanças nos nomes e valores dos atributos dos objetos gerenciados. Aqui é descrito como devemos usar o serviço PASS-THROUGH para mapear uma operação de gerenciamento para o serviço correspondente do CMISE.

**Figura 4.5 - Conceitos de Domínios Gerenciais**

**Função de Gerenciamento de Estado (*STMF - State Management Function*) [ISO10164-2]**

Esta função é usada para representar as condições instantâneas que se referem a disponibilidade e operacionalidade de um recurso sob a visão do gerenciamento. Cada classe de objetos gerenciados tem o seu próprio conjunto de atributos de estado, que são usados para expressar e controlar os aspectos de operação dos recursos associados a cada classe.

A função de gerenciamento de estado deve ser padronizada, pois deve ser comum a um grande número de recursos gerenciados, e expressa os aspectos essenciais que se referem a operacionalidade de um recurso num dado intervalo de tempo.

Tem como objetivo o controle da disponibilidade geral deste recurso, tornando as informações sobre esta disponibilidade visíveis, e no caso do recurso estar inoperante, a função define quais ações devem ser tomadas para colocá-lo operante.

Deve fornecer definições genéricas para permitir a obtenção de informações, a mudança do estado de um dos objetos, e a emissão de notificações sobre as mudanças no estado de um objeto, sempre que decorrerem de alguma operação realizada no sistema aberto.

São definidos dentro do escopo de gerenciamento de estado, três fatores que afetam o estado de gerenciamento de um objeto em relação a disponibilidade do recurso associado a este objeto:

- **Operacionalidade:** Se um dado recurso está ou não instalado, e no caso de estar instalado, se está ou não em operação;
- **Utilização:** Se um dado recurso está ou não em uso em um dado instante de tempo, e se este é ou não capaz de aceitar mais outros usuários adicionais;
- **Administração:** Através dos serviços de gerenciamento, impõe-se a permissão ou proibição do uso de um dado recurso.

**Figura 4.6 - Áreas Funcionais do Gerenciamento OSI**

**Atributos para Representação de Relacionamento (*ARR - Attributes for Representing Relationship*) [ISO10164-3]**

Um relacionamento é um conjunto de regras que são usadas para descrever como uma operação realizada numa parte de um sistema aberto poderá afetar alguma outra parte deste sistema.

Um relacionamento existe entre dois objetos gerenciados quando uma operação executada em um deles, afeta uma operação executada no outro. Para que este relacionamento seja reconhecido no modelo OSI, devem ser conhecidas informações suficientes de gerenciamento, que permitam ao Usuário do Serviço de Informação de Gerenciamento identificar quais são os objetos gerenciados envolvidos, e quais as regras que governam as suas interações.

A partir de modelos e conceitos definidos no padrão da ISO, foram definidos os seguintes atributos de relacionamento: **objeto provedor, objeto usuário, atributo par, primário, secundário, identificação da instância de objeto Backup, identificação da instância de objeto Backed-up, membro, proprietário e grupo de atributos de relacionamento.**

**Função de relatório de alarme (*ARF - Alarm Report Function*) [ISO10164-4]**

Tem como objetivo fornecer informações que permitam ao gerente atuar sobre as condições operacionais e a qualidade do serviço de um sistema gerenciado. Devem ser definidos critérios para identificar um mal funcionamento no sistema gerenciado, em função da ocorrência de falhas, que permitam avaliar qual o grau de mal funcionamento do recurso.

O nível de severidade do alarme é avaliado em função do nível de degradação que este irá provocar na qualidade do serviço oferecido ao usuário deste sistema, ou pelo estado da capacidade de uso de um determinado objeto gerenciado.

Podemos ter diversos níveis de severidade de alarme, deste um nível de alerta que não provoca nenhuma degradação sobre o serviço prestado ao usuário, até um alarme crítico que diz que o serviço não pode mais ser fornecido ao usuário.

**Função de Gerenciamento de Relatório de Evento (*ERMF - Event Report Management Function*) [ISO10164-5]**

A função de gerenciamento de Relatório de Evento tem como objetivos:

- Definir um serviço para o controle de relatórios de eventos que permita selecionar quais relatórios devem ser enviados para um sistema de gerenciamento particular;
- Definir quais devem ser os destinos dos relatórios de eventos gerados;
- Definir um mecanismo de transmissão de relatórios que permita o controle sobre o repasse destes relatórios;
- Possibilitar que um sistema de gerenciamento externo altere as condições usadas para emitir os relatórios;
- Definir endereços secundários (usados para *back-up*) aos quais enviamos os relatórios de eventos, caso o endereço primário não esteja disponível.

**Função para o controle de Log (*LCF - Log Control Function*) [ISO10164-6]**

O objetivo de função de controle de log (um repositório de dados que contém registros com informações que devem ser preservadas) é o de permitir as demais funções de gerenciamento, preservar informações sobre os eventos que ocorreram, ou sobre as operações executadas nos objetos gerenciados. Uma vez que estas informações podem mudar, a função de controle de log deve satisfazer as seguintes características:

- O Controle de Log deve ser flexível para permitir a seleção de quais registros do log devem ser preservados pelo sistema de gerenciamento;
- Deve permitir que um sistema externo altere os critérios usados na preservação dos registros;
- Deve permitir a um sistema externo saber se foi alterada alguma característica de preservação, ou se um registro foi perdido;
- Definir mecanismos para controlar o tempo durante o qual devem ser realizadas as atividades de preservação das informações;
- Deve permitir que um sistema externo recupere e elimine os registros em um log, como também criar e eliminar logs.

**Função de Relatório de Alarme de Segurança (*SARF - Security Alarm Reporting Function*) [ISO10164-7]**

É a função de gerenciamento do sistema pela qual um usuário do gerenciamento de segurança recebe as notificações sobre os eventos relacionados a segurança da rede. Este usuário deve saber quais as operações que falharam (*miss operations*) nos serviços e mecanismos de segurança, os atentados (*attacks*), e as violações (*breaches*) a esta segurança, quando estes atentados foram detectadas pelos mecanismos de segurança, além de outros processamentos relacionados com a segurança do sistema. Deve-se também notificar ao usuário a gravidade das operações erradas, dos atentados e violações na segurança do sistema.

**Função de Registro para Auditoria de Segurança (*SATF - Security Audit Trail Function)* [ISO10164-8]**

O usuário do gerenciamento de segurança usa esta função para gravar todos os eventos potenciais relacionados à segurança no seu domínio de gerenciamento. Estas informações são gravadas num objeto de log de auditoria de segurança (*security audit log*).

Através de uma comparação com utilização planejada do sistema de gerenciamento e a utilização real gravada neste log, o usuário pode saber qual é o grau de atendimento dos requisitos definidos pela política de segurança.

Uma análise ou auditoria dos relatórios de alarmes de segurança possibilita que o usuário detecte desvios no uso das normas da política de segurança, e correlacione estes desvios com os alarmes de segurança de menor severidade, ou com qualquer outro evento normal, para descobrir quais são os pontos vulneráveis ou quais partes do mecanismo de segurança que estão funcionando precariamente.

Para a execução desta auditoria, é necessário que estejam no log todos os eventos relativos à segurança como: as conexões, as desconexões, todos os eventos relativos à utilização dos mecanismos de segurança, as próprias operações para gerenciamento da rede, e a contabilização da utilização de cada recurso gerenciado.

O log pode ser local, em que os registros podem ser recuperados por um gerente, ou remoto, em que enviamos os registros ao gerente sempre que eventos relativos a segurança ocorrem.

Resumidamente, o usuário do gerenciamento de segurança, precisa ter a capacidade de controle dos mecanismos de auditoria de segurança em relação a sua operação, e na escolha dos eventos de interesse do sistema que devem ser auditados, para que seja possível perceber atentados contra a segurança, ou problemas ao se concretizar qual deve ser a política de segurança a ser adotada.

**Função de Registro para Controle de Acesso ou Função de Controle de Acesso (*OAAC - Objects and Attributes for Access Control*) [ISO10164-9]**

O modelo de controle de acesso faz parte do gerenciamento dos mecanismos de segurança descritos na arquitetura de segurança do modelo OSI. Este controle de acesso aos objetos envolvidos na gerência da arquitetura OSI, é uma interpretação do modelo básico usado pelas aplicações de gerenciamento.

No contexto do gerenciamento da segurança do sistema, pode permitir ao administrador de um domínio prevenir-se de acessos não autorizados aos recursos. Para isso, são disponibilizados mecanismos para controle do acesso, para que somente usuários autorizados possam ter acesso a um recurso de gerenciamento específico. Deve-se também evitar o envio das notificações a destinatários não-autorizados, impedir o acesso as operações de gerenciamento por entidades que não sejam autorizadas, e proteger as informações de gerenciamento contra sua divulgação indesejável.

**Função de Medida de Contabilização (*AMF - Accounting Metering Function*) [ISO10164-10]**

Define os mecanismos necessários para a coleta de informações sobre a utilização dos recursos no ambiente OSIE (*Open System Interconnection Environment*), uma representação para que estas informações sejam armazenadas de maneira adequada, e associar tarifas as medidas de utilização de cada um dos recursos gerenciados.

Definem-se objetos de dados e de controle de medida de contabilização. Pode-se realizar operações sobre as instâncias destes objetos para se obter informações sobre a utilização de um recurso, iniciar, retomar e suspender as medidas de contabilização do uso do recurso, e também manter um registro dessa medida.

**Função de Monitoração de Carga de Trabalho (*WMF - Workload Monitoring Function*) [ISO10164-11]**

O seu objetivo é a avaliação da demanda necessária de um recurso e a real utilização de um dado recurso do OSIE, além da avaliação da eficiência das atividades de comunicação. Deve incluir as seguintes funções:

- Obtenção de informações estatísticas;
- Manutenção e análise dos registros do histórico do sistema;
- Determinação do desempenho do sistema sob condições naturais e artificiais;
- Alteração do modo de operação do sistema, com objetivo de realizar atividades referentes ao gerenciamento do seu desempenho.

**Função de Gerenciamento de Teste (*TMF - Test Management Function*) [ISO10164-12]**

Objetiva satisfazer o controle remoto de testes, além de estabelecer a estrutura básica dos testes a serem realizados sobre os recursos gerenciados. A necessidade de execução de operações de teste pode ser necessária em diferentes áreas funcionais.

Um teste é uma operação de monitoração de um sistema aberto (ou parte deste sistema aberto), num ambiente de gerenciamento que permita obter informações sobre a funcionalidade e/ou desempenho dos sistemas que são sujeitos aos testes. O teste necessita da criação de um ambiente de teste, de uma operação de teste, e finalmente, o retorno ao ambiente normal após a execução do teste.

O objetivo de uma operação de teste é o de monitorar e controlar um sistema. O controle inclui atividades como a suspensão, a reinicialização ou o término do teste. Deve-se identificar cada um dos testes univocamente. Pode-se realizar os testes de acordo com uma programação que pode ter tanto testes periódicos, quanto esporádicos. Pode-se combinar testes simples para criar-se testes complexos.

Em alguns casos, pode ser necessária a execução de um conjunto de testes particulares para alguma necessidade específica, e logo após o término do teste, deve-se correlacionar os resultados de cada teste, para a formulação do resultado final.

**Função de Sumarização (*SF - Summarization Function*) [ISO10164-13]**

Este função é usada para obter informações a partir de observações relativas a múltiplos objetos gerenciados. Para isso, deve-se incluir os relatórios de eventos, e o escalonamento das observações ao se especificar as funções.

São definidos métodos para a observação e o relato de valores dos atributos dos objetos gerenciados, determinados métodos para o relato de estatísticas com base em diversos valores de atributos, sendo que cada um destes foi observado em um mesmo instante. Os valores dos atributos e as estatísticas fornecem uma informação sumarizada do conjunto de objetos gerenciados e seus atributos, em um ou mais intervalos de tempo distintos. Como consequência, as estatísticas são calculadas em função do conjunto de objetos gerenciados e não em relação ao tempo.

Em resumo, ela suporta a habilidade para agregar os valores de atributos observados e/ou disponibilizar informações estatísticas sobre estes valores de atributo.

## 4.4.1 Gerência de Configuração

O objetivo da gerência de configuração é o de permitir a preparação, a iniciação, a partida, a operação contínua, e a posterior suspensão dos serviços de interconexão entre os sistemas abertos, tendo então, a função de manutenção e monitoração da estrutura física e lógica de uma rede, incluindo a verificação da existência dos componentes, e a verificação da interconectividade entre estes componentes.

A gerência de configuração portanto, é correspondente a um conjunto de facilidades que permitem controlar os objetos gerenciados, indentificá-los, coletar e disponibilizar dados sobre estes objetos para as seguintes funções:

- Atribuição de valores iniciais aos parâmetros de um sistema aberto;
- Início e encerramento das operações sobre objetos gerenciados;
- Alteração da configuração do sistema aberto;

- Associação de nomes a conjuntos de objetos gerenciados.

### 4.4.2 Gerência de Desempenho

Na gerência de desempenho temos a possibilidade de avaliar o comportamento dos recursos num ambiente de gerenciamento OSI para verificar se este comportamento é eficiente, ou seja, preocupa-se com o desempenho corrente da rede, através de parâmetros estatísticos como atrasos, vazão, disponibilidade, e o número de retransmissões realizadas.

O gerenciamento de desempenho é um conjunto de funções responsáveis pela manutenção e exame dos registros que contém o histórico dos estados de um sistema, com o objetivo de serem usados na análise das tendências do uso dos componentes, e para definir um planejamento do sistema através do dimensionamento dos recursos que devem ser alocados para o sistema, com o objetivo de atender aos requisitos dos usuários deste sistema, para satisfazer a demanda de seus usuários, ou seja, garantir que não ocorram insuficiências de recursos quando sua utilização se aproximar da capacidade total do sistema.

Para atingir estes objetivos, deve-se monitorar taxa de utilização dos recursos, a taxa em que estes recursos são pedidos, e a taxa em que os pedidos a um recurso são rejeitados. Para cada tipo de monitoração, definimos um valor máximo aceitável (*threshold*), um valor de alerta, e um valor em que se remove a situação de alerta. Definem-se três modelos para atender aos requisitos de monitoração do uso dos recursos do sistema:

- **Modelo de Utilização:** Provê a monitoração do uso instantâneo de um recurso.
- **Modelo de Taxa de Rejeição:** Provê a monitoração da rejeição de um pedido de um serviço.
- **Modelo de Taxa de Pedido de Recursos:** Provê a monitoração dos pedidos do uso de recursos.

### 4.4.3 Gerência de Falhas

A gerência de falhas tem a responsabilidade de monitorar os estados dos recursos, da manutenção de cada um dos objetos gerenciados, e pelas decisões que devem ser tomadas para restabelecer as unidades do sistema que venham a dar problemas.

As informações que são coletadas sobre os vários recursos da rede podem ser usadas em conjunto com um mapa desta rede, para indicar quais elementos estão funcionando, quais estão em mal funcionamento, e quais não estão funcionando.

