ALEXANDRE VITORETI DE OLIVEIRA

DEFINIÇÃO DE UM GATEWAY CMIP-SNMP

Dissertação apresentada como requisito
parcial à obtenção do grau de Mestre,
Curso de Pós-Graduação em Ciências
da Computação, Centro Tecnológico
Universidade Federal de Santa Catarina.
Orientadora: Elizabeth S. Specialski

FLORIANÓPOLIS

1996

ALEXANDRE VITORETI DE OLIVEIRA

DEFINIÇÃO DE UM GATEWAY CMIP-SNMP

Esta dissertação foi julgada para obtenção do título de Mestre em Ciências da Computação da Universidade Federal de Santa Catarina, e aprovada em sua forma final pelo programa de Pós-Graduação em Ciências da Computação:

Prof. Murilo Silva de Camargo - Dr.
Coordenador do Curso - UFSC

Profa. Elizabeth S. Specialski - M.Sc.
Orientadora - UFSC

Banca Examinadora:

Profa. Elizabeth S. Specialski - UFSC - M.Sc.
Presidente

Prof. Murilo Silva de Camargo -UFSC - Dr.
Co-orientador

Prof. Vitório Bruno Mazzola - UFSC - Dr.

Profa. Tereza Cristina Melo de Brito Cavalho - USP - Dra.

Profa. Maria Marta Leite - UFSC - M.Sc.

Florianópolis, 05 de setembro de 1996.

Agradecimentos

Agradeço àqueles que direta ou indiretamente
auxiliaram na conclusão deste trabalho,
mas especialmente aos meus pais,
minha esposa e minha orientadora Beth,
pessoas fundamentais nesta caminhada.

Sumário

Lista de Ilustrações/Tabelas

Lista de Abreviaturas/Siglas

ACSE - *Association Control Service Element*

APDU - *Application Protocol Data Unit*

ASE - *Application Service Element*

ASN.1 - *Abstract Sintax Notation One*

CAPT - *Called Application Process Title*

CMIP - *Common Management Information Protocol*

CMIPDU - *Common Management Information Protocol Data Unit*

CMIPM - *Common Management Information Protocol Machine*

CMIS - *Common Management Information Service*

CMISE - *Common Management Information Service Element*

DN - *Distinguished Name*

IAB - *Internet Architecture Board*

IP - *Internet Protocol*

ISO - *International Organization for Standadization*

LPP - *Lightweigh Presentation Protocol*

MIB - *Management Information Base*

MIS-Users - *Management Information Service Users*

NMS - *Network Management Station*

OID - *Object Identifier*

OSI - *Open Systems Interconnection*

PDU - *Protocol Data Unit*

PPDU - *Presentation Protocol Data Unit*

RDN - *Relative Distinguished Name*

ROSE - *Remote Operations Service Element*

SMAE - *Systems Management Application Entity*

SMASE - *Systems Management Application Service Element*

SNMP - *Simple Network Management Protocol*

TCP - *Transmission Control Protocol*

UDP - *UserDatagram Protocol*

Resumo e Abstract

Resumo

O objetivo principal deste trabalho é permitir a interoperabilidade entre sistemas de gerenciamento de redes que seguem modelos de gerenciamento diferentes. Como os modelos de gerenciamento mais difundidos são os modelos OSI e Internet, é apresentada uma proposta de desenvolvimento de um gateway que possibilite esta integração. Além disso, a Aplicação Gateway definida permite também, que a funcionalidade fornecida pelo modelo de gerenciamento de OSI possa ser aplicada sobre o modelo de gerenciamento Internet.

Abstract

The aim of this work is to allow internetworking between network management systems based on different management models. As the best well-known models are OSI and Internet we present a proposal to develop a gateway that allows this integration. Moreover the defined Application Gateway also allows that the funcionality provided by OSI model can be applied to Internet model.

Capítulo 1. Introdução

O surgimento de novas tecnologias e a constante necessidade de automatização dos processos fez com que as redes de computadores se tornassem fundamentais para as empresas. Com a constante evolução, essas redes estão cada vez mais complexas.

Em função do aumento da complexidade, decorrente principalmente do aumento do número de pontos e da inclusão de novas tecnologias, o gerenciamento manual tornou-se praticamente inviável. Os administradores de redes necessitavam de ferramentas que facilitassem a tarefa do gerenciamento.

Desta necessidade surgiram diversas ferramentas automatizadas que facilitaram a tarefa do gerenciamento. Entretanto, nenhuma destas ferramentas possibilitou um gerenciamento completo, ou seja, que abrangesse as cinco áreas funcionais definidas pela ISO. Para obter-se o gerenciamento completo, os administradores começaram a utilizar mais de uma ferramenta de gerenciamento. Como as ferramentas foram desenvolvidas por fabricantes diferentes e os recursos que são gerenciados também foram desenvolvidos por fabricantes diferentes, a integração entre estas ferramentas não é uma tarefa trivial. A solução ideal seria a utilização de um conjunto de ferramentas padronizadas e que permitisse o gerenciamento completo da rede.

A maioria das ferramentas automatizadas existentes atualmente foram desenvolvidas utilizando o protocolo SNMP. Buscando o gerenciamento completo a ISO desenvolveu o modelo de gerenciamento OSI que utiliza o protocolo CMIP. A tendência é a adoção do modelo OSI como padrão na área de gerenciamento de redes. Entretanto, não se pode deixar de lado toda a estrutura de gerenciamento SNMP existente atualmente.

Seguindo esta tendência, iniciou-se um projeto que visa fazer o gerenciamento de redes, inclusive redes que não seguem o modelo OSI, através de uma plataforma de gerenciamento OSI. Entretanto, não se pode desconsiderar todo o trabalho de gerenciamento já existentes nas redes, e como já foi dito a maioria das ferramentas de gerenciamento utilizam o protocolo SNMP, em razão da funcionalidade que este gerenciamento possibilita e também dos custos realizados. A solução ideal é a migração para um modelo mais completo, no caso o modelo OSI, sendo possível também, gerenciar os recursos que já estão sendo gerenciados através de outras ferramentas.

Este trabalho apresenta uma proposta de desenvolvimento de um Gateway, como forma de alcançar a interoperabilidade entre sistemas de gerenciamento de redes que seguem modelos de gerenciamento diferentes. Neste trabalho serão apresentados e discutidos os dois modelos mais difundidos, que são o modelo OSI (*Open Systems Interconnection*) e o modelo Internet.

O documento está estruturado em seis partes. No capítulo 2 são descritos os modelos de gerenciamento OSI e Internet, enfatizando suas características. No capítulo 3 é realizada uma breve discussão sobre aspectos que devem ser considerados para realizar a integração entre os dois modelos de gerenciamento. No capítulo 4 são analisadas as alternativas para a definição e implantação do Gateway CMIP (*Common Management Information Protocol*) - SNMP (*Simple Network Management Protocol*). No capítulo 5 é apresentada uma proposta de estrutura para o Gateway, enquanto que no capítulo 6 são discutidas as conclusões verificadas durante o desenvolvimento desta dissertação de mestrado.

1.1. Gerenciamento de Redes de Computadores

As redes de comunicação de dados tornaram-se cada vez mais importantes para as operações diárias de muitas empresas. O crescimento dessa importância das redes de computadores deve-se ao fato de que junto com a utilização dessas redes vem a eficiência e a competitividade [Stallings90]. A organização sente imediatamente o impacto quando o sistema de informação não está disponível. Além disso, a operação confiável do sistema de informação e sua rede de comunicação, é imperativo para a empresa.

Complexidade é uma característica das redes de comunicação de dados. Ela compreende muitos componentes, tanto de hardware quanto de software. Novos componentes oferecendo integração dados/voz, multiplexadores e roteadores, além de tantos outros, continuarão a adicionar complexidade a esse ambiente [Yemini93].

Adicionado a esta complexidade, está a necessidade das organizações de fornecer disponibilidade de seu sistema de informação 24 horas por dia, todo dia. As organizações estão enfrentando mais e mais demandas de seus clientes, e existe uma competição para fornecer serviços 24 horas, todos os dias do ano, especialmente quando esses serviços são internacionais e/ou atravessam muitas regiões [Subrata93].

Para gerenciar essa complexidade e permitir que a rede funcione com o desempenho máximo e disponível, são necessários ***talentos*** de várias disciplinas, altamente especializados [Bean93]. Dentre os profissionais necessários, destacamos o programador do sistema e o operador da rede, que executam papéis chave no gerenciamento da rede. Os programadores do sistema planejam, instalam e mantêm o software de gerenciamento e fornecem diagnósticos dos problemas da rede. Devido a experiência e treinamento formal necessário a essa tarefa, o programador do sistema é um recurso altamente especializado e escasso. A principal responsabilidade do operador

da rede é manter a rede funcionando eficientemente pela monitoração de seu estado e tomando algumas medidas quando situações anormais acontecem. O operador da rede também tem uma valiosa função, além do número de profissionais ser escasso, devido à extensiva experiência requerida [Hunter92]. O funcionamento perfeito e eficiente da rede depende diretamente da atuação desses profissionais na manutenção da rede.

Além disso, para gerenciar uma rede, certas condições devem ser consideradas [Yemini93]:

- deve existir um gerente, que executa aplicações de gerenciamento de rede que, por sua vez, coletam, processam, armazenam e mostram dados da rede;
- o sistema gerenciado deve conter agentes que respondam às requisições feitas pelo gerente;
- deve existir informação sobre as características físicas e lógicas dos objetos da rede que são parte de um sistema gerenciado;
- devem existir maneiras do gerente e do agente compartilharem e processarem informações sobre os objetos da rede.

Com isso, mostra-se nos capítulos a seguir, não a necessidade do gerenciamento, questão já considerada como aceita por todos, mas sim a utilização dos modelos e produtos de gerenciamento existentes no mercado e uma proposta que seria a transição desses modelos, não-padronizados segundo o modelo OSI, para o modelo de gerenciamento OSI, considerado mais completo.

1.2. A Ajuda de Ferramentas Automatizadas

A principal preocupação de um sistema de gerenciamento de redes deve ser a utilização, com a maior eficiência e eficácia possível, dos recursos das redes de computadores [Bean93]. Entretanto, para conseguir isso, é necessário que esse

gerenciamento seja integrado. Sem um sistema de gerenciamento integrado, a rede pode degradar até se tornar completamente ineficiente. Por isso, a interoperabilidade é de vital importância, pois é através dela que se conseguirá a integração dos sistemas de gerenciamento de redes, que permitirão o suporte às áreas funcionais definidas pela ISO (*International Organization for Standardization*), que são : gerenciamento de falhas, de segurança, de contabilização, de configuração e de desempenho[YEA93].

Geralmente, as empresas utilizam várias ferramentas para gerenciar a rede. Essas ferramentas permitem, dentre outras funções:

- apresentar a estrutura da rede através de uma interface amigável;
- diagnosticar problemas da rede;
- acompanhar o desempenho da rede através de funções de monitoração;
- mudança dinâmica do intervalo de *polling* para coletar mais ou menos dados;
- mecanismos para controle de acesso aos serviços de um agente.

Em [Brisa93] encontram-se definidos alguns produtos para gerenciamento de redes existentes atualmente no mercado, tais como, SunNet Manager® e NetView/6000®.