Opcionalmente, pode-se aqui gerar um registro das ocorrências na rede, um diagnóstico das falhas ocorridas, e uma relação dos resultados deste diagnóstico com as ações posteriores a serem tomadas para o reparo dos objetos que geraram as falhas.

O ideal é que as falhas que possam vir a ocorrer em um sistema sejam detectadas antes que os efeitos significativos decorrentes desta falha sejam percebidos. Pode-se conseguir este ideal através da monitoração das taxas de erro do sistema, e da evolução do nível de severidade gerado pelos alarmes (função de relatório de alarme), que permite emitirmos as notificações de alarme ao gerente, que pode definir as ações necessárias para corrigir o problema e evitar as situações mais críticas.

### 4.4.4 Gerência de Contabilidade

A gerência de Contabilidade provê meios para se medir e coletar informações a respeito da utilização dos recursos e serviços de uma rede, para podermos saber qual a taxa de uso destes recursos, para garantir que os dados estejam sempre disponíveis quando forem necessários ao sistema de gerenciamento, ou durante a fase de coleta, ou em qualquer outra fase posterior a esta. Deve existir um padrão para obtenção e para a representação das informações de contabilização, e para permitir a interoperabilidade entre os serviços do protocolo OSI.

A função de contabilização deve ser genérica para que cada aplicação trate os dados coerentemente de acordo com as suas necessidades. Estas funções podem ser usadas para várias finalidades como tarifas sobre serviços prestados, controle de consumo dos usuários, etc.

A função de contabilização é implementada através de objetos gerenciados especiais associados à contabilização (a

utilização dos recursos ligados a estes objetos que representam as características de um dado recurso monitorado) chamados de **"Objetos Contabilizados"**. Existem dois tipos de objetos: **Objetos de Controle de Medida de Contabilização e Objetos de Dados de Medida de Contabilização**.

Os **Objetos de Controle de Medida de Contabilização** permitem que o sistema ao coletar as informações sobre o uso de um determinado recurso, selecione quais dados são relevantes, além de permitir que este sistema defina sobre quais circunstâncias deve ser realizada a coleta.

Este controle irá definir quais eventos são gerados ao se atualizar e notificar as informações sobre o uso de um recurso. Apresenta uma visão genérica de gerenciamento, para ser particularizada para a contabilização de recursos específicos, além de usar os pacotes especificados num controle de medida, para incorporar as funcionalidades necessárias a contabilização. Alguns tipos de eventos que podem ocorrer são:

- Escalonamento por períodos de tempo;
- Ações de controle do próprio sistema de gerenciamento;
- Estímulos provenientes da mudança de valores dos atributos.

Os **Objetos de Dados de Medida de Contabilização** são usados para representar um recurso usado por um usuário, contendo informações como: qual é o usuário do recurso, qual a unidade de medida usada na contabilização, qual a quantidade consumida, etc. Estas informações podem ser obtidas através de um GET para a obter os valores dos atributos dos dados de medida, ou através do uso de parâmetros nas notificações enviadas pela gerência de contabilização. Novamente são definidas apenas propriedades genéricas que podem ser especializadas conforme a necessidade.

Os objetos de dados de medida só podem ser criados se existir uma instância de um objeto de controle de medida para controlá-lo. Um objeto de controle de medida só pode ser destruído quando todos os objetos de dados de medida controlados por este objeto forem também destruídos. Uma instância de um objeto de controle de medida pode controlar várias instâncias de objetos de dados de medida.

Devemos sempre ter pelo menos uma instância do objeto de dados na memória que seja responsável por monitorar um objeto contabilizado, para que possamos enviar solicitações sobre seu uso.

A figura 4.7 mostra o relacionamento entre os objetos de controle de medida, os objetos de dados de medida e os objetos contabilizados. Nela, notamos que os objetos de controle de medida (***AccMeterControlObj***) referenciam os objetos de dados de medida que controlam (***AccMeterDataObj***). Cada um dos objetos de dados de medida contém uma referência a uma instância do objeto contabilizado, em que realizamos as coletas das informações de contabilização.

**Figura 4.7 - Relacionamento entre os objetos gerenciados**

Ao implementamos a função de gerência de contabilização, devemos considerar os seguintes aspectos:

- Controlar o registro e a emissão dos dados relacionados a contabilização através dos objetos de controle de medida de contabilização;
- Coletar os dados de contabilização, usando os objetos de dados de medida de contabilização para representar os recursos contabilizados;
- Armazenar os resultados da contabilização para criar históricos de contabilização dos recursos através do uso de registros de contabilização.

## 4.4.5 Gerência de Segurança

O objetivo do gerenciamento de segurança é o de dar subsídios à aplicação de políticas de segurança, que são os aspectos essências para que uma rede baseada no modelo OSI seja operada corretamente, protegendo os objetos gerenciados e o sistema de acessos indevidos de intrusos. Deve providenciar um alarme ao gerente da rede sempre que se detectarem eventos relativos a segurança do sistema.

São distinguidos dois conceitos no modelo OSI em relação a segurança:

- Arquitetura de Segurança do Modelo OSI;

- Funções de Gerenciamento de Segurança, estas compondo a área funcional de gerência de segurança.

O objetivo da Arquitetura de Segurança do modelo OSI é o de dar uma descrição geral dos serviços de segurança e dos mecanismos associados a este, e de definir em que posição do modelo de referência situam-se os serviços de segurança e os seus mecanismos associados. A norma de referência da Arquitetura de Segurança trata exclusivamente da segurança dos canais de comunicação, através de mecanismos como a criptografia, a assinatura numérica, e a notarização, que permite aos sistemas que usam este canal se comunicarem de forma segura. Para isso, define-se os seguintes serviços:

- Autenticação tanto de entidades pares quanto da origem dos dados (*authentication*);
- Controle de acesso aos recursos da rede (*access control*);
- Confidencialidade dos dados (*confidenciality*);
- Integridade dos dados (*integrity*);
- A não-rejeição ou não-repudiação (*non-repudiation*);

Os mecanismos a serem adotados dependem do uso de uma política de segurança, que é feita pelo uso das Funções de Segurança do Gerenciamento de Redes OSI. Estas funções tratam do controle dos serviços de segurança do modelo OSI, e dos mecanismos e informações necessárias para se prestar estes serviços.

Então, os objetivos do gerenciamento de segurança são:

- O fornecimento de relatórios de eventos relativos à segurança e o fornecimento de informações estatísticas;
- A manutenção e análise dos registros de histórico relativos à segurança;
- A seleção dos parâmetros dos serviços de segurança;
- A alteração, em relação a segurança, do modo de operação do sistema aberto, pela ativação e desativação dos serviços de segurança.

Para que estes objetivos sejam atingidos, deve-se olhar as diferentes políticas de segurança a serem adotadas no sistema aberto. Todas as entidades que seguem uma mesma política de segurança pertencem ao mesmo domínio de segurança.

Devido ao gerenciamento necessitar distribuir as informações de gerenciamento de segurança entre todas as atividades que se relacionam com a segurança, os protocolos de gerenciamento assim como os canais de comunicação devem ser protegidos, usando os mecanismos previstos na arquitetura de segurança.

As informações de gerenciamento de segurança são armazenadas numa MIB especial que deve dar apoio as três categorias de atividades de gerenciamento de segurança existentes. Esta MIB é chamada de SMIB (*Security Management Information Base*).

## 4.5 A Plataforma OSIMIS

OSIMIS é uma plataforma de gerência orientada a objetos e desenvolvida principalmente na linguagem C++. Através do encapsulamento dos detalhes existentes no acesso aos serviços de gerenciamento, ela fornece um ambiente para o desenvolvimento de aplicações com uma interface orientada a objetos, o que permite aos desenvolvedores se preocuparem com a construção da aplicação, ao invés dos detalhes necessários para se acessar um serviço/protocolo de gerência. Usa o modelo OSI gerente-agente e os objetos gerenciados para abstrair os recursos reais. Não há nenhuma restrição para que aplicação desempenhe os dois papéis ao mesmo tempo (de agente e gerente).

A plataforma OSIMIS originou-se dos resultados obtidos das pesquisas voltadas para a área de gerenciamento de sistemas de comunicações e de sistemas distribuídos, realizada nos últimos anos (pelos europeus). Atualmente novos recursos vêm sendo acrescentados à arquitetura, para que seja mais genérica possível na implementação das facilidades oferecidas pelo modelo OSI de gerência.

Atualmente o OSIMIS está na versão 4.0 que permite a fácil integração entre sistemas (inclusive os proprietários) devido aos diferentes modelos e facilidades de gerência que apresenta. Nesta versão existe uma aplicação que permite a coexistência entre os modelos OSIMIS (CMIS/P) e *Internet* (SNMPv1). Originalmente, a plataforma foi designada para usar o ISODE (***ISO Development Environment***), mas já existem trabalhos para migrá-la para outras interfaces como a XOpen.

Os serviços e aplicações que esta plataforma disponibiliza são:

- Implementação completa dos serviços CMIS e do protocolo CMIP;
- Agente OSI que realiza todas as funções especificadas no modelo de gerência;
- Objetos padrões;
- Bibliotecas de classes C++ para determinados tipos de atributos, com os respectivos codificadores e decodificadores para as suas sintaxes;
- Uso de um objeto coordenador (*coordinator*) que gerência todo o processo de comunicação do sistema;
- Métodos genéricos de interação entre o objeto coordenador e os objetos gerenciados, através dos objetos chamados de fontes de conhecimento (*knowledge sources*);
- Compilador para a linguagem formal de especificação dos objetos de gerência OSI (*GDMO - Guidelines for the Definition of Management Objects*);
- Interface de alto nível para os desenvolvedores de aplicações gerentes (RMIB e SMIB);
- Mecanismo de transparência à localização de agentes, utilizando a implementação ISODE do serviço de diretório OSI;
- Aplicação genérica de passarela (*gateways*) entre os modelos de gerência OSI (CMIS/P) e *Internet* (SNMPv1).

Este modelo fornece uma implementação completa dos serviços e protocolos comuns do modelo OSI, permitindo ou usar a especificação completa do protocolo, ou uma versão mais leve deste. Fornece um suporte suplementar para o ISODE, que permite codificar, decodificar e analisar as cadeias de caracteres na especificação ASN.1.

Programar um gerente usando a interface do CMIS é tediosa. Então, o OSIMIS oferece uma interface de alto nível chamada RMIB, que oferece os seguintes serviços aos desenvolvedores:

- Estabelecimento e liberação de associações;
- Uso de nomes mais informais ao invés dos identificadores de objetos;
- Manipulação transparente de estruturas ASN.1;
- Listas de respostas (*linked lists*);
- Interface de alto nível para os relatórios de eventos;
- Tratamento de erros em diferentes níveis.

Na plataforma OSIMIS é oferecido suporte para organizar os processos dirigidos por eventos, para facilitar a integração com outros mecanismos de coordenação, como as interfaces gráficas dos usuários, que devem tratar dos eventos que ocorrem.

As aplicações que realizam o papel de gerenciadoras interagem com os agentes que cuidam de determinados objetos gerenciados, utilizando apenas o título e as classes dos objetos gerenciados. Isso é possível através da **transparência de localização** que neste caso, tem a função de identificar para a aplicação, quais são os agentes que possuem os objetos gerenciados associados a determinados recursos, assim como deve localizar este agente para a aplicação.

A transparência de localização é realizada através do serviço de diretório OSI que armazena as informações de forma hierárquica e distribuída, sendo ideal para o armazenamento das informações necessárias as aplicações gerentes, e para a localização dos agentes. Para armazenarmos as informações necessárias as aplicações gerentes, usamos o mecanismo DSA/DUA (*Directory Service Agent/Directory User Agent*) que permite as aplicações informarem a sua existência e os serviços que a aplicação disponibiliza aos seus usuários. Essas aplicações devem notificar ao DSA/DUA quando terminarem.

Esse mecanismo funciona da seguinte forma:

- Ao iniciar a operação, cada agente e cada aplicação gerenciadora se cadastram na árvore de informação de diretório (*DIT-Directory Information Tree*), como mostra o passo S na figura 4.8.
- Quando a aplicação de gerência deseja acessar um recurso, ela executa os seguintes passos:
    - Obter o nome do agente que gerencia a MIB que tem os objetos que representam o recurso requerido (passo 1);
    - Obter o endereço de apresentação em que o agente está em execução (passo 2).
- Associação com o agente que esta identificou (passo 3)

**Figura 4.8 - Mecanismo de Transparência de Localização**

Além destas facilidades, a OSIMIS apresenta algumas outras, como:

- Um compilador GDMO para construção de classes de objetos gerenciados;
- Mecanismo de transparência na localização de objetos gerenciados, através do uso do serviço de diretório X.500;
- Implementação do protocolo SNMP a nível de aplicação através do uso da coexistência entre os dois protocolos;
- Conjunto de aplicações genéricas e agentes específicos para a camada de transporte OSI e para a versão OSI da MIB da camada TCP/IP.

### 4.5.1 Coexistência dos protocolos CMIP e SNMP na gerência de redes

Devido ao protocolo padrão de gerência de redes mais utilizado ser o SNMP, a plataforma OSIMIS apresenta um modelo de conversão entre as duas plataformas, que deve prover mecanismos capazes de permitir a existência mútua destas duas plataformas em um ambiente de gerenciamento de redes. As principais restrições funcionais necessárias a esta coexistência são:

- Tradução da MIB SNMP em GDMO;
- Conversão das operações SNMP em operações CMIP;
- Redução do tráfego de informações de gerência decorrente do processo de "*polling*" do SNMP através do modelo OSI de notificações.

A coexistência é realizada através de uma aplicação de passarela, que tem a função de um agente OSI na camada superior e gerente SNMP na camada inferior. Usa as facilidades apresentadas pelo GMS (*Generic Management System*) para suportar todas as funções realizadas por um agente OSI, e usa os objetos especializados do GMS e os gerentes SNMP para realizarem juntos a conversão das informações de gerenciamento.

A técnica adotada para a coexistência nada mais é do que a definição de um conjunto genérico de regras de conversão entre os dois modelos, e o uso de um processo de aplicação capaz de operar sobre qualquer MIB. Para isso, é necessário um conversor de objetos SMI SNMP para objetos GDMO OSI, além de um compilador GDMO para compilar os objetos convertidos.

Tal esquema de coexistência pode ser visto na figura 4.9.

**Figura 4.9 - Arquitetura de passarela entre os protocolos CMIP e o SNMPv1**

### 4.5.2 Plataforma OSIMIS e orientação a eventos

Para implementar o mecanismo assíncrono da orientação a eventos, a plataforma OSIMIS apresenta dois objetos especiais: **o Coordenador (*coordinator*) e as Fontes de Conhecimento (*Knowledge Sources*)**, que fornecem abstrações especiais na implementação deste mecanismo.