Essas ferramentas permitem, dentre outras tarefas, produzir registros de auditoria para todas as conexões, desconexões, ocorrências de filas, falhas na rede, e outros eventos significativos na rede [Hayes93]. Esses registros permitem determinar futuras necessidades de adicionar equipamentos, identificar erros comuns de um cliente, e estudar outras tendências de uso. Outro ponto importante é o uso de softwares de monitoração para diagnosticar e resolver problemas de desempenho. Essa é, talvez, uma tática óbvia que a maioria dos usuários de redes locais está empregando.

1.3. Sistemas de Gerenciamento de Redes

A maioria do pessoal responsável pelo gerenciamento de redes reconhece a importância de um software de gerenciamento de redes [Pyle93]. No entanto, dois problemas afrontam o gerente. Primeiro, a variedade de ferramentas necessárias para se efetuar um gerenciamento de redes completo faz com que surjam diferentes pacotes com diferentes interfaces de usuários, bem como diferentes requisições de plataformas de hardware; segundo, se os recursos incluem equipamentos de um número de fabricantes diferentes, é difícil encontrar um software que trabalhe efetivamente sobre todos os produtos.

A tendência atual é que as redes locais se tornem redes de redes locais e, com isso, a necessidade de gerenciar essas redes torna-se ainda mais importante. Até agora, os usuários de redes locais têm contado com simples facilidades para controle da rede fornecida pelos fabricantes de hardware, tais como NetView® da IBM, SunNet Manager® da Sun, OpenView® da HP e Managewise® da Novell [Nance92] e [Udell92]. Essas soluções são inadequadas por serem específicas. O problema maior da utilização destes produtos é que eles não permitem uma integração completa uns com os outros, e isso é fundamental no gerenciamento de redes, devido à variedade de equipamentos e softwares de redes que existem atualmente.

Assim, o que se necessita é que as seguintes requisições sejam cumpridas:

- sistemas de gerenciamento integrados com interfaces de aplicação para o usuário consistentes;
- definições de dados comuns e capacidade para compartilhar dados entre aplicações diferentes;
- reutilização de código para suportar equipamentos similares de diferentes fabricantes;
- mecanismos de comunicação para distribuir e coordenar tarefas de gerenciamento entre diferentes sistemas de gerenciamento;

- mecanismos de segurança para controlar a troca de informações de gerenciamento.

Como se vê, tais requisições visam possibilitar o gerenciamento de mais de um tipo de rede, isto é, redes que utilizam diferentes padrões de gerenciamento (OSI e Internet, por exemplo).

Para cumprir as requisições acima e, ao mesmo tempo, reduzir custos e complexidade do gerenciamento da rede, a **consistência** é fundamental.

Torna-se necessário então, um padrão de gerenciamento de redes que funcione como base para ferramentas de gerenciamento multifabricantes e multi-redes. A ISO tem desenvolvido um padrão para gerenciamento de redes chamado Estrutura de Gerenciamento OSI. Assim, é através da utilização de um padrão que espera-se solucionar o problema da interoperabilidade. Em [Brisa93] são descritos alguns modelos de gerenciamento de redes existentes. Além daqueles, pode-se citar também:

- *OSIMIS (OSI Management Information Service)* - é uma plataforma genérica de gerenciamento OSI. O sistema foi desenvolvido usando o ISODE (*ISO Development Environment*) e possui partes genéricas e específicas. As

partes genéricas formam uma infra-estrutura orientada a objetos para o desenvolvimento de agentes e gerentes OSI e um conjunto de aplicações de gerenciamento genéricas, isto é, independentes de MIBs (*Management Information Base*) específicas. As partes específicas são agentes que suportam MIBs específicas e aplicações de gerenciamento a elas associadas. A infra-estrutura referida compreende uma implementação do CMIS (*Common Management Information Services*)/CMIP junto com a infra-estrutura para desenvolver aplicações usando uma aproximação dirigida a eventos, exercendo um controle centralizado, fornecendo suporte para *polling* e também a infra-estrutura para manipulação de objetos definidos através da notação ASN.1 (*Abstract Syntax Notation One*) [ISO8824].

- *Plataforma de Gerenciamento*: Em [Thiry94] temos a proposta de uma plataforma de gerenciamento para redes heterogêneas, considerando para tal as normas da ISO, através de seu Modelo de Referência OSI. É importante ressaltar que a plataforma de gerenciamento descrita consiste de um suporte para o desenvolvimento de sistemas de gerenciamento, não devendo ser confundida com uma implementação de um sistema de gerenciamento OSI.

Do ponto de vista do usuário, a melhor aproximação seria obter um conjunto de ferramentas para o gerenciamento de redes que fornecesse várias características. Poderia conter uma simples interface para o operador com um poderoso, mas amigável, conjunto de comandos para executar muitas ou todas as tarefas de gerenciamento de redes. Poderia requerer uma quantidade mínima de equipamentos separados, isto é, a maioria do hardware e software necessário para o gerenciamento da rede poderia ser incorporado aos equipamentos existentes. Um sistema que fornece esse tipo de integração é geralmente referido como um Sistema de Gerenciamento de Rede.

Capítulo 2. Modelos de Gerenciamento de Redes

Conhecida a necessidade de ferramentas que auxiliem no gerenciamento de redes, apresenta-se agora os dois modelos de gerenciamento mais difundidos na comunidade internacional que são o Modelo de Gerenciamento OSI e o Modelo de Gerenciamento Internet.