O **Objeto Coordenador** é um centralizador dos eventos externos, além de tratar das chamadas dos temporizadores. Só existe uma instância deste objeto no sistema. Também é usado para implementar os mecanismos de iteração entre os objetos gerenciados e os recursos que representam, através de um comando CMIS GET. São possíveis três mecanismos:

- Acesso por demanda de aplicações gerentes;
- Acesso por *polling*;
- Acesso por eventos assíncronos.

Os **Objetos Fontes de Conhecimento** são usados para ativar, em tempo real, os objetos gerenciados quando ocorrerem eventos externos. São os pontos de comunicação externos onde ocorrem os eventos. Nas aplicações gerentes, obtêm-se com estes as informações sobre os recursos gerenciados. Também podem gerenciar o *polling* para acessar os objetos gerenciados. Ao contrário do objeto anterior, podemos ter várias instâncias deste objeto.

### 4.5.3 Sistema Genérico para gerenciamento

É implementado através de uma interface para o desenvolvimento de aplicações, que oculta totalmente os detalhes necessários para se usar os serviços oferecidos pelo CMIS para endereçar objetos, o escopo dos níveis de hierarquia, a filtragem dos atributos dos objetos, e as correções de erro. A facilidade para o desenvolvimento de aplicações apresentada pela interface é devido ao uso de duas classes de objetos: ***MO e MOClassInfo***.

A **classe MO** é a classe raiz da hierarquia das classes de objetos gerenciados, e possui métodos e atributos que automatizam os acessos as informações usando o protocolo CMIP, além de conter informações sobre as posições dos objetos gerenciados na árvore de informação de gerenciamento (MIT).

A **classe MOClassInfo** apresenta informações comuns a todos os objetos pertencentes a uma classe, permitindo criar uma instância deste objeto através da alocação dinâmica, além de poder definirmos valores default para alguns dos atributos deste objeto.

### 4.5.4 Interfaces para construção de processos gerentes

São APIs (*Aplication Programs Interfaces*) amigáveis para o acesso eficiente a objetos MIB remotos, o que facilita a construção de processos gerentes. São interfaces implementadas em OSIMIS, portanto orientadas a objetos, para acesso aos serviços CMIS. Usam-se dois enfoques distintos, explicados abaixo:

- A ***Remote MIB (RMIB)*** apresenta uma abstração no acesso a MIBs Remotas, através do uso da noção de um **objeto de associação**, que é usado para encapsular uma operação de gerenciamento com um objeto remoto, obscurecendo então, os serviços usados do protocolo CMIP assim como o acesso a MIT remota.
- A ***Shadow MIB (SMIB)*** abstrai os objetos gerenciados num espaço de endereçamento local, possibilitando a total transparência do uso do protocolo de gerenciamento. Parâmetros de gerenciamento podem ser substituídos por ponteiros.

---

# 5. Distribuição da Gerência na Rede

Com o crescimento das redes de computadores, em tamanho e complexidade, sistemas de gerência baseados em um único gerente responsável por todas as funções de gerenciamento são inapropriados, devido ao volume das informações que devem ser tratadas e que podem pertencer a localizações geograficamente distantes do gerente.

Desta forma, evidencia-se a necessidade da distribuição da gerência na rede, através da divisão das responsabilidades gerenciais entre gerentes locais que controlem domínios distintos e da expansão das funcionalidades dos agentes. Cada gerente local de um domínio pode prover acesso a um gerente responsável (pessoa que interage com o sistema de gerenciamento) local e/ou ser automatizado para executar funções delegadas por um gerente de mais alto nível, geralmente denominado de **Centro de Operações da Rede** (***NOC - Network Operation Center***). O NOC é responsável por gerenciar os aspectos inter-domínios, tal como um enlace que envolva vários domínios, ou aspectos específicos de um domínio, devido à inexistência de gerente local.

Os tipos mais básicos de tarefas de gerenciamento de uma rede são: monitoração e controle. A monitoração consiste na observação periódica de objetos gerenciados importantes para a política de gerenciamento. A partir da monitoração, o gerente tem conhecimento do estado da rede e, desta forma, pode efetuar operações de controle sobre a mesma. A distribuição das funções de monitoração é mais preemente em relação as funções de controle, pois a monitoração consome mais recursos da rede, bem como a atenção do gerente, pois através dela é que se obtém o estado da rede em relação ao tempo, enquanto que as funções de controle são invocadas em menor número, geralmente com objetivos de alteração de configuração e erradicação de problemas.

Os modelos de gerência diferenciam-se nos aspectos organizacionais envolvendo a disposição dos gerentes na rede, bem como no grau da distribuição das funções de gerência.

## 5.1 Modelo Internet

O modelo inicial de gerência Internet concentra as funções de controle e monitoração em um único gerente responsável pelo acesso aos diversos agentes da rede. Os agentes são simples fornecedores das variáveis da MIB,

enquanto que o gerente, através do mecanismo de *polling*, monitora a rede, efetuando, quando necessário, operações de controle. Não é definido nenhum mecanismo para a comunicação entre gerentes. Tal abordagem objetiva a simplificação dos agentes, permitindo o rápido desenvolvimento destes e a minimização dos recursos usados nos elementos de rede. Contudo o gerente é sobrecarregado com todas as funções, gerando grande tráfego na rede e degradando o tempo de resposta aos eventos da rede.

Com intuito de prover a monitoração remota em um ambiente de gerenciamento Internet, foi definida a MIB **RMON** (***Remote Network Monitoring***). Tal MIB permite que as funções de monitoração sejam realizadas através da captura dos pacotes que transitam por uma sub-rede (por enquanto, do tipo ethernet) sem a interferência constante do gerente. A RMON é composta por nove grupos: Statistics, History, Alarm, Host, HostTopN, Matrix, Filter, Packet Capture e Event. O grupo Statistics mantêm estatísticas das interfaces do agente, por exemplo, o número de colisões. History armazena amostras de informações colhidas no grupo Statistics. O grupo Alarm fornece mecanismos usados para a monitoração de variáveis de gerenciamento do tipo Integer, com valores-limites configurados que podem disparar eventos ao serem atingidos pelo valor monitorado. O Host contêm informações referentes aos nodos da sub-rede, como o número de pacotes enviados por cada nodo. O grupo HostTopN classifica as informações obtidas pelo grupo Host, gerando, por exemplo, os nodos que mais transmitiram pacotes. O Matrix possui informações referente a comunicação entre dois nodos da sub-rede. O Filter provê mecanismos de filtros para os pacotes recebidos da sub-rede, que podem disparar um evento ou um processo de armazenamento de pacotes. O Event controla a geração e notificação dos eventos definidos, por exemplo, um relativo a um alarme especificado no grupo Alarm.

Com o crescimento da rede Internet, foi proposta uma adaptação do modelo de gerência original baseado no protocolo SNMP. Tal proposta, denominada de SNMP 2.0, aumenta as funcionalidades dos agentes, através da flexibilização na geração de notificações assíncronas e da capacidade de um processo assumir ambas as funcionalidades de gerente e agente, permitindo a comunicação entre gerentes de níveis diferentes.

O SNMP 2.0 adiciona dois tipos novos de operações no protocolo, ***GetBulkRequest*** e ***InformRequest***, que permitem um aumento das funcionalidades dos agentes e gerentes intermediários. A GetBulkRequest otimiza a recuperação de um volume considerável de variáveis, principalmente em relação à recuperação de entradas de tabelas. Por exemplo, para a recuperação de 10 entradas de uma tabela é necessário o envio de 10 operações GetNextRequest no SNMP original, e com a versão 2.0, somente um GetBulkRequest é suficiente. A InformRequest permite um gerente enviar de forma assíncrona uma notificação de algum evento, sendo análogo ao TRAP, contudo é um serviço confirmado.

Dentro do contexto do SNMP 2.0 foi definido uma MIB, denominada de M2M, que suporta a distribuição de funções de monitoração entre os gerentes da rede. Tal monitoração é baseada em amostras realizadas em variáveis do tipo COUNTER, GAUGE e TIMETICKS de agentes. Os valores de tais atributos são comparados com valores-limites configurados, e caso sejam atingidos, um InformRequest ou um TRAP é enviado pelo gerente que implementa a MIB M2M a outro gerente.

A MIB é especificada a partir dos conceitos de alarme, evento e notificação. O alarme é uma condição configurada que é verificada periodicamente. Se um alarme for detectado, é disparado o evento associado, que por sua vez, pode gerar uma notificação para um gerente especificado. Tal MIB é análoga aos grupos Alarm e Event da RMON.

# 5.2 Modelo OSI

O modelo OSI possibilita a delegação das funções de monitoração aos agentes, através da definição de um ambiente orientado a objetos que incorpora tais procedimentos e da abordagem orientada a notificações assíncronas do modelo. Contudo as funções de controle ainda ficam relegadas ao gerente, pois o conhecimento relativo à tomada de decisões gerenciais não se adapta para ser codificado em classes de objeto, ao contrário do conhecimento referente à monitoração, que é mais simples, geralmente estático e periódico.

Tal modelo gera agentes mais complexos de serem desenvolvidos, consumindo mais recursos dos elementos de rede, enquanto que economiza o uso da rede, devido a minimização dos pedidos de informações (*pollings*) necessários para obter dados sobre os objetos gerenciados, livrando o gerente para tarefas mais “inteligentes”.

Além da definição de um agente mais funcional, o modelo OSI introduz o conceito de hierarquia de gerentes, através da possibilidade de um mesmo processo de aplicação funcionar como gerente e agente, sendo denominado de gerente intermediário. Desta forma é possível ao NOC delegar tarefas para gerentes intermediários responsáveis por certos domínios da rede. A comunicação entre gerentes é realizada pelo acesso a objetos da MIB que possuem as

informações que devem ser compartilhadas ou funcionalidades que devem ser delegadas entre os gerentes.

As funcionalidades que podem ser delegadas a um agente ou a um gerente intermediário são exemplificadas nas funções de gerenciamento definidas pelo ISO, que fornecem serviços padronizados para as cinco áreas funcionais de gerenciamento OSI: configuração da rede, falhas dos componentes, níveis de desempenho, segurança de acesso e contabilização do uso dos recursos. Tais funções são relativas, principalmente, à monitoração remota, como a monitoração de um atributo de um objeto que representa a utilização de um recurso, ou ao controle de algum mecanismo necessário para o gerenciamento, como o registro de logs.

Abaixo são exemplificadas algumas das funções de gerenciamento:

Æ **Função de gerenciamento de objeto**: especifica três tipos de notificações relativas à criação de objeto, remoção de objeto e mudança de valor de atributo. Nesse caso o gerente não precisaria ficar realizando *polings* para a verificação de tais condições, e sim, o objeto emitiria as notificações. Para tal, a classe do objeto monitorado deve importar as definições das notificações. Em tempo de execução, se o gerente quiser receber somente determinados tipos de notificações, por exemplo relativas à criação de objetos, pode usar o objeto *eventForwardingDiscriminator* definido na função de controle de relatórios, que repassa somente tipos de notificações configurados no objeto;

Æ **Função de relatório de alarmes**: define notificações genéricas de alarme referentes à falhas, fornecendo informações tais como o tipo da falha, causa provável e índice de gravidade. Os tipos de falhas variam desde problemas de comunicação, como perda de sinal, até alarmes cobrindo problemas ambientais, como umidade alta. Tal função possibilita grande autonomia aos elementos gerenciados, que notificarão de forma assíncrona o gerente na ocorrência de uma falha. Nesse caso, o gerente somente precisará verificar periodicamente a conectividade com o agente, e iniciar procedimentos de recuperação de falhas no recebimento de uma notificação. As classes de objetos que necessitam se auto-gerenciar, devem importar a definição padrão da notificação de alarme. Também é definida uma estrutura de registro relativo ao gerenciamento. O mecanismo de registro de informações em um log é definido na função de controle de log, que define um objeto que seleciona as notificações que devem ser armazenadas localmente sobre forma de um objeto que representa um registro de log.

Æ **Função de monitoração de carga de trabalho**: define objetos métricos que realizam a monitoração de atributos de outros objetos dos tipos COUTER e GAUGE, que podem representar a utilização de um recurso, a taxa de requisição de um recurso ou a taxa de rejeição de acesso a um recurso. O objeto métrico pode enviar notificações quando o atributo monitorado atinge valores-limite pré-determinados. Além da análise pura do valor amostrado, são definidos também objetos que usam a média e a variância dos valores do atributo monitorado para a comparação com os valores-limite. Tal função permite a análise de desempenho de recursos gerenciados, sem a necessidade de consultas periódicas por parte do gerente.

# 5.3 Gerência via Servidores Elásticos

A abordagem de gerência via servidores elásticos é realizada através da distribuição de programas independentes que encapsulam funções de controle e monitoração de objetos gerenciados.

O termo elástico se refere a capacidade do servidor de alterar dinamicamente a sua funcionalidade, através da execução e remoção dos programas delegados. Um servidor elástico pode atuar como um agente ou como um gerente intermediário, sendo que cada programa delegado pode ser armazenado até que o gerente responsável invoque um comando de execução. As instâncias dos programas podem ser suspensas, reiniciadas ou finalizadas. Tais funções são realizadas através do protocolo de delegação.

Uma instância de um programa de gerenciamento pode se comunicar com o gerente criador, com outros programas, invocar funções de bibliotecas disponíveis no ambiente e acessar os objetos gerenciados. O acesso aos objetos gerenciados são realizados pelos Pontos de Controle de Observação, que representam uma interface genérica de acesso, desta forma escondendo os detalhes de implementação. Tal interface pode acessar diretamente os recursos gerenciados, possibilitar uma conversão de protocolo, por exemplo para o acesso à redes CMIP ou SNMP, e ainda possibilitar o acesso a outros servidores elásticos, através do protocolo de delegação. No último caso, evidencia-se a capacidade da definição de um gerenciamento composto por vários gerentes.

Abaixo é mostrado um trecho de um programa de gerenciamento codificado em um subconjunto da linguagem C referente à monitoração de um enlace de comunicação e procedimentos de tratamento de possíveis erros:

```
if ((link.control.stat >  normal.start) and (link.q.length > normal.q))
{
        link.test( );
        if (link.failure)
        {
                recover (link.failure.type);
                notify(manager, link.failure.params);
        }
}
```

A primeira condição testada representa uma indicação de problemas de um enlace. Caso esse seja detectado, é invocado um teste. Se o resultado desse apresentar algum erro, é disparado um procedimento que tenta solucionar o problema, sendo após enviado ao gerente os parâmetros que representam os resultados do procedimento de recuperação.

No caso do uso de um servidor elástico, tais procedimentos são executados de forma autônoma em relação ao gerente. Se o modelo de gerência OSI fosse usado, as duas condições que representam o indício de problemas, deveriam ser periodicamente requisitadas através do serviço M-GET. Ou o agente poderia ser configurado para enviar em tais condições uma notificação, contudo seria necessário uma para cada condição, pois o modelo OSI não permite notificações compostas. Para executar o teste, seria necessário a invocação de um M- ACTION. Após, para verificação do resultado do teste, outro M-GET necessitaria ser enviado. Caso a função de recuperação de problemas ser necessária, o gerente invocaria a ação de recuperação de problemas com um M-ACTION e, por fim, outro M-GET seria enviado para recuperar os resultados.

Tal exemplo demostra que a abordagem via servidor elástico minimiza o tráfego na rede, bem como toma ações de forma mais rápida em relação aos eventos da rede. O servidor elástico é configurado somente para as funções realmente necessárias para cada momento dentro da política de gerenciamento da rede, ao contrário de agentes OSI ou Internet que podem permanecer atualizando objetos da MIB que não estão sendo usados.

---

# 6. Arquitetura de Segurança para Gerência de Redes

Esta seção apresenta um levantamento dos riscos de segurança associados à Sistemas de Gerência de Redes e descreve uma Arquitetura de Segurança aplicável a tais sistemas, garantindo a autenticação, integridade e confiabilidade nas comunicações entre as entidades de gerência.

Protocolos de Gerência de Redes e os canais de comunicação que transportam informação de gerência são potencialmente vulneráveis a atentados contra a segurança. Cuidados particulares devem, portanto, ser tomados para assegurar que tais protocolos e informações estejam protegidos. A definição de vulnerabilidade e dos riscos de segurança dos Sistemas de Gerência e a criação de ferramentas para tratar estes problemas fazem parte do conjunto de ações fundamentais para o funcionamento confiável das redes. A especificação de ferramentas, seu comportamento e seus inter-relacionamentos compõem uma arquitetura de segurança.

## 6.1 Segurança em Redes de Computadores

Existem diversos elementos sobre os quais podem incidir ameaças contra a segurança em um ambiente informatizado e, em especial, em um ambiente interligado por redes de computadores. Segurança em informática pode ser compreendido como a garantia ou confiança que os usuários tem em determinado sistema. Segurança aplicada no domínio das Redes de Computadores, então, deve garantir que o sistema não seja comprometido por ameaças cuja origem não esteja localizada, necessariamente, no computador local mas remotamente.

Existem muitas formas de comprometer sistemas porque normalmente existem muitos pontos expostos. Estes pontos de exposição podem ser classificados conforme estas seis categorias: hardware, software, informação/dados, pessoal, documentação e suprimentos.

No jargão de segurança, os itens que se enquadram nas categorias acima são chamados ***ativos***. São os ativos de uma instalação que devem ser protegidos de ameaças porque é o comportamento apropriado deste ativos que vai permitir o funcionamento dos sistemas. Uma alteração, destruição, erro ou indisponibilidade de algum destes ativos pode gerar um comprometimento do sistema.

Analisando os ativos apresentados no escopo de segurança em redes de computadores, apenas Hardware, Software e Informação/Dados são passíveis de serem protegidos por meios do que se convenciona chamar de ***Segurança Lógica***, em contraposição à ***Segurança Física***, onde esta última é a segurança tradicional de informática, centrada em restrições de acesso físico às instalações e equipamentos, prevenção de acidentes de trabalho e planos de recuperação de desastres como incêndios e inundações. Já a Segurança Lógica lança mão de software para garantir quatro princípios básicos: ***autenticação de usuário***, ***disponibilidade de recursos***, ***integridade das informações*** e ***confidencialidade das informações***.

### 6.1.1 Agressões e Falhas

Outra forma de analisar os problemas de segurança é fazer uma classificação das ameaças entre ***agressões*** e ***falhas***. Falhas são acontecimentos acidentais que, de uma forma ou de outra, põem em risco a segurança das instalações e dos sistemas porque atentam contra a confiabilidade e/ou disponibilidade de um sistema. Exemplos de falhas são: inundações, incêndios, terremotos, roedores que atacam a fiação e provocam curtos-circuitos, acidentes ou erros humanos (derramamento de líquidos sobre equipamento, manipulação incorreta de equipamento, digitação de dados incorretos, ...) e falhas de hardware, de software e de comunicação. As agressões, por outro lado, são intencionais e hostis. São exemplos de agressões as ameaças de bombas, roubo, operação inadequada proposital de equipamentos, software propositadamente incorreto, vírus, invasões através de redes, (tentativas de) acesso a informações confidencias, etc.

Não é possível abordar todos os problemas a partir do enfoque de Segurança Lógica. Os problemas relacionados às falhas dizem respeito à outros aspectos que não seja Segurança Lógica (manutenção e segurança física, por exemplo). Na lista de agressões também se encontram diversas ameaças que não são tratáveis por medidas de Segurança Lógica, como ameaças de bomba e roubo.

Então, sob esta ótica, as ameaças que dizem respeito à segurança em redes de computadores são agressões efetuadas por pessoas não autorizadas (as quais serão chamadas de ***invasores***) que objetivam obter benefícios indevidos ou prejudicar o funcionamento dos sistemas.

### 6.1.2 Acesso à Informação e à Capacidade de Processamento

O que está em "disputa" neste contexto de proteção de Sistemas em Rede pode ser resumido nestes dois itens:

- Æ A ***informação*** em si: o acesso, a destruição e a modificação de informação e o acesso a serviços; e
- Æ O acesso à capacidade de ***processamento*** de informação e ao equipamento: roubo de ciclos de máquina, acesso à serviços, redes e software, e uso da capacidade de armazenamento.

Por informação deseja-se representar muitas coisas: dados para processamento, tecnologia, know-how, conhecimento científico, informações econômico-financeiras, estratégicas e políticas, projetos, etc. Computadores podem manter informações confidenciais sobre pessoas, sobre objetivos militares, informações vitais para empresas ou governos, saldos bancários, e assim por diante. O valor destas e de outras informações é alto, apesar de ser muito difícil, na maioria dos casos estabelecer o valor intrínseco de determinada informação.

Desta forma, a capacidade de acesso a informação, bem como a capacidade de alterá-la ou destruí-la, representa então poder, que é protegido pelos legítimos detentores da mesma e que é buscado (de forma ilegítima) pelos invasores.

O acesso ao hardware e ao software (e o decorrente acesso à capacidade de processamento) também representa poder, já que a utilização dos mesmos permite o processamento de informações. O acesso ilegítimo à capacidade de processamento pode ser apenas roubo de tempo de processamento, mas esse tipo de ato pode levar a conseqüências sérias, como o aumento de custo para usuários legítimos ou, em caso extremo, a negação de serviço para o usuários legítimos, uma vez que a cpu e/ou memória estão ocupadas realizando tarefas estranhas à instalação. Um exemplo típico de roubo de tempo de cpu é a utilização de máquinas para decifrar informações criptografadas (como arquivos de senhas) para ter acesso a novas informações: uma agressão (roubo de ciclos) que alimentará outra agressão (a invasão de outros sistemas).

As agressões referentes à informação e à capacidade de processamento podem ser executadas basicamente de três formas: por escuta ou monitoração da rede, por invasão ao sistema e por mascaramento.

A **escuta/monitoração** do canal é tarefa simples e mesmo com recursos pouco sofisticados é possível alcançar tal feito. Exemplos de formas de se conseguir monitoração de redes vão desde o uso de analisadores de protocolos (recurso caro) até

a modificação do software de um computador comum para atuar como escuta. Existem ainda equipamentos próprios para escuta (passiva) que se valem de emanações eletromagnéticas dos cabos e conectores. O comprometimento da segurança de um nó intermediário em redes ***store-and-forward*** ou de ***gateways*** e roteadores leva à exposição de informações a terceiros. Então, toda informação que circula pelas redes pode ser interceptada e, se medidas de segurança não tiverem sido adotadas, esta informação se torna não confidencial. Pouco se pode fazer a nível de software para impedir este tipo de agressão. O uso de criptografia deve ser considerado pois, apesar de não impedir o ataque, é uma forma de reforçar o sigilo das comunicações.

A **invasão** de sistemas com o objetivo de ganhar acesso a informações e a recursos computacionais é uma das agressões mais comuns. As vulnerabilidades relativas à invasão de sistemas podem ser geradas de muitas formas; por exemplo pela não instalação de senhas por parte de usuários ou por falhas de implementação de softwares. Os riscos associados são os mesmos relacionados para a escuta do canal e mais a possibilidade de negação de serviços para usuários legítimos em função dos invasores estarem usufruindo de serviços de forma não autorizada. Grande parte desta vulnerabilidade é responsabilidade do sistema operacional hospedeiro, reduzindo a responsabilidade dos mecanismos de segurança das redes. Na verdade, esta forma de agressão normalmente se transforma ou em escuta ou em mascaramento após concretizada a invasão.

O terceiro item, **mascaramento**, consiste na tentativa de personificação de uma terceira entidade em uma comunicação. O objetivo é o mesmo que os anteriores: acesso a informações ou a recursos computacionais. O mascaramento pode ser conseguido por invasão simples (como visto acima) ou por meios muito mais sofisticados como a alteração de pacotes que fluem na rede ou ainda forjando pacotes. Estando mascarado de uma entidade comunicante legítima da rede, o software "clandestino" pode ter acesso às informações sensíveis ou a recursos importantes e até provocar eventos anonimamente. Fica claro que este tipo de agressão é complexa o que pressupõe a necessidade do invasor ser conhecedor profundo dos protocolos de comunicação utilizados.

# 6.2 Gerência de Redes e Segurança

Gerência de Redes é uma aplicação distribuída onde processos de gerência (agentes e gerentes) trocam informações com o objetivo de monitorar e controlar a rede. O processo gerente envia solicitação ao processo agente que por sua vez responde às solicitações e também transmite notificações referentes aos objetos gerenciados que residem em uma base de informação de gerenciamento (MIB).

Toda e qualquer informação produzida pelo Sistema de Gerência, em um determinado instante, está ou em uma MIB ou trafegando pela rede (em uma comunicação típica entre um agente e um gerente ou entre dois gerentes) ou ainda poderá ser deduzida (reproduzida) com informações parciais oriundas destas duas fontes. Toda informação produzida pelo Sistema de gerência é útil para a manutenção da rede em operação com confiabilidade. Sem dúvida, os Sistemas de Gerência facilitam a administração das redes seja pela automatização de algumas atividades, seja por permitir maior controle sobre os recursos da rede ou ainda por fornecer informações (estatísticas, por exemplo) que permitirão ajustes, correções ou adaptações às necessidades dos usuários.

Entretanto, neste ponto também é possível observar que o próprio Sistema de Gerência e as informações por ele geradas são de extrema valia para indicar pontos vulneráveis à ataques, ter acesso e controlar indevidamente recursos da rede, manipular informações, em suma, realizar atividades prejudiciais à rede, aos sistemas e/ou aos usuários.

Sob certa ótica, é possível até afirmar que uma rede com Sistema de Gerência formal implantado é ***menos*** segura do que a mesma rede sem o Sistema de Gerência. Os exemplos a seguir explicam esta afirmação:

- Æ Se um agente emite um alarme acerca de uma falha em um mecanismo de segurança e este alarme é interceptado por um invasor; então está-se fornecendo uma informação valiosa para que um intruso possa realizar outras agressões.
- Æ Uma notificação de alarme forjada por um intruso pode levar a alguma ação (por parte do gerente "iludido") que libera informações ou serviços a usuários que não teriam autorização em situações normais.
- Æ Uma entidade infiltrada que se mascara de gerente pode ter acesso à informações sensíveis mantidas

na MIB, inclusive com poder de alteração (como desativação de serviços de segurança ou alteração de registros de contabilização).
- Æ Um agente mascarado pode fornecer acesso à recursos da rede para usuários não autorizados e/ou indisponibilizar tais recursos para usuários legítimos; ou ainda forjar informações com o intuito de forçar o gerente para a alocação de mais ou melhores recursos.

Com estas e muitas outras vulnerabilidades é que pode-se concluir que um Sistema de Gerência de Redes torna a rede mais insegura por um lado, ao mesmo tempo que cria mecanismos de controle que serão úteis também na manutenção da segurança da rede.

## 6.2.1 Ameaças sobre Sistemas de Gerência

As ameaças que serão abordadas dizem respeito às agressões que podem ser executadas por intrusos na rede ou por usuários que tentam obter mais recursos ou informações do que são autorizados. Podem ser classificadas em: Mascaramento, Monitoração ou Escuta Passiva e Escuta Ativa;

Æ **Mascaramento:**

É a pretensão de uma entidade de se fazer passar por outra de modo a ter acesso a informações, ganhar novos privilégios, afetar os sistemas, etc.

Um fato é certo: para criar uma entidade mascarada, o agressor deve ter acesso à rede, podendo ser um acesso autorizado (lícito) ou não. Então, uma primeira barreira contra este tipo de agressão é um Sistema de Controle de Acesso à rede o mais confiável possível. Controle de Acesso envolve identificação, autenticação e autorização, além de uma política de segurança e consciência por parte dos usuários da importância da segurança para a rede e seus sistemas. Por **identificação** entende-se uma estrutura de nomes que garanta a identificação única para cada entidade da rede. Mas não basta identificação porque as entidades podem não ser confiáveis ao se identificar, sendo necessário então a confirmação da entidade: **autenticação**, ou seja, a validação de que uma entidade é quem ou aquilo que diz ser. A **autorização** permite indicar se determinada entidade (identificada e autenticada) possui acesso legítimo (autorizado) a determinado recurso ou operação e deve evitar acesso caso contrário.

Mas não se pode pensar em Controle de Acesso somente no momento do primeiro acesso em uma sessão (***login***) mas também em outras atividades durante a sessão, de forma continuada, sob pena de abordar o problema de maneira muito pobre.

É então importante para a segurança em um Sistema de Gerência que cada entidade componente do mesmo esteja devidamente identificada e autenticada e tenha os direitos de acesso definidos e controlados. Para tanto, faz-se necessária a especificação e implantação de serviços de Identificação/Autenticação específicos para entidades comunicantes, e de Confidencialidade de Acesso aos recursos (no caso, o acesso à MIB).

Æ **Monitoração ou Escuta Passiva:**

Neste caso, há apenas coleta de informações que transitam na rede. Apesar de, em um primeiro momento, os riscos que representa a escuta passiva parecerem pequenos, é possível recolher muitas informações úteis para o comprometimento de uma rede. Alguns exemplos são as informações que dizem respeito à segurança da rede ou sobre falhas, que fluem entre agentes e gerentes da rede. Estas informações podem ser senhas de usuários, informações trocadas entre entidades para autenticação, informações sobre configuração, informações sobre falha de algum mecanismo de segurança, etc.

Quando informações sensíveis como as citadas devem transitar pela rede, é fundamental que sejam adotadas medidas para evitar que tais informações sejam acessadas indevidamente. Para tanto é necessário definir um Serviço de Confidencialidade de Comunicação.

Æ **Escuta Ativa:**

A escuta ativa difere da escuta passiva por não apenas coletar informações que fluem pela rede, mas também por alterá-las de alguma forma, seja no conteúdo, na seqüência, no tempo ou pela destruição ou criação de mensagens; de forma a realizar ou induzir ações não autorizadas ou criar condições para ações não autorizadas ou ainda encobrir atos ilícitos praticados.

Duas medidas de proteção se tornam então necessárias: autenticação da origem das mensagens e garantia da integridade das mensagens. Sem estes dois serviços a rede continuará aberta a ataques.