2.1. O Modelo de Gerenciamento Internet

O modelo básico mostrado na figura 2.1 [Kalyan93], compreende o seguinte: no mínimo uma estação de gerenciamento de rede (**NMS** - *Network Management Station*), vários nós gerenciados e um protocolo de gerenciamento para comunicação da informação de gerenciamento entre uma NMS e um nó gerenciado. Uma NMS é um *host* que executa aplicações de gerenciamento de rede e para isso, se utiliza do protocolo de gerenciamento. Um nó gerenciado poderia ser um *host*, um *gateway* ou meio de transmissão. Esse modelo objetiva ter uma funcionalidade mínima na maioria dos nós e uma maior funcionalidade numa pequena fração dos nós na rede.

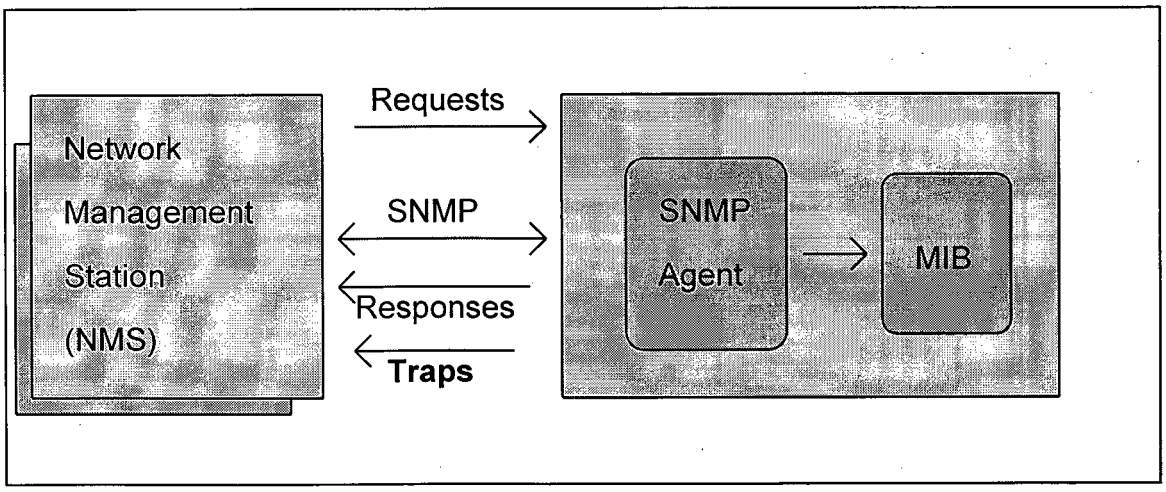


Figura 2.1 - Modelo Internet de Gerenciamento

Cada nó gerenciado suporta os protocolos operacionais, o protocolo de gerenciamento e a entidade de interface. A entidade de interface projeta as variáveis "reais" definidas e usadas pelos protocolos operacionais como objetos gerenciados. Essa entidade também define um conjunto de operações nessas variáveis. O conjunto de objetos e operações gerenciados são armazenados na MIB (*Management Information Base*). Cada NMS executa o papel de um gerente e suporta o protocolo e aplicações de gerenciamento. Os objetos definidos neste esquema podem ser dois tipos: escalares e tabela. Os objetos do tipo escalar funcionam como variáveis e os objetos tabela são listas de variáveis.

O modelo Internet é baseado no paradigma "*request-response*". Os papéis de gerente e agente são fixos. O gerente, pela mudança de um valor de um objeto gerenciado, pode instruir o agente a executar uma determinada operação de gerenciamento. O gerente pode requerer os valores de certos objetos gerenciados (que são modificados pelo nó gerenciado), para determinar o estado atual do nó. Além disso, todo aspecto de comunicação desse modelo está reduzido a apenas modificar e requerer valores (SET e GET, respectivamente) dos objetos gerenciados. O modelo Internet define objetos que têm as seguintes características: **acesso, estado, nome e sintaxe,** conforme a figura 2.2. O **acesso** controla os privilégios *read/write*; **estado** classifica os objetos gerenciados como opcionais, obrigatórios e obsoletos, na implementação, o **nome** denota o nome do objeto gerenciado; a **sintaxe** define a especificação da sintaxe do objeto. O nome usado para identificar um objeto é chamado *Object Identifier* (OID) que é um nome único na árvore de nomeação global.

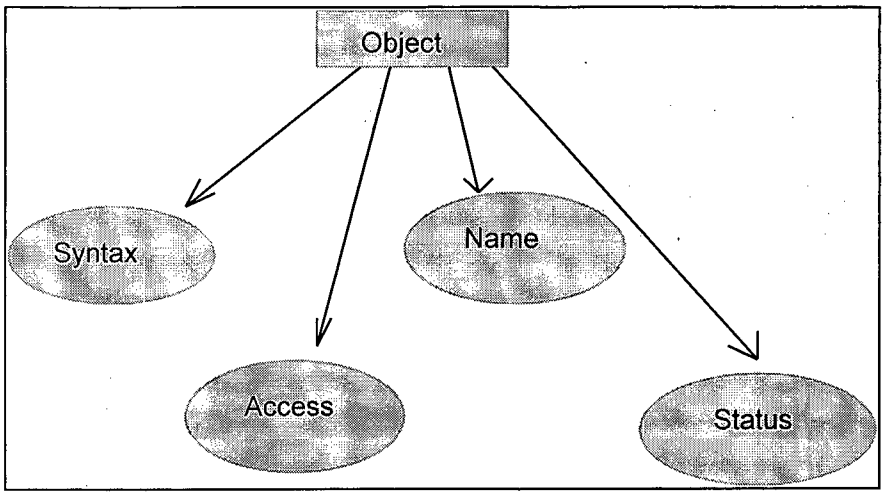


Figura 2.2 - Definição do Objeto Internet

O protocolo de gerenciamento suportado pelo modelo Internet é o SNMP. O protocolo SNMP usa como suporte o UDP (*User Datagram Protocol*). A versão 1 do SNMP suporta as seguintes operações básicas: Get , GetNext, Set e Trap. A operação Get permite ao gerente requerer de um agente o valor de uma variável particular. A operação GetNext é uma operação poderosa que permite MIB transversal, que é uma forma mais eficiente de o acesso às entradas de uma tabela. A operação Set é usada pelo gerente para modificar ou não o valor de uma determinada variável. Finalmente, a operação Trap permite ao agente assincronamente relatar a ocorrência de um evento ou falha para o gerente.