Os Sistemas de Gerência de Redes estão sujeitos a todas estas ameaças porque estão baseadas na separação das funções de gerência com distribuição das informações, sendo necessária a comunicação entre entidades de gerência. Deve-se então acrescentar ao Serviço de Identificação/Autenticação de entidades a tarefa de autenticar a origem, a forma e o momento do envio das mensagens. Além disso, um Serviço de Integridade de mensagens deve ser estabelecido e será o responsável pela garantia de que uma mensagem não sofreu alterações em seu caminho desde a origem até o destinatário, envolvendo as tarefas de evitar alteração de informações, re-seqüenciamento e a simples destruição.

A autenticação, por sua vez, também depende da integridade das mensagens para algumas tarefas. Por exemplo, de nada adianta validar a origem de uma mensagem que foi alterada por uma escuta ativa ou se durante uma autenticação de entidade as mensagens podem ser afetadas de modo a validar uma entidade mascarada.

## 6.2.2 Requisitos de Proteção

No contexto de um Sistema de Gerência, as ameaças que cabem ser analisadas, dentre todas as ameaças à segurança em uma rede de computadores, são as seguintes:

a) acesso não autorizado à informação de gerência que flui pela rede;

b) acesso não autorizado à informação de gerência mantida na MIB;

c) alteração e re-sequenciamento de mensagem de gerenciamento; e

d) geração de mensagens de gerenciamento por terceiros (entidades que não fazem parte da arquitetura de segurança).

Os Serviços de Segurança que precisam estar disponíveis para contrapor estas ameaças, conforme visto anteriormente são:

- Æ Confidencialidade (contra (a) e (b));
- Æ Integridade (contra (c)); e
- Æ Autenticação (contra (d)).

Para suportar o ***Serviço de Confidencialidade*** (ou privacidade) é necessário o uso de criptografia. Há dois tipos básicos de criptografia em uso nos dias atuais: por chave secreta e por chave pública. A segunda alternativa - chave pública - é a mais usada devido a eficiência do processo da distribuição das chaves. Um sistema de chaves públicas prevê a existência de duas chaves simétricas: o que uma chave cifra e seu par decifra e vice-versa. Uma das chaves é mantida em segredo (chave privada) e a outra é divulgada (chave pública - daí o nome do sistema) através de um serviço de diretório por exemplo. O uso de criptografia é a única forma de garantir que uma mensagem que esteja trafegando pela rede e seja interceptada não forneça informações valiosas para o agressor. A mensagem poderá ser interceptada, mas dificilmente será decodificada.

Da mesma forma o acesso a MIB pode ser aberto pois, se as informações lá contidas estiverem cifradas, elas não serão úteis para invasores, uma vez que estes não terão tempo hábil para decifrá-las antes que ocorram alterações nas mesmas. Assim, um grau de segurança adicional é conseguido com o uso de criptografia sobre a MIB. Empregando o conceito de que a MIB somente poderá ser acessada por um único agente, toda a informação poderá ser guardada criptografada com a chave pública deste agente. Desta forma, só o mesmo agente pode ter acesso às informações geradas ou manipuladas por ele próprio.

Para suportar o ***Serviço de Integridade***, duas providências se fazem necessárias:

Æ Para evitar que uma mensagem alterada seja considerada válida, a ação a ser tomada é a sifragem de um campo que contenha o ***checksum*** de toda a mensagem. A chave que deve ser utilizada para isto é a chave privada do remetente da mensagem (a chave privada do esquema de chave pública). Com isto se garante a integridade da mensagem, pois um agressor não terá como alterar a mensagem, gerar um novo ***checksum*** e criptografá-lo pois não possuirá a chave correta. Já o destinatário pode verificar a integridade simplesmente usando a chave pública do remetente para conferir o ***checksum*** calculado com o decifrado. Qualquer alteração da mensagem é imediatamente

detectada.

Æ Para evitar o re-sequenciamento, o que deve ser feito é a inclusão de um campo que indicará a ordem da seqüência da mensagem. Este campo deverá conter um valor dentro de uma seqüência determinada a cada comunicação entre cada par de entidades (ou seja, a cada mensagem, o remetente incluirá o valor da seqüência e indicará qual valor deverá ser usado na próxima comunicação entre estas duas entidades). Estes dois campos também devem ser criptografados com a chave privada do remetente.

O ***Serviço de Autenticação*** deve garantir que a origem das mensagens de gerenciamento são de entidades legítimas para evitar a execuções de ações indevidas ou o acesso a informações por terceiros. Uma observação à providência referente a alteração de mensagens citada acima pode levar a conclusão de que a própria integridade oferece meios de aferir a autencidade das mensagens, uma vez que somente o interlocutor autêntico conhecerá sua chave privada e com isto poderá gerar os campos criptografados de acordo com o esperado. A autenticação, então, está automaticamente incluída no Serviço de Integridade.

# 6.3 Arquitetura de Segurança para Gerência de Redes

O que será apresentado a seguir é uma Arquitetura de Segurança seguida de seus algorítimos utilizados para incorporar segurança em um Sistema de Gerência de Redes genérico.

## 6.3.1 Modelo da Arquitetura de Segurança

Cada agente e gerente do Sistema de Gerência deve possuir uma Interface de Segurança que deve garantir que as mensagens recebidas pelos agentes e gerentes são realmente internas ao Sistema (autênticas) e que não foram alteradas (íntegras). Além disso, pode ser desejável a confidencialidade da comunicação entre as entidades do Sistema e esta característica também deve ser garantida pela Interface de Segurança. Esta interface atuará como uma "clearing house" entre cada par comunicante, impedindo que informações de gerência sejam acessadas, alteradas ao forjadas por entidades não autorizadas.

Os serviços diretamente implementados pela Interface de Segurança são:

- Æ **Serviço de Autenticação:** garantindo a origem autêntica das mensagens;
- Æ **Serviço de Integridade:** que impede o processamento de mensagens adulteradas ou forjadas;
- Æ **Serviço de Confidencialidade de Comunicação:** tornando as mensagens não acessíveis por terceiros, enquanto úteis;
- Æ **Serviço de Confidencialidade de Acesso:** que garante a proteção às informações de gerência mantidas na MIB;

Todos os Serviços acima devem estar presentes nas Interfaces de Segurança dos agentes e dos gerentes componentes do Sistema de Gerência, à exceção do último, cuja presença somente é necessária nas entidades agente, pois diz respeito apenas a atividades destes.

Acessoriamente, são facilmente conseguidos como "efeito-colateral" da implantação dos serviços acima os seguintes:

- Æ **Serviço de Controle de Acesso:** em função da Confidencialidade de Acesso. Se apenas o proprietário da MIB pode "compreender" os dados lá mantidos, o problema de acesso está resolvido;
- Æ **Serviço de Não-Repudiação:** resultante da implantação do Serviço de Autenticação. Uma vez garantida a origem da mensagem, também não há como o remetente negar a autoria da mesma pois somente ele poderia gerar uma mensagem com campos criptografados pela chave privada dele.

Além dos serviços citados, é de grande importância que as Interfaces de Segurança mantenham registros de ocorrências em ***logs*** para que seja possível a realização de auditorias, como atividade de gerenciamento de segurança.

Interface de Segurança é, portanto, uma redoma que encapsula totalmente cada agente e cada gerente do Sistema de Gerência, de forma que toda comunicação entre estas entidades se dê somente através da Interface. Esta abordagem permite que a instalação da Interface seja transparentes para os agentes e gerentes, ou seja, nada é alterado nos agentes e gerentes para que a Arquitetura de Segurança seja implantada. As comunicações entre entidades sempre passarão por filtros (as Interfaces de Segurança) em cada um dos lados desta comunicação, para verificação de

integridade e autoria e para garantir privacidade.

A figura abaixo mostra a Interface de Segurança.

**Figura 6.1 - Interface de Segurança**

Somente trafegarão na rede (no escopo de Sistemas de Gerência) pacotes de comunicação entre Interfaces de Segurança que encapsulam pacotes de agentes e gerentes, com todos os mecanismos de segurança para evitar possíveis agressões. A interface de segurança emissora é responsável pela incorporação dos mecanismos no pacote original e a interface do lado receptor é responsável pelas verificações e liberação ou não de pacotes.

O acesso às informações de gerência guardadas na MIB também deve ser restrito. Existem duas formas de prover tal restrição: pela instalação de um mecanismo de controle de acesso próprio ou a criação de um mecanismo de confidencialidade, onde as informações armazenadas na MIB estariam sifradas. A estratégia de estabelecer um Serviço de Confidencialidade no acesso à MIB, garantindo que um e somente um agente (o seu criador e mantenedor) terá acesso direto às informações lá contidas, se apresenta como a mais interessante. Três ítens motivam esta escolha:

Æ O mecanismo de confidencialidade já está disponível para outros serviços de segurança (Confidencialidade de Comunicação), eliminando a necessidade de construção de um novo mecanismo de segurança, o controle de acesso;

Æ Elimina a necessidade de criação de uma Interface de Segurança também para a MIB, centralizando nos agentes a implementação dos mecanismos de segurança;

Æ Não possui certas vulnerabilidades presentes nos mecanismos de controle de acesso, como a abertura para o mascaramento;

A forma de implantar o Serviço de Confidencialidade no acesso à MIB é o uso de criptografia em todos os acessos à mesma. O agente responsável pela MIB possui uma chave que é utilizada para sifrar todas as informações antes de armazená-las e decifrar as informações quando do acesso. A sifragem das informações contidas na MIB pode ser feita com a mesma chave que o agente utiliza para garantir a privacidade das comunicações ou com uma chave própria para a tarefa, podendo ser inclusive com o uso de uma técnica de chave secreta, mais eficiente em termos de tempo para criptografar e decriptografar. Isto porque o acesso à MIB é completamente independente de todo o processo de comunicação entre entidades. Tal mecanismo permite inclusive que o acesso em si possa ser realizado sem restrições, mas uma vez que as informações estão criptografadas, não há liberação efetiva das mesmas para aqueles que não possuírem as chaves.

**Figura 6.2 - Integridade e Autenticação no Agente.**

## 6.3.2 Algorítmos

Os algorítimos utilizados para a implementação das Interfaces de Segurança são divididos em Algorítimos para Autenticação e Integridade e Algorítimos para Confidencialidade. Estes dois algorítimos são apresentados a seguir:

- Æ Algorítimo para Autenticação e Integridade

Para se conseguir a garantia de autenticidade e integridade das mensagens que são trocadas entre as entidades componentes do Sistema de Gerência, cada Interface de Segurança deve implementar (indistintamente para agentes e gerentes) os algorítimos para o envio e o recebimento de mensagens a seguir, além de uma negociação preliminar para a troca de informações que serão necessárias para o desenrolar das comunicações, como as chaves públicas das duas interfaces e a determinação do primeiro valor que será utilizado para garantir a ordem das mensagens. A figura 6.2 mostra a Integridade e Autenticação no Agente.

A figura a seguir mostra a Integridade e Autenticação no Gerente.

**Figura 6.3 - Integridade e Autenticação no Gerente.**

- Envio de mensagens
    1. Recebe mensagem da Entidade “pura”

2. Calcular checksum da mensagem e criptografá-lo com chave privada própria, agregando-o na mensagem
3. Agregar campo “ordem atual”, conforme negociado previamente entre as partes
4. Gerar e agregar campo “próxima ordem”
5. Criptografar campos “ordem atual” e “próxima ordem” com chave pública do destinatário
6. Enviar mensagem para Interface de Segurança homóloga

- Recepção de mensagens
  1. Recebe mensagem da Interface de Segurança homóloga
  2. Decifrar checksum criptografado, usando a chave pública do remetente
  3. Calcular e conferir checksum
  4. Decifrar campos “ordem atual” e “próxima ordem” com chave privada própria
  5. Conferir campo “ordem atual” com valor esperado, conforme negociado previamente
  6. Armazenar campo “próxima ordem” para uso em futura comunicação
  7. Enviar mensagem para Entidade “pura”
- Negociação Preliminar

Quando da disponibilização de um novo agente para um gerente, este último toma a iniciativa de enviar mensagem para “negociação”. Esta negociação se dará em 2 passos:

1. Devem ser trocados entre o gerente e o agente em questão suas chaves públicas, por iniciativa do gerente.
2. Deve ser realizada a negociação sobre o valor inicial que conterá a ordem das mensagens para a manutenção da integridade e autenticidade. O gerente deve enviar mensagem indicando o primeiro valor que deverá ser usado na primeira mensagem operacional. O valor da primeira ordem deve ser confirmada pelo agente. Estas mensagens devem ser confidenciais, já utilizando as chaves públicas trocadas a priori.

1. Æ Algoritmos para Confidencialidade

Conforme apresentado, há dois momentos onde é necessária a confidencialidade: na comunicação e no acesso à MIB. Abaixo estão os algoritmos utilizados neste casos.

Æ **Confidencialidade na Comunicação**

A confidencialidade na comunicação deve ser garantida quando requisitada. As atividades relativas à confidencialidade devem ser realizadas sobre uma mensagem já preparada pelos mecanismos de integridade e Autenticação, sendo realizada, portanto, um nível abaixo.

A Interface de Segurança deve, então, implementar o algoritmo abaixo:

- Envio de mensagem
  1. Recebe mensagem da Entidade “pura”
  2. Agregar mecanismos de Integridade/Autenticação
  3. Criptografar mensagem com chave pública do destinatário
  4. Enviar mensagem para Interface de Segurança par
- Recepção de mensagem
  1. Recebe mensagem da Interface de Segurança par
  2. Decifrar mensagem, usando chave privada própria
  3. Verificar Integridade/Autenticação
  4. Enviar mensagem para Entidade “pura”

Æ **Confidencialidade no Acesso à MIB**

Para a confidencialidade das informações contidas na MIB, é necessário apenas que a Interface de Segurança dos Agentes contemple o seguinte:

- Acesso à Informação da base
  1. Recebe solicitação de acesso da Entidade “pura”
  2. Busca na MIB a informação desejada, que estará criptografada
  3. Decifra a informação, com a chave privada própria
  4. Envia informação para Entidade “pura”
- Manutenção de Informação

1. Recebe solicitação de manutenção de informação da Entidade “pura”
2. Criptografa informação com chave pública própria
3. Armazena/Altera informação na MIB

## 6.4 Conclusão

Nesta seção foi apresentado uma Arquitetura de Segurança (veja referência [9]) genérica para aplicação em Sistemas de Gerência de Redes. Tal arquitetura é extremamente flexível uma vez que permite sua aplicação em todos os Sistemas de Gerência baseados em entidades agentes e gerentes, ao mesmo tempo, não exige que tais entidades sofram alterações para suportá-la, tornando transparente a sua instalação.

As novidades da abordagem utilizada dizem respeito à utilização de seqüenciamento constante (a cada mensagem trocada entre entidades é realizada uma verificação da origem - autenticação). A ampliação do conceito de assinatura digital (que permite não só validar a origem mas também a integridade das mensagens) e ao controle de acesso à MIB ser feito através do serviço de confidencialidade (que não impede o acesso mas não revela as informações senão para o legítimo proprietário).