Outro importante aspecto desse modelo é o conceito de *proxy agent*. Um *proxy agent* ajuda no gerenciamento de nós que não implementam o conjunto completo de funções definidas pelo protocolo de gerenciamento, ou nós que tem um protocolo diferente do utilizado na rede. Ele responde a requisições de gerenciamento em favor dos agentes gerenciados e pode, além disso, atuar como um gateway quando existe a necessidade de conectar à rede um protocolo de gerenciamento diferente.

2.2. O Modelo de Gerenciamento OSI

O modelo de gerenciamento OSI é um modelo mais poderoso e melhor elaborado que os outros existentes; em função disto esse modelo se torna mais complexo. O gerenciamento de rede é feito sobre associações estabelecidas entre processos da camada de aplicação (chamados processos de gerenciamento). Um processo de gerenciamento pode assumir um dos possíveis papéis: Gerente ou Agente. O mesmo processo pode assumir diferentes papéis em diferentes associações, mas os papéis são fixos para uma dada associação, como determinado durante o estabelecimento desta associação. O processo gerente requer a execução de operações de gerenciamento pelo agente, que as executa e retorna as respostas para o gerente. O agente pode, também, gerar relatório de eventos assincronamente para serem enviados para o gerente como mostra a figura 2.3. As interações gerente-agente são suportadas pelo serviço CMIS [ISO9595] e implementadas pelo protocolo CMIP [ISO9596].

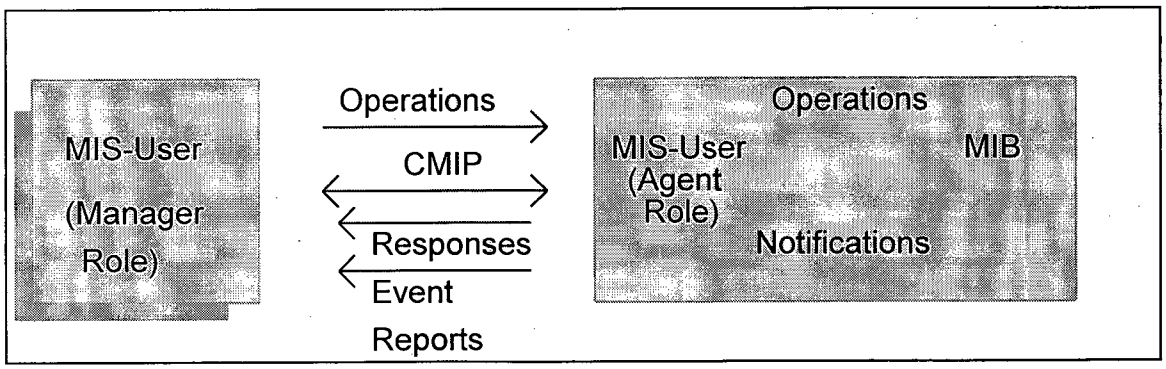


Figura 2.3 - Modelo OSI de Gerenciamento de Redes

Como no modelo Internet, cada agente OSI controla uma MIB. Entretanto, a estruturas das MIBs nos dois modelos são bem diferentes. O modelo OSI segue uma abordagem orientada a objetos. Um objeto gerenciado OSI é definido em termos de seus atributos, comportamentos, notificações e operações. **Atributos** são características específicas de um objeto; **operações** são aquelas que podem ser executadas no objeto;

notificações são emitidas pelo objeto para indicar algum evento; o **comportamento** dita as mudanças no objeto devido a operações executadas sobre ele.

Os serviços do CMIS utilizados pelo modelo OSI de gerenciamento são:

- M-GET: usado pelo gerente para recuperar valores de atributos dos objetos gerenciados;
- M-CANCEL-GET: permite ao gerente cancelar uma operação GET antes que tal operação se complete (este serviço é opcional);
- M-SET: permite ao gerente modificar os atributos dos objetos gerenciados;
- M-ACTION: permite implementar ações específicas baseadas em acordos bilaterais;
- M-CREATE: permite que o gerente requisite ao agente a criação de uma instância de um objeto gerenciado;
- M-DELETE: permite que o gerente apague uma instância de um objeto gerenciado;
- M-EVENT-REPORT: é usado pelo agente para relatar ao gerente a ocorrência de um evento associado com um objeto gerenciado.

A principal vantagem da estrutura do modelo OSI resulta de três funções permitidas pelo CMIS, chamadas *Filtering, Scoping e Synchronization* [Raman93]. A função *Scoping* permite ao gerente selecionar um simples objeto ou múltiplos objetos na sub-árvore da MIB para uma operação específica. A função *Filtering* permite ao gerente especificar um conjunto de condições que têm que ser satisfeitas para que a operação seja executada no objeto. A função *Synchronization* pode ser **atômica** ou de **melhor esforço**. Na sincronização atômica todos os objetos são verificados antes de executar uma determinada operação. Se todos eles aceitam/permitem a execução da operação, então ela é executada. Caso contrário, se um objeto não permite que a operação seja executada, então ela não é executada em nenhum. Já na sincronização de melhor esforço a execução de uma determinada operação é realizada nos objetos que atendem às condições e, para os objetos onde a operação falhou, relatórios de erros são enviados ao gerente. Numa requisição típica do gerente, *scoping, filtering* e *synchronization* podem

ser especificadas. O agente primeiro aplica o *scoping* e retorna a coleção de objetos adequadas para a operação. Então, aplica o filtro específico nos objetos resultantes. A operação requisitada é executada em todos os objetos selecionados pelo filtro, de acordo com a forma de sincronização especificada.

2.3. CMIP x SNMP

O objetivo deste item é mostrar as diferenças existentes entre os dois padrões de gerenciamento de redes (OSI e Internet) [Gering93]. A apresentação de tais diferenças é feita através da comparação das principais características dos padrões.

2.3.1. Metodologia de Estruturação da Informação

Uma metodologia que permita a estruturação da informação deve, entre outros pontos:

- fornecer ferramentas para modelar objetos do mundo real;
- reduzir a complexidade, explorando similaridades naturais no mundo real;
- fornecer um limite claro entre a implementação da informação e o modo como ela é acessada;
- promover a reutilização de especificações e implementações.

Ambos, CMIP e SNMP, usam o termo **objeto** para representar algo que pode ser manipulado através de seus protocolos. As metodologias orientadas a objetos fornecem esses mecanismos para atingir os seguintes objetivos: encapsulamento, classes, herança e polimorfismo. Mostra-se, a seguir, como esses conceitos são tratados pelos protocolos CMIP e SNMP.

2.3.1.1. Características do protocolo SNMP

A estrutura de gerenciamento Internet (SMI SNMP) especifica as regras e linhas gerais para nomes, sintaxe dos dados e notação para os objetos gerenciados. As várias MIBs são definidas de acordo com tal estrutura. O termo **variável** é freqüentemente usado como **objeto,** porque cada objeto representa um elemento de dado atômico. Um exemplo de uma variável é *IfAdminStatus* que é usada para representar o estado *up/down/testing* de uma interface física.

Encapsulamento

O protocolo SNMP não fornece um mecanismo formal para encapsular as variáveis utilizadas. O principal problema é que os programadores não tem um guia formal para manter a consistência entre as variáveis mantidas pelo sistema. As variáveis são acessadas e modificadas separadamente. O protocolo permite que múltiplas requisições GET ou SET sejam transmitidas em cada PDU, ele permite também, que as requisições sejam processadas fora de ordem.

A estrutura da informação de gerenciamento (SMI) do SNMP permite que as variáveis sejam agregadas em listas e tabelas. Esse não é um mecanismo de encapsulamento adequado, desde que o SNMP não tem listas ou tabelas orientadas a operações, isto é, um gerente não pode operar numa lista ou numa tabela como um todo. Além disso, a estrutura fornece um mecanismo não formal para especificar quais variáveis numa lista ou tabela são utilizadas de um modo especial, isto é, que um conjunto de variáveis representa um recurso gerenciado.

A operação SNMP SET causa efeitos colaterais. Se, por exemplo, for modificado o valor de uma variável *powerStatus* para zero, isso poderia desligar o equipamento.

Com isso, é possível verificar que a variável ficou habilitada para encapsular o comportamento. Entretanto, a utilidade dessa característica é significativamente reduzida porque as variáveis são limitadas a sintaxes não-estruturadas; não existe uma maneira estruturada para modificar uma variável através de uma lista de parâmetros. Uma alternativa para isso poderia ser a de definir uma variável por parâmetro e usar um múltiplo SET para alcançar o efeito desejado. Entretanto, isso introduz problemas de consistência, integridade e sincronização, que são contrários ao encapsulamento.

Classes

O conceito de classes de objetos no SNMP é representado pelas variáveis MIB ou Tipos de objetos. A estrutura define as regras e linhas gerais para os tipos de variáveis MIB e uma notação formal para especificá-las. Um tipo de variável MIB inclui:

- um identificador único para o tipo (OBJECT IDENTIFIER em ASN.1);
- uma sintaxe para o tipo;
- acesso *read-only, read-write, not-accessible*;
- status: *mandatory, optional, obsolete.*

Uma das principais vantagens do conceito de classe é a identificação e expressão de similaridades entre instâncias de objetos que estão sendo modelados. Essa vantagem não é aproveitada pelo SNMP porque ele combina a identificação das classes e das instâncias dessas classes. Todo nome de uma instância de uma variável MIB inclui o nome de sua classe, e a estrutura do nome da instância depende de como os dados estão agregados na MIB. Isso significa que a definição de um tipo de variável não pode ser reutilizada em diferentes partes da MIB para representar objetos similares. Por exemplo, se fosse definida uma variável na MIB para contar o número de erros de protocolos, e se desejasse usar diferentes instâncias desse tipo de variável para contar os erros do protocolo *token-ring*, erros do protocolo de sessão, etc. A menos que todos

esses contadores de erros de protocolos estejam estruturados na mesma lista ou tabela, eles não podem compartilhar a mesma definição de classe.

Herança

O protocolo SNMP não suporta herança. As principais vantagens da herança são reutilização da especificação, e, por extensão, reutilização do código. Isto não é feito no modelo Internet.

Polimorfismo

O protocolo SNMP define somente essas operações polimórficas nas variáveis:

- GET para recuperar o(s) valor(es) de uma ou mais variáveis;
- GET-NEXT para recuperar o próximo valor de uma ou mais variáveis;
- SET para modificar o(s) valor(es) de uma ou mais variáveis.

Além disso, define outra mensagem, TRAP, que não é dirigida a qualquer variável, mas ao sistema como um todo.