Embora a Arquitetura de Segurança apresentada seja genérica, ela já é implementada nos sistemas de gerência que estão baseados no modelo OSI. Isto ocorre porque a camada de Apresentação deste modelo já implementa serviços de criptografia das mensagens.

---

# 7. Estudo de Caso e Implementação de Aplicação SNMP

A partir do momento em que for definido qual o modelo de gerência de rede a ser implantado, o próximo passo será selecionar as ferramentas que auxiliem a execução deste trabalho. Se for um ambiente local, a complexidade é bem menor, visto que o próprio fornecedor de hardware e software, geralmente, oferece alguma ferramenta para gerenciar este ambiente.

Para um ambiente interligado, ou seja, onde existem vários sistemas interligados, muitas vezes empregando tecnologias heterogêneas, o índice de complexidade é infinitamente superior. Portanto, é muito importante que ocorra a escolha de produtos para gerência de rede, em conformidade à filosofia de gerenciamento da rede para o sucesso e bom desempenho da rede.

Nesta seção são analisados rapidamente dois sistemas de gerenciamento de redes existentes, e em seguida é apresentada uma aplicação específica desenvolvida sobre um deles. Os sistemas analisados são:

- Æ SunNet Manager desenvolvido pela Sun Microsystems Inc;
- Æ AIX NetView/6000 desenvolvido pela IBM Corporation;

## 7.1 SunNet Manager

O SunNet Manager é um pacote de software que contém serviços que auxiliam a gerenciar elementos de uma rede de computadores. Este software é composto por uma interface gráfica de apresentação da topologia da rede representada por figuras e que permite a interação dos usuários com os elementos componentes desta através da manipulação dos respectivos ícones e uma biblioteca de serviços gerente/agente que realizam a monitoração de vários aspectos da rede.

O ambiente SunNet Manager é baseado no modelo gerente/agente no qual o gerente é um processo disparado pelo usuário, normalmente da console do SunNet Manager, e o agente é um processo que coleta dados dos objetos gerenciados reportando-os ao gerente. A biblioteca de serviços do SunNet Manager possui vários serviços de monitoração e um conjunto de agentes disponíveis para o usuário. O ambiente SunNet Manager é uma plataforma que suporta o desenvolvimento de novas aplicações de gerência através da possibilidade de confecção de novos agentes que passam a fazer parte do conjunto original de funções de gerência.

# 7.2 AIX NetView / 6000

O NetView 6000 é uma aplicação de gerenciamento de redes disponível para a plataforma UNIX da IBM (AIX). Como todo software razoável desta natureza, o NetView permite o controle da rede sob seu domínio administrativo através de uma interface gráfica, com funções de configuração, verificação falhas e execução de funções de gerência sobre os recursos de sua rede.

O AIX NetView/6000 é uma aplicação de gerenciamento para redes baseadas nos protocolos TCP/IP. As informações gerenciadas são armazenadas na MIB e mantidas por um software agente que é executado em todos os sistemas gerenciados. A aplicação de gerenciamento recupera as informações da MIB comunicando- se com o agente através do protocolo SNMP. **Subagentes** podem ser usados para a comunicação com outros sistemas de gerenciamento. Um subagente é um processo que mantém uma MIB privada e comunica-se com um agente local ou remoto a fim de transmitir suas informações aos gerentes. Os subagentes se comunicam com os agentes através de dois protocolos: SNMPDPI para sistemas operacionais VM, MVS e OS/2, e SMUX para sistemas Unix. O uso desses protocolos é transparente para o gerente, este envia suas consultas e recebe as respostas e *traps* através do protocolo SNMP. O agente SNMP é responsável por traduzir as consultas para o protocolo apropriado e enviá-las aos subagentes.

Para gerenciar recursos novos (que não estão catalogados pelo sistema de gerenciamento) usando o *AIX NetView/6000*, o usuário deve adicionar a definição dos novos objetos na MIB utilizando a sintaxe ASN.1 e implementar um subagente para manter os novos objetos.

## 7.2.1 Conceitos de Gerenciamento do Sistema NetView 6000

Os exemplos de aplicação apresentados a seguir foram desenvolvidos utilizando-se ferramentas oferecidas pelo NetView 6000. Por esta razão este produto será um pouco mais detalhado.

A representação da rede alvo no NetView 6000 é feita através de um **mapa** composto por uma hierarquia de **submapas**, com um nível de detalhamento cada vez maior conforme percorremos esta hierarquia. Cada submapa é composto de objetos que são representados graficamente através de símbolos. Pode-se coletar dados dos objetos, executar aplicações de monitoramento sobre estes objetos, ou verificar eventos (***traps*** em SNMP) gerados por estes objetos que podem ser tratados pelo gerente (AIX NetView 6000).

Definiremos nos tópicos seguintes, estes conceitos importantes da interface gráfica do NetView 6000. Podemos ver um exemplo de um mapa e seus componentes na figura 1.

### 7.2.1.1 Objeto (*Object*)

Um objeto é uma representação lógica de uma entidade lógica ou física que existe em algum lugar de sua rede. É constituído de uma série de campos, chamados de **atributos** deste objeto, que especificam todas as características deste objeto. Todos os objetos e seus atributos são armazenados na base de dados do NetView 6000. Alguns exemplos de recursos que podem ser representados por objetos são: um computador, algum processo em algum computador, um endereço IP. Os objetos podem ou ser criados pelos usuários através da interface, ou pelas aplicações integradas ao NetView 6000, ou pelas aplicações criadas pelos próprios usuários.

Os atributos mais comuns de um objeto são: o seu nome, o seu endereço IP, se este suporta ou não SNMP, o tipo de hardware ou software que este representa e o seu estado atual. Um dos mais importantes atributos de um objeto é o **atributo de capacidade**, que é um valor booleano (TRUE ou FALSE) que indica certas características ou capacidades que o objeto apresenta, como, por exemplo, se o objeto suporta ou não SNMP. A capacidade define quais as ações que podem ser feitas quando trabalhamos com objetos, determinando as opções de gerenciamento que estarão ativas para este objeto.

### 7.2.1.2 Símbolos (*symbol*)

Um símbolo é usado para representar graficamente um objeto que aparece em um submapa de um mapa particular, sendo portanto uma representação gráfica dos elementos da rede e não sua representação real (lógica) como os objetos presentes na base de dados do NetView 6000. Pode-se usar diferentes símbolos para representar o mesmo objeto, mesmo que estes símbolos estejam em submapas diferentes.

Além de serem uma representação gráfica dos objetos, os símbolos apresentam características adicionais além daquelas do objeto que está representando. Estas características particulares variam de acordo com os símbolos que representam um objeto particular. Estas características particulares dos símbolos são:

- **Tipo do símbolo:** Consiste da classe a que pertence o símbolo que especifica qual é a figura externa na representação do símbolo, e de uma subclasse do símbolo que especifica o gráfico que será mostrado dentro desta figura. O NetView 6000 já vem com uma variedade de símbolos pré-definidos. Se for necessário, pode-se definir novos símbolos através de arquivos de registro de símbolos;
- **Variedade do símbolo:** Um símbolo pode tanto ser um ícone que representa um objeto da rede, como um símbolo de conexão que representa uma conexão entre objetos;
- **Localização do símbolo:** Um símbolo pode tanto residir no plano da aplicação, como no plano do usuário para um dado submapa;

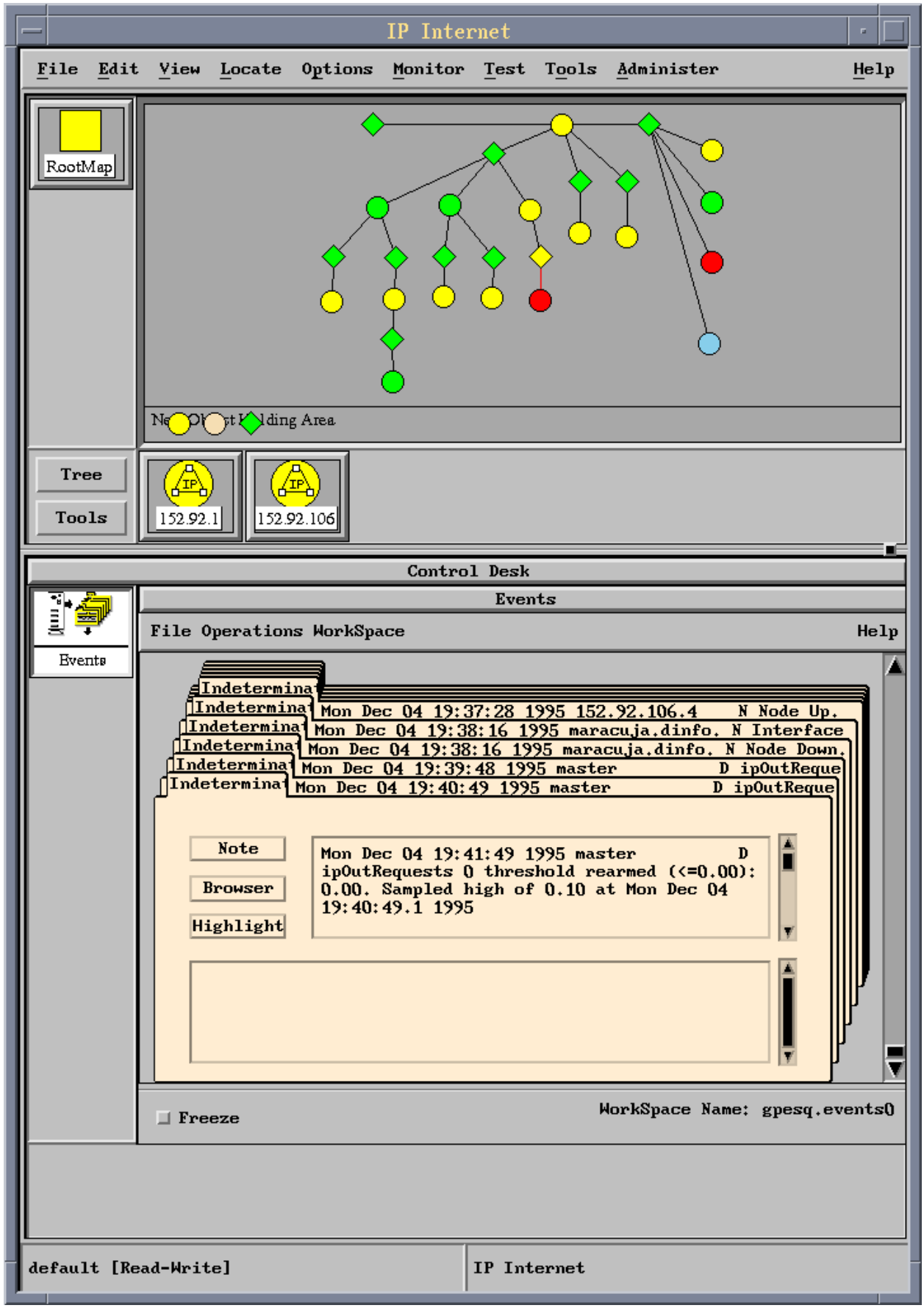


**Figura 7.1 - Um mapa representando uma rede no AIX NetView 6000**

- **Comportamento do símbolo:** Indica o que acontece ao se dar um duplo-clique com o mouse sobre o símbolo. Se um símbolo for **explosível** este se abre num novo submapa que apresenta mais detalhes sobre o objeto representado pelo símbolo. Este submapa é chamado de **submapa filho**. Este é o comportamento default de um símbolo. Se for um símbolo **executável**, o programa que aquele símbolo representa será executado. Um símbolo executável é representado em um submapa como um ícone em relevo;
- **Label do símbolo:** É usado para descrever o objeto que é representado pelo símbolo. O label aparece abaixo do símbolo que descreve. Como este não é usado para identificar o símbolo, pode-se utilizar labels duplicados. Não é necessariamente mostrado na representação gráfica do símbolo;
- **Estado do símbolo:** Mostra informações sobre o objeto ou a conexão representada pelo símbolo, ou seja, mostra o estado atual de um objeto ou de uma conexão, baseado em alguns conjuntos pré- determinados de regras. Os estados de cada símbolo são representados no submapa através do uso de cores diferentes para cada estado. Este estado é baseado em certos atributos do objeto que está sendo representado pelo símbolo. Podemos ver os significados destas cores na tabela 1.1.

#### 7.2.1.3 Mapas (*maps*)

Um mapa é uma coleção de objetos do AIX NetView 6000 e os seus inter-relacionamentos. Ele contém um subconjunto dos objetos da base de dados do NetView 6000. Como já vimos, são representados por símbolos que estão presentes nos vários submapas deste mapa. Pode-se utilizar diferentes mapas para representar diferentes domínios de gerenciamento, ou para prover diferentes representações de um mesmo domínio (por exemplo, para definir diferentes prioridades de gerenciamento dentro do mesmo domínio).

Os usuários podem criar ou deletar mapas, ou escolher um mapa para apresentar a partir dos mapas disponíveis. Para cada mapa, pode-se definir qual aplicação será responsável por seu controle. Somente um mapa pode ser aberto na interface gráfica, mas deve ser notado que não vemos um mapa diretamente, e sim, somente os submapas contidos neste mapa, sendo que cada submapa contém símbolos representado algum conjunto de objetos deste mapa.

Diz-se que um mapa aberto é aquele que está correntemente ativo. Você pode definir um escopo (domínio) para o mapa, e pode também alterá-lo ou deixar que as aplicações integradas ao NetView 6000 façam isso. Também permite que as aplicações alteram dinamicamente o mapa a fim de refletir o estado atual da sua rede. Somente os mapas abertos são dinamicamente alterados.

Você pode particularizar os mapas para atender as necessidades individuais de diferentes usuários. Pode-se particularizar as informações sobre objetos nos mapas que são criados. Se vários mapas contém o mesmo objeto, devem mostrar as mesmas informações sobre este objeto, pois na base de dados só existe um objeto representando uma mesma entidade da rede. Você decide particularizar os mapas quando:

  - Deseja distribuir a responsabilidade de gerência de sua rede a várias pessoas;
  - Deseja usar aplicações de gerenciamento que executem alguma tarefa determinada;
  - Deseja criar um mapa para refletir a responsabilidade de um administrador ou de uma esfera de influência.

Os modos de acesso aos mapas permitem que você limite ou proíba o acesso a um mapa aos usuários de seu sistema. Existem três modos de acesso:

- **Sem acesso (*No access*):** O usuário não pode acessar o mapa;
- **Acesso somente para leitura (*Read-only access*):** Os usuários podem ver a mudança dos estados dos objetos de um mapa, e ver as mudanças da topologia da rede, mas não podem alterar o mapa (deletar, criar e alterar objetos, submapas ou símbolos);
- **Acesso para leitura e escrita (*Read-write access*):** Os usuários além de verem os estados dos objetos de mapa e a mudança da topologia da rede, podem também alterar o mapa. Somente um usuário pode abrir o mapa neste modo. Se outro usuário tentar abrir o mapa neste modo quando algum usuário já o abriu neste modo, o mapa será aberto com acesso somente para leitura.

Cada mapa é composto pelas seguintes informações:

- **Nome (*Name*):** O nome dado ao mapa no momento em que foi criado. O nome do mapa deve ser único;
- **Submapa Raiz (*Root submap*):** O submapa de nível mais alto do mapa. Não podemos deletar este submapa;

- **Submapa Inicial (*Home submap*):** É o submapa que é mostrado ao se abrir um mapa. Pode-se definir qualquer submapa como o submapa inicial. Por default, este será o submapa raiz;
- **Layout de camada para o submapa raiz (*Layout algorithm for root submap*):** O algoritmo de camada que deverá ser usado no submapa raiz. O default é o de linha/coluna. Uma vez definido este algoritmo, não será possível alterá-lo;
- **Esquema de Composição de Estado (*Compound status scheme*):** É aplicado a todo o mapa e define como o estado dos símbolos em um submapa filho deverá ser propagado ao símbolo associado a este submapa no submapa pai;
- **Aplicações de configuração (*Configurable applications*):** Quaisquer aplicações para a configuração de um mapa disponíveis no sistema. Podemos habilitá-las ou desabilitá-las ao criamos um novo mapa;
- **Comentários (*Comments*):** Qualquer comentário ou anotação sobre o mapa, podendo ser usado para documentar o mapa, ou qualquer outra informação relevante sobre este mapa.

Podemos criar uma imagem estática de um mapa particular que conterá a representação gráfica dos estados de todos os símbolos de todos os submapas no momento da criação desta imagem, podendo ser usada para documentar os estados dos componentes de sua rede, para auxílio em grandes mudanças de configuração da rede, etc. Chamamos estas imagens estáticas de **imagens instantâneas de um mapa (*map snapshots*)**.

#### 7.2.1.4 Submapas (*submaps)*

Um submapa é uma coleção de símbolos que são mostrados em uma mesma janela gráfica. Permite visualizar uma determinada parte do mapa, isto é, uma parte da rede que é representada por este mapa. Cada submapa mostra uma diferente perspectiva das informações presentes no mapa sobre a rede representada. Os submapas são tipicamente organizados em hierarquias que permitem a você ver sua rede no nível de detalhamento que você achar melhor. Pode-se particularizar a organização dos submapas de acordo com os propósitos determinados pelos usuários.

Os submapas podem ser criados tanto por aplicações como por usuários. Uma aplicação pode criar um submapa para mostrar os símbolos relacionados aos objetos que esta gerência. Pode-se também deletar os submapas e modificar as características dos submapas em um mapa.

Chama-se de **objeto pai** ao objeto representado graficamente por um símbolo explosível que quando for explodido, mostrará o submapa associado a este objeto, chamado de **submapa filho**, que nos mostra uma visão detalhada deste objeto (o seu conteúdo). Um submapa pode ser independente, não apresentando inicialmente nenhum objeto pai, e sendo chamados neste caso de **órfão**.

Em resumo, um submapa permite:

- Criar uma visão particular de uma parte de um domínio de gerenciamento numa rede;
- Escolher uma coleção de símbolos para serem mostrados em um submapa particular.

O submapa raiz (*root submap*) permite construir uma visão em que podemos representar múltiplas redes ou subredes, permitindo representar redes diferentes num mesmo mapa.

As aplicações de rede e de gerenciamento podem usar o submapa raiz para construir hierarquias de submapas, sendo neste caso, um repositório no qual as aplicações colocam os símbolos que representam os objetos da rede. Você pode seleciona um objeto e mostrá-lo no submapa raiz para representar o nível mais alto de uma hierarquia de submapas.

O submapa inicial (*home submap*) é aquele que é mostrado ao se abrir o mapa. Pode-se escolher qualquer submapa como submapa inicial de um mapa. Por default e ao se deletar o submapa inicial, o submapa raiz será o submapa inicial.

A apresentação de um submapa permite que você escolha como serão apresentados os símbolo e o gráfico de fundo em um submapa. Pode-se optar por escolher a **apresentação em escala** em que é mostrada uma visão completa do submapa, com todos os símbolos e com o fundo do submapa, sendo que os tamanhos dos símbolos se adequam ao tamanho da janela disponível. Na **apresentação em zoom**, barras de rolagem aparecem na janela quando esta não for adequada ao tamanho do submapa. O fator default de zoom é de 1.

Os submapas apresentam três camadas ou planos:

- **Plano de Fundo:** Permite usar uma figura de fundo atrás dos símbolos de um submapa. Pode-se usar esta figura para definir um significado particular ao submapa ao vermos os símbolos na representação deste submapa. Pode-se criar diferentes fundos para diferentes submapas;
- **Plano de Aplicação:** Representa os objetos que são gerenciados por pelo menos uma aplicação de rede ou de sistema de gerenciamento. Se mais de uma aplicação gerencia um mesmo objeto, mais de um símbolo é mostrado no plano de aplicação representando aquele objeto. Se nenhuma aplicação gerencia um objeto, todos os símbolos que representam este objeto estarão no plano do usuário;
- **Plano do Usuário:** Contém os objetos criados pelos usuários e que não são gerenciados por aplicações.

O **layout de camada** define como os símbolos em um submapa são arrumados. Os métodos usados para ordenar os símbolos em um submapa são chamados de **algoritmos de camada**. Os símbolos podem ser dispostos em um submapa automaticamente através de um destes algoritmos ou manualmente pelo usuário. Os algoritmos definidos no AIX NetView 6000 são mostrados na tabela 1.2. Você escolhe o algoritmo ao criar o submapa, e uma vez definido não pode ser mais alterado. Se o submapa é criado por uma aplicação, esta pode especificar o algoritmo. Se nenhum algoritmo é especificado, um default é usado baseado no tipo do símbolo do objeto pai associado ao submapa (algoritmo automático de camada).

Um **símbolo de metaconexão** representa múltiplas conexões entre símbolos ou entre um símbolo e uma barramento pertencente a um submapa. A esta conexão está associado um **submapa de metaconexão** que mostra o estado atual de cada uma das conexões. Um submapa deste tipo é criado ao se adicionar mais de uma conexão entre dois símbolos ou entre um símbolo e um barramento. É permitido adicionar infinitas conexões entre dois símbolos ou entre um símbolo e um barramento em um mapa. Cada nova conexão é automaticamente adicionada ao submapa de metaconexão. Um submapa deste tipo apresenta as seguintes características particulares:

- Mostra todas as conexões representadas pelo símbolo de metaconexão;
- Usa o algoritmo de linha/coluna a não ser em conexões entre um símbolo e um barramento;
- Mostra dois pontos finais da conexão no submapa da metaconexão para cada conexão neste submapa.

#### 7.2.1.5 Aplicações (*applications*)

Uma aplicação é um programa que interage com os usuários através da interface gráfica do AIX Netview 6000. Os usuários podem criar suas próprias aplicações para interagirem com o AIX Netview 6000. As aplicações podem executar as seguintes ações:

- Processar requisições dos usuários;
- Criar ou deletar objetos, símbolos ou submapas;
- Mudar o conteúdo de um mapa;
- Prover funções especiais para amostragem.

A **aplicação IPMap** é a aplicação primária usada pelo programa AIX Netview 6000 e tem o objetivo de processar submapas e símbolos. Ela cria um objeto para cada nó IP de sua rede. Ela atualiza o mapa sempre que as informações sobre os objetos na base de dados mudarem, e que eventos ocorram no sistema ou alterações que ocorram nas conexões da rede. O Estado do objeto é propagado de duas formas: ou os objetos de um submapa de um nó contribuem para propagar o estado ao objeto pai, ou pode-se configurar a aplicação para que alguns objetos do submapa de nó contribuam para esta propagação de estado.

Esta aplicação apresenta os seguintes submapas:

- **Submapa Internet (*Internet submap*):** Mostra o particionamento lógico das redes e subredes IP conectadas através de gateways. Pode ser criado para distribuir os recursos da rede ao redor dos submapas, permitindo melhor controle da organização de seu mapa;
- **Submapa de rede (*Network submap*):** Apresenta a topologia física de uma rede em níveis de segmentos de rede;
- **Submapas de segmento (*Segment submaps*):** Apresenta a topologia física de um segmento de sua rede ao nível de conexões e conectores;
- **Submapas de nó (*Node Submaps*):** Mostra os componentes de um nó dispostos num formato de linha/coluna. A aplicação IPMap coloca as interfaces de um mesmo nó neste submapa.

A **aplicação XXMap** permite a você visualizar submapas que mostram as informações sobre objetos em topologias abertas. A aplicação permite que os usuários alterem os mapas. Mas tudo que o usuário adicionar só existirá ao nível do usuário. Também permite ver uma lista dos protocolos de gerenciamento que estão rodando em um objeto (se este for uma interface ou um nó), o que permite utilizar outros protocolos de gerenciamento além do SNMP.

Também é disponibilizada uma ferramenta de **construção de aplicações MIB** que permite que aos usuários criarem suas próprias aplicações para coletar, mostrar e salvar dados de objetos de uma MIB em tempo real. Com esta ferramenta, podemos criar aplicações sem ter que programar. O uso da ferramenta é útil quando queremos monitorar, em tempo real, o desempenho de objetos MIB específicos. Criamos aplicações MIB que contém certos objetos MIB de alguma base de dados regular. Estas aplicações podem produzir sua saída (com as informações sobre estes objetos) nos seguintes formatos:

- **Formulários (*Form*):** Usado quando somente temos uma única instância de um objeto;
- **Tabelas (*Tables*):** Pode-se ter várias instâncias de um mesmo objeto ao mesmo tempo;
- **Gráfico (*Graph*):** Usado somente com objetos MIB inteiros, contadores ou de porcentagem. Estes objetos podem ter mais de uma instância ao mesmo tempo.

Para executarmos uma aplicação MIB, deve-se selecionar um objeto no mapa da rede como parâmetro, e no menu Monitor escolher o item correspondente a aplicação. Nas aplicações de Formulários e de Tabelas, pode- se atualizar os dados ou dar o endereço da máquina manualmente num campo em que entra-se com o nome ou endereço IP de uma entidade da rede. Em uma aplicação gráfica pode-se realizar tarefas como:

- Escolher quais linhas do gráfico serão traçadas;
- O tamanho destas linhas;
- Se o gráfico é colorido ou monocromático;
- Salvar os dados em um arquivo;
- Escolher o intervalo de atualização do gráfico;
- Definir como as informações são mostradas, etc.

Também são disponibilizadas aplicações pelo NetView 6000 que permitem os usuários monitorar o desempenho da rede em tempo real. A maioria destas aplicações dispõe os seus dados em gráficos, mas algumas não mostram gráficos de seus dados, e sim, dispõem os seus dados em formulários ou tabelas. São disponibilizadas as seguintes aplicações: monitoramento do desempenho da CPU, monitoramento de tráfego de uma interface, aplicações para monitorar redes ethernet, para monitorar conexões TCP, e para monitorar o tráfego de mensagens SNMP.

#### 7.2.1.6 Eventos (events)

Eventos são informações sobre alterações que ocorrem nos objetos de rede, enviadas por agentes que monitoram os objetos e que são captadas pelo AIX NetView 6000, permitindo que a tarefa de o gerente gerenciar sua rede seja eficiente.

Nas grandes redes compostas de muitos objetos, vários agentes podem gerar eventos, o que faz com que o gerente fique sobrecarregado com o tráfego de mensagens, alocando um intervalo de tempo excessivamente grande para processar os eventos que chegam. Um gerente pode gerar pedidos ao agente ou para pedir informações sobre um objeto quando ele capta os eventos gerados pelos agentes, ou para informar o que o agente deve fazer com o objeto (que operação deve ser realizada).

Em uma rede gerenciada com o protocolo SNMP, os eventos ocorridos são chamados de ***traps***. Um *trap* é uma mensagem enviada por um agente SNMP para o gerente sem um pedido específico deste gerente ao agente. Os agentes enviam *traps* para o gerente para indicar que alguma condição particular existe no sistema do agente, como a ocorrência de algum erro ou algum evento.

Os eventos podem ser gerados nas seguintes circunstâncias:

- Algum limite definido no coletor de dados MIB foi excedido ou operações que excedem limites definidos para os resultados destas operações;
- Ocorreu alguma mudança na topologia da rede, não incluindo aqui adições manuais ou deleções de objetos que estão no plano do usuário em um submapa;

- Uma mensagem de informação foi gerada ou um erro ocorreu, o que pode indicar uma inconsistência, ou ocorrecia de algum comportamento inesperado;
- Um estado de um objeto mudou, ou o recurso associado ao objeto ficou inoperante, ou uma interface parou em resposta a um requerimento ICMP;
- Ocorreu uma mudança em uma configuração de um nó;
- Uma trap SNMP foi recebida de um nó gerenciável.

Existem dois tipos de eventos que podem ocorrem no AIX Netview 6000:

1. **Eventos de Mapa (*Map Events*):** Notificações que ocorrem quando um usuário ou uma aplicação faz algo que afeta o estado do mapa corrente, na interface gráfica do AIX Netview 6000. Como exemplo, se você adicionar uma conexão entre uma workstation servidora e um servidor em um submapa, será gerado um evento que é armazenado no arquivo de log. O contexto do submapa muda para incluir a nova conexão criada;
2. **Eventos de Rede (*Network Events*):** Ocorrem quando uma mensagem é enviada por um agente ou por outros gerenciadores para notificar uma ocorrência que afeta algum objeto da rede. Os eventos não são necessariamente refletidos no mapa. Como exemplo, se um agente SNMP que não pertence a sua região de gerenciamento for configurado para enviar *traps* para o NetView 6000, você receberá os eventos gerados por este agente.

O Coletor de dados MIB (*MIB Data Colletor*) que é executado automaticamente quando o AIX NetView 6000 começa sua execução, e cuja função é a de coletar dados de uma instancia simples de um objeto MIB para algumas entidades da rede, permite que os dados coletados sejam armazenados, que sejam traçados gráficos destes dados, ou que sejam gerados eventos ao AIX NetView 6000 (com certos números de *traps*) quando um limite definido for excedido. Ao contrário, as aplicações construídas através da ferramenta de construção de aplicações MIB, não podem gerar eventos e são apenas usadas para o monitoramento dos objetos da rede.

## 7.2.2 Aplicações construídas usando a ferramenta para construção de aplicações MIB (*MIB Application Builder*)

As aplicações dos exemplos que se seguem foram construídas no AIX NetView 6000 usando uma ferramenta para construções de aplicações MIB (*MIB Application Builder*). Não geram portanto, eventos para o AIX Netview 6000, pois as aplicações geradas por esta ferramenta não tem esta.

Por serem criadas com o uso desta ferramenta, as aplicações são constituídas por objetos da MIB do AIX NetView 6000, que armazenam certas características ou estatísticas relacionadas as entidades de uma rede, ou de algum dos componentes de alguma destas entidades.