2.3.1.2. Características do protocolo CMIP

O gerenciamento OSI define um paradigma orientado a objetos para o modelo de informação de gerenciamento padrão. A seguir, veremos os contrastes e semelhanças com o SNMP já descrito.

Encapsulamento

O gerenciamento OSI fornece um mecanismo formal para encapsulamento. Os objetos gerenciados OSI encapsulam comportamento, atributos, operações e

notificações no escopo do objeto. O escopo do objeto é um modelo formal para especificação dos aspectos do recurso gerenciado que são visíveis aos processos de gerenciamento. Os aspectos de um recurso gerenciado que estão dentro dos limites do objeto são invisíveis aos processsos de gerenciamento. A especificação de uma classe de objetos define os comportamentos encapsulados, atributos, operações, notificações e restrições de integridade e consistência para instâncias de classes.

Uma diferença chave entre os objetos CMIP e SNMP é que um simples objeto CMIP pode modelar um recurso complexo. Por exemplo, objetos CMIP podem conter atributos para modelar características daquele recurso, enquanto um objeto SNMP modela somente uma característica do recurso. Assim sendo, objetos SNMP são como atributos CMIP, e o objeto CMIP não tem contrapartida em SNMP.

Classes

O gerenciamento OSI utiliza o paradigma da orientação a objetos, onde cada instância de objeto é um membro de uma classe. A especificação para uma classe de objetos usa uma notação formal, descrita em [10165-4].

Assim como os tipos de variáveis MIB SNMP, as classes de objetos gerenciados OSI são associados a identificadores únicos globais. Entretanto, porque a identificação da instância do objeto gerenciado OSI é independente da identificação da classe do objeto gerenciado, instâncias da mesma classe podem ocorrer em diferentes locais dentro de um mesmo esquema.

A especificação dos componentes das classes para atributos, ações, e notificações são associados a identificadores únicos que são independentes uns dos outros e do identificador da classe. Isso permite reutilizar as especificações dos atributos, ações e notificações em muitas classes de objetos.

Herança

O gerenciamento OSI permite a herança como um meio para definir classes de objetos que são extensões ou refinamentos de outras classes de objetos. Uma subclasse herda as características (atributos, ações, notificações e comportamentos) de sua superclasse e pode acrescentar características adicionais. O gerenciamento OSI requer herança estrita, que significa que as características não herdadas podem ser suprimidas ou redefinidas, e permite herança múltipla que possibilita a uma subclasse ter mais que uma superclasse.

Pelo fato dessas características serem associadas a identificadores globais e a herança estrita ser necessária, a herança múltipla não introduz ambigüidade para as características que são herdadas de mais que uma superclasse.

Polimorfismo

O gerenciamento OSI prevê uma forma mais avançada de polimorfismo, definida como alomorfismo, que permite a uma instância de objeto de uma classe ser gerenciada como se ela fosse uma instância de outra classe. Normalmente, classes alomórficas de um objeto são também superclasses deste objeto, mas esta não é uma condição obrigatória. Os relacionamentos alomórficos são modelados nos objetos gerenciados OSI e podem ser negociados e trocados com o protocolo CMIP.

O benefício primário do alomorfismo é ajudar a migração e co-existência para gerentes e agentes de diferentes versões e capacidades. A tabela 2.1 mostra um resumo da comparação efetuada nesta seção.

Característica	SNMP	CMIP
Encapsulamento	Simples, dados atômicos	Atributos,ações,notificações
Classes	Não reutilizável	Reutilizável
Herança	Nenhum tipo	Múltipla, estrita
Polimorfismo	Get, GetNext, Set como definido pelo SNMP	Get, Set, Action Event, como especificado pela classe

Tabela 2.1 - Características da OO nos protocolos CMIP e SNMP

2.3.2. Nomeação do Objeto

Os nomes das instâncias de objetos identificam as fontes e os destinos das operações e notificações de gerenciamento. Os protocolos CMIP e SNMP diferem claramente no modo como eles dão nome às instâncias de objetos.

A unicidade do nome é importante porque ele determina que os identificadores podem ser utilizados sem a possibilidade de uma interpretação errônea devido à ambigüidade. Um nome é ambíguo quando ele se refere a mais de uma instância de objeto num contexto onde é usado. Note que isso não acaba com a possibilidade de um objeto possuir mais que um nome. No gerenciamento, ambos, gerente e agente, devem compreender e manter o contexto de nomeação onde a ambigüidade não é permitida. Para o gerenciamento de redes de empresas, nomeação global é um requisito onde os recursos a serem gerenciados geralmente não podem ser restringidos a um conjunto de contextos pré-determinados. Por exemplo, não é razoável para uma empresa entender somente nomes SNA e requerer que todos os recursos gerenciados na empresa sejam associados a nomes SNA.

2.3.2.1. Nomeação de variáveis SNMP

A estrutura dos nomes das variáveis SNMP são definidas no padrão SMI para TCP/IP. Os nomes das variáveis SNMP são hierárquicos, não-tipados e baseados no tipo ASN.1 *object identifier*. Cada variável usa a sintaxe ASN.1 da forma {x,y}, onde x e y são seqüências de componentes ASN.1 *object identifier*. A primeira parte x é associada e fixada quando o tipo da variável é definido. A segunda parte y é associada e fixada quando a variável é instanciada na MIB.

Os nomes das variáveis SNMP são únicas somente no sistema; o mesmo nome de variável é reutilizado para instanciar variáveis em diferentes sistemas. Para fornecer nomeação global, os nomes das variáveis devem ser qualificadas como nomes globalmente únicos no sistema. Entretanto, o protocolo SNMP não faz previsão para transportar essa informação adicional. Uma limitação que isso impõe é que um gerente SNMP não tem definido como dirigir uma requisição para outro gerente SNMP para executar uma operação numa variável MIB num agente.