### 7.2.2.1 Aplicação para gerar informações sobre um objeto

Esta aplicação é do **tipo formulário (*form*)** e objetiva mostrar as seguintes informações sobre um determinada entidade da rede:

- Nome da entidade (tipo da máquina);
- Uma pequena descrição desta entidade;
- Tempo em que a entidade está ativa após um *power-up* desta;
- Numero de partições que tem o disco;
- Percentagem de uso da CPU.

Foram usados os seguintes objetos ao se construir a aplicação:

- Três objetos da subárvore de sistema (system) da MIB (as três primeiras informações):
  - **iso.org.dod.internet.mgm.mib-2.system.sysName:** Nome administrativo para o nó selecionado (por default, é o seu endereço completo de rede);
  - **iso.org.dod.internet.mgm.mib-2.system.sysUpTime:** Tempo em que o sistema está ativo, isto é, tempo decorrido desde o último *power-up* do nó.
  - **iso.org.dod.internet.mgm.mib-2.system.sysDescr:** Dá uma descrição do nó, podendo descrever várias informações relacionadas a este nó (é um campo composto somente por caracteres ASCII);
- Dois objetos próprios do AIX NetView 6000, gerenciados pelos seus subagentes (as duas últimas

informações):

- **iso.org.dod.internet.private.ibm.netview6000SubAgent.nv6saFileSystem.nv6saFileSystemMounted:** Número de sistemas de arquivos montados na máquina (para funcionar, deve-se instalar o subagente do NetView 6000 na máquina);
- **iso.org.dod.internet.private.ibm.netview6000SubAgent.nv6saComputerSystemLoad:** Mostra a percentagem de carga da CPU da máquina que está sendo monitorada, com uma ordem de grandeza de 100 em relação a porcentagem de uso. Por exemplo, 2500 representa um uso de 25% da CPU (também só funciona se os subagentes do NetView 6000 forem instalados na máquina).

Na figura 7.2, vemos a interface da aplicação com o usuário e um exemplo de uso desta aplicação sobre o objeto (uma estação de trabalho) do mapa da rede alvo. Através desta figura, podemos ter uma melhor idéia das informações que são geradas pela aplicação.

**Figura 7.2 - Um exemplo de uma tela de saida da aplicação**

### 7.2.2.2 Aplicação para gerar informações sobre as partições de um disco em uma máquina

É uma aplicação do **tipo tabela (*table*)** que usa os objetos gerenciados pelos subagentes do NetView 6000 que o monitoram as partições do disco (sistemas de arquivos). Portanto, para que esta aplicação funcione em uma máquina particular, devemos instalar os subagentes do NetView 6000 nesta máquina. Para cada partição, são apresentadas as seguintes informações:

- Nome do volume associado a partição;
- Tamanho total (em KB);
- Número de blocos livres;
- Tamanho do bloco utilizado (em bytes);
- Diretório em que for montada;

**Figura 7.3 - Um exemplo de uma saida da aplicação**

Nesta aplicação, utilizamos alguns dos atributos de uma entrada particular da tabela **iso.org.dod.internet.private.ibm.nv6000SubAgent.nv6saFileSystemTable,** que contém uma entrada para cada sistema de arquivo existente na máquina. Cada entrada contém as seguintes informações em seus campos (os nomes abaixo são precedidos pelo nome do objeto acima mais **nv6saFileSystemEntry,** e mais o nome do campo dado abaixo):

- Campo **nv6saFileSystemName:** Nome do sistema de arquivo (*)
- Campo **nv6saFileSystemBlock:** Número de blocos totais neste sistema (*);
- Campo **nv6saFileSystemBfree:** Número de blocos livres no sistema (*)
- Campo **nv6saFileSystemBavail:** Número de blocos disponíveis para os usuários comuns do sistema (não super-usuários);
- Campo **nv6saFileSystemBsize:** Tamanho do bloco usado no sistema de arquivo (*)
- Campo **nv6saFileSystemFiles:** Número total de nós (*node*s) disponíveis para uso com arquivos;
- Campo **nv6saFileSystemFfree:** Número de nós (*nodes*) livres para uso no sistema;
- Campo **nv6saFileSystemDir:** Diretório em que foi montado o sistema de arquivo (*);

**Observação**: Os campos acima com * foram os usados pela aplicação.

Na figura 2.2, vemos um exemplo em que são mostradas informações sobre as partições da máquina alvo.

### 7.2.2.3 Aplicação para traçar um gráfico sobre o espaço disponível de disco em uma máquina

O objetivo desta aplicação do **tipo gráfica (*graph*)** é o de mostrar, durante um certo intervalo de tempo de atualização (definido através do menu da aplicação) de acordo com as especificações definidas pelo usuário (através do menu da aplicação), a variação do espaço disponível no disco (em número de blocos) para cada partição (sistema de arquivo) existente no disco. Para que a aplicação funcione, deve-se instalar os subagentes do NetView 6000, pois foi usado um objeto MIB que é gerenciado por estes subagentes.

Esta aplicação geralmente é mostrada na janela *Control Desk Window*, e utiliza a mesma tabela para sistemas de arquivo em disco descrita na aplicação acima, mas só utilizamos o campo que informa o número de blocos livres no sistema de arquivo. Para cada partição (*file sistem*) existente no disco, é desenhada uma linha de cor diferente, que representa a variação do espaço em disco para aquela partição.

Através do menu da aplicação, pode-se escolher entre mostrar algumas das informações (no caso, o espaço livre da partição) ao invés de todas. No exemplo da figura 7.4, vemos a variação do espaço livre de uma partição (a de nome /dev/dinfohome, montada em /home), que no caso é a partição do diretório dos usuários da máquina, num intervalo de 1m de atualização.

**Figura 7.4 - Um exemplo da aplicação gráfica, mostrando o espaço disponível na partição /home**

#### 7.2.2.4 Aplicação para traçar um gráfico sobre o tráfego de mensagens pelas interfaces de uma máquina

Esta aplicação do **tipo gráfica (*graph*)** foi construída para monitorar o tráfego de mensagens IP pelas interfaces de uma máquina. São consideradas tanto as mensagens que saem como as que entram na interface.

Assim como as outras aplicações gráficas, podemos definir como o gráfico apresentará os seus dados, quais informações devem ser mostradas, ou gravar os dados em um arquivo.

Utiliza os seguintes objetos da MIB relacionados ao endereçamento IP:

- **iso.org.dod.internet.mgm.mib-2.ip.ipInReceives:** Número de pacotes IP recebidos pelas interfaces da máquina;
- **iso.org.dod.internet.mgm.mib-2.ip.ipOutRequests:** Número de pacotes IP enviados pela máquina por suas interfaces;

Aqui também cada objeto será representado por uma cor diferente. Como exemplo, na figura 7.5 é mostrado o tráfego de pacotes IP pelas interfaces da máquina alvo.

**Figura 7.5 - Tráfego de mensagens pelas interfaces de uma máquina em uma rede**

---

# 8. Conclusão

O contínuo crescimento em número e diversidade de componentes das redes de computadores tem tornado a atividade de gerenciamento de rede cada vez mais importante. Os benefícios da integração dos sistemas computacionais de uma empresa, de natureza e portes diferentes, como forma de distribuir as tarefas e compartilhar os recursos disponíveis, são hoje uma realidade. As grandes redes corporativas, que são inter- redes formadas pela interconexão de pequenas redes, assumiram um papel fundamental para os negócios das empresas que delas se utilizam. Por este motivo, estas redes requerem um sistema de gerenciamento eficiente para que as informações da corporação estejam sempre disponíveis no local e no momento onde forem requisitadas.

Desde de 1986, vários grupos têm trabalhado para definir arquiteturas padronizadas (e abertas) para o gerenciamento de inter-redes heterogêneas, ou seja, inter-redes compostas por equipamentos de diferentes fabricantes. As principais arquiteturas abertas de gerenciamento de redes são relacioandas as tecnologias TCP/IP, também conhecida como Internet, e OSI da ISO e estas são conhecidas mais facilmente pelos nomes dos protocolos de gerenciamento utilizados, ou seja, o Simple Network Management Protocol - SNMP de TCP/IP, e o Common Management Information Protocol - CMIP de OSI.

Muitos produtos de gerenciamento já foram desenvolvidos obedecendo estes padrões. Por razões históricas, os primeiros produtos seguiram o padrão SNMP, e até hoje este é o protocolo que possui o maior número de implementações. Embora atualmente existam algumas aplicações de gerenciamento muito sofisticadas, a maioria destas aplicações possibilita apenas o monitoramento dos nós de uma rede e não possui “inteligência” para auxiliar os administradores de rede na execução de sua tarefa.

A arquitetura de gerenciamento SNMP, adotada na tecnologia TCP/IP, supõe a existência de estações de

gerenciamento, onde são executadas as aplicações de gerenciamento, e os nós gerenciados, que são os elementos da rede (estações, roteadores e outros equipamentos de comunicação), que desempenham funções de comunicação na operação normal da inter-rede, através dos chamados protocolos úteis. Estes protocolos são instrumentados para permitir o monitoramento e controle do seu funcionamento.

Este monitoramento e controle é exercido usando o paradigma de depuração remota, onde o nó gerenciado é concebido possuir uma coleção de objetos, cujos valores podem ser obtidos ou alterados através de operações remotas get e set. A coleção de objetos forma uma MIB, sob a guarda de um agente de gerenciamento. Comunicação entre este agente e um gerente, localizado na estação de gerenciamento, é feita utilizando o protocolo SNMP. Além das operações get e set, o SNMP inclui o trap, que é a maneira pela qual um agente possa tomar a iniciativa de avisar ao seu gerente da ocorrência de um evento excepcional.

O estado de desenvolvimento da maioria das aplicações SNMP disponíveis atualmente é desencorajador. As aplicações ou são muito específicas, ou são genéricas, e em geral elas são muito simples.

As aplicações específicas conseguem fazer um bom gerenciamento das entidades de rede que pertencem a um único fornecedor. Elas podem exibir várias informações de gerenciamento, permitem o controle de entidades através da operação set e podem até auxiliar o usuário na realização de tarefas de gerenciamento. Porém, estas aplicações não interoperam bem com módulos de MIBs definidos por outras organizações. Se a rede possui elementos de diversos fornecedores, normalmente diferentes aplicações são necessárias para o gerenciamento de diferentes dispositivos, com a perda das vantagens de um gerenciamento de rede integrado.

Em contraste, aplicações genéricas procuram oferecer ferrementas que podem ser configuradas pelo usuário para a coleta de informações de diversos módulos de MIBs. Existem dois formatos deste tipo de aplicação: browsers e ferramentas de monitoramento configuráveis. Ambas são muito limitadas, uma vez que as informações mais importantes sobre os objetos de uma MIB não estão disponíveis em um formato que possa ser compreendido por uma máquina.

Um browser simplesmente coleta dados de entidades da rede e exibe-os para o usuário. Uma ferramenta de monitoramento configurável pode ser programada para coletar informações da MIB selecionadas pelo usuário, executar algumas manipulações nos dados coletados e gerar alarmes, também definidos pelo usuário, baseados nos dados coletados ou na manipulação destes dados. Estes alarmes podem ser baseados na definição de valores limites ou em asserções como um comando condicional de uma linguagem de programação.

Além disso, várias plataformas de desenvolvimento de aplicações de gerenciamento de redes que já estão disponíveis permitem ao usuário desenvolver as suas próprias aplicações ou utilizar aplicações que sejam compatíveis com esta plataforma.

Uma parte significativa do processo de gerenciamento baseia-se na aquisição de informações sobre a rede, sendo as mais importantes aquelas relativas a erros, falhas e outras condições excepcionais. Tais dados devem ser armazenadas em forma bruta, sendo importante definir os valores aceitáveis como limiares de tolerância que, quando ultrapassados, determinam uma sinalização para pedir intervenção de um operador, ou o início de uma operação corretiva. Tais limites não são necessariamente absolutos, tais como a taxa de erros num enlace de dados, sendo necessário dispor de estatísticas de erros em função do tráfego existente. Um determinado limiar pode ser aceitável numa situação de carga leve na rede, mas intolerável numa outra situação, de carga mais intensa, no qual o número de ratransmissões faria com que o tráfego total excedesse a capacidade do enlace, afetando seriamente o tempo de resposta.

---

## Agradecimentos

Gostaria de agradecer os alunos Alexandre Porto, Flávia Carvalho, Patrícia da Silva e Pedro Melo, meus orientados no curso de graduação no Departamento de Informática da UERJ pela ajuda na coleta do material e no estudo do sistema NetView 6000 para a elaboração dos aplicativos.

---

# Referências Consultadas

1. ___, “TCP/IP Tutorial and Technical Overview”, International Technical Support Center
2. ___, NetView 6000 Programmers Guide, IBM
3. ___, NetView 6000 Reference Guide, IBM
4. BRISA, “Gerenciamento de Redes - Uma abordagem de Sistemas Abertos”, Makron Books, 1992
5. Carrilho, J. A, Madeira, E. R. M., “Gerência por Domínios”, 12º Simpósio Brasileiro de Redes de Computadores, Anais vol. II, Curitiba, maio de 1994
6. Case, J., Fedor, M., Schoffstalll, M. e Davin, J., “A Simple Network Management Protocol (SNMP) (RFC 1157), ,aio de 1009
7. Commer D. C. , Stevens, D. L., “Internetworking with TCP/IP - Design Implementation and Internals”, Volume II, , Prentice Hall Segunda Edição, USA, 1992
8. Commer, D. C., “Internetworking with TCP/IP - Principles, Protocol and Architecture”, Volume I, Prentice Hall Segunda Edição, USA, 1991
9. Leuca, J. E, Estphall, C, B, Specialski, E. S., “Uma Arquitetura de Segurança para Gerência de Redes”, 12º Simpósio Brasileiro de Redes de Computadores, Anais vol II, Curitiba, maio de 1994
10. McCloghrie, K. e Rose, M. T., “Management Information Base for Networkk Management of TCP/IP based Internets (RFC 1156)”, maio de 1990.
11. Moutinho, C. M, Stanton. M. A, “Aplicações de Gerenciamento de Redes Inteligentes” 12º Simpósio Brasileiro de Redes de Computadores, Anais vol I, Curitiba, maio de 1994
12. Rocha, M. A, Westphall, C. B “Gerência de Redes de Computadores através de novos Agentes”, 12º Simpósio Brasileiro de Redes de Computadores, Anais vol II, Curitiba, maio de 1994
13. Stalllings, W., “Data and Computer Communications”, Macmillan Publishing Co., Segunda Edição, USA, 1998.
14. Tanembaum, A. S., “Modern Operationg Systems”, Prentice-Hall Inc., USA, 1992.
15. Weissheimer, C. G, Tarouco L. M. R, “Distribuição da Gerência na Rede”, 12º Simpósio Brasileiro de Redes de Computadores, Anais vol I, Curitiba, maio de 1994