A estrutura do nome para uma variável SNMP é fixada quando a variável é projetada. O único meio para um usuário modificar essa estrutura é definir um novo tipo de variável, com um nome único.

Por exemplo, para identificar uma instância da variável *sysDescr*, a classe de objetos para *sysDescr* é:

iso org dod internet mgmt mib system sysDescr

1 3 6 1 2 1 1 1

Assim, o nome da instância que é referenciado por {x.y}, sendo y=0, por ser a primeira instância seria 1.3.6.1.2.1.1.1.0.

2.3.2.2. Nomeação dos Objetos CMIP

O modelo de informação de gerenciamento OSI define o mecanismo de nomeação para instâncias de objetos gerenciados baseados no *distinguished name* do Diretório X.500. Os nomes são hierárquicos, tipados e globais. Como eles são compatíveis com o serviço de diretório X.500, as aplicações de gerenciamento podem, mas não necessariamente, usar um serviço de diretório padrão para obter informações úteis sobre recursos. A natureza hierárquica dos nomes dos objetos gerenciados OSI é definida pela árvore de nomeação onde cada nó representa uma instância de objeto. O nome de cada instância é relativo ao seu nome superior. O nome global para uma instância é formado pela concatenação dos nomes relativos de cada uma das instâncias no endereço desde o ROOT até a posição da instância como pode ser visto na figura 2.4.

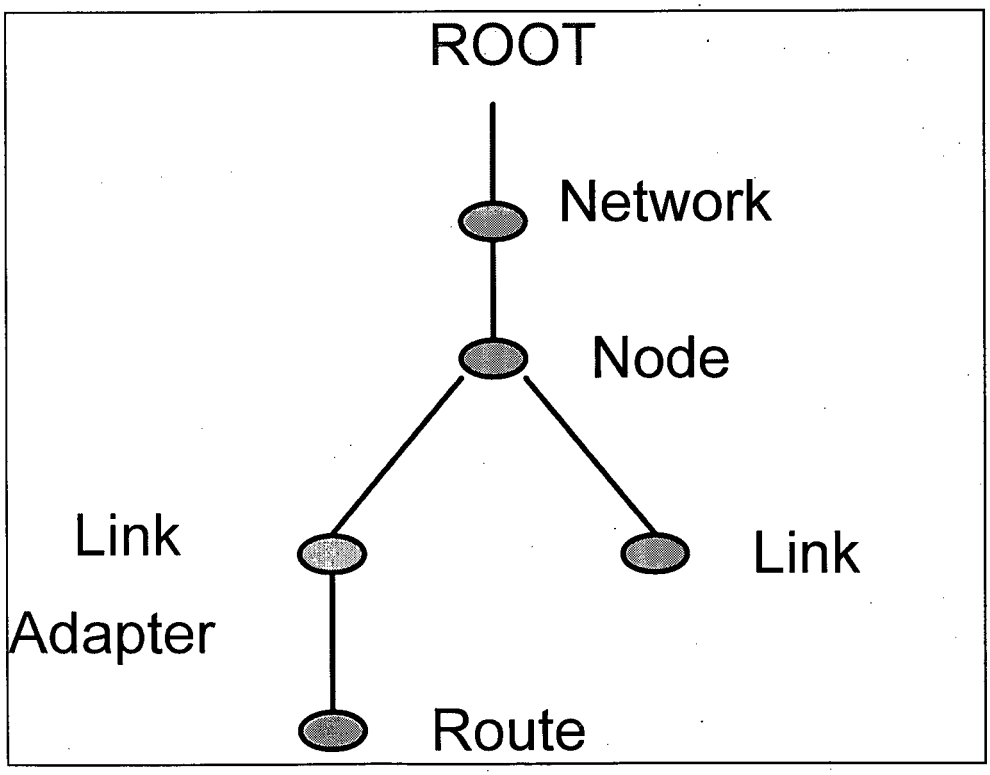


Figura 2.4 - Estrutura de nomeação no CMIP

O gerenciamento OSI usa nomes tipados porque os RDNs (*Relative Distinguished Names*) para uma instância de objeto são formados pela combinação de um de seus tipos de atributos e o valor dos atributos associados. Os nomes tipados permitem maior flexibilidade que nomes não-tipados. Como os tipos de atributos têm identificadores globais, os fabricantes podem definir estruturas de nomeação para suas classes de objetos gerenciados sem se preocupar com nomes ambíguos. Ao mesmo tempo, os usuários possuem flexibilidade para associar valores de atributo para cada instância.

Pelo fato dos nomes dos objetos serem tipados, hierárquicos e permitirem diferentes sintaxes ASN.1, eles requerem mais octetos que o protocolo SNMP para sua codificação. A tabela 2.2 apresenta um resumo das principais características de cada protocolo quanto aos seus esquemas de nomeação de objetos.

SNMP	**CMIP**
Único somente em um sistema simples	Globalmente único
Baseado no nome da classe quando da especificação	Independente do nome da classe, modificado na instanciação
Não-tipado	Tipado
Compact encoding	*Lengthy encoding*
Não compatível com X.500	Compatível com X.500

Tabela 2.2 - Diferenças no esquema de nomeação entre SNMP e CMIP

2.3.3. Representação dos Dados

O gerenciamento distribuído requer a troca dos dados entre gerentes e agentes para executar operações e transportar informação sobre os recursos a serem gerenciados.

Ambos, CMIP e SNMP, usam uma notação padrão, ASN.1, para especificar a representação dos dados, e um padrão relacionado, BER (*Basic Encoding Rules*), para codificação dos dados a serem transmitidos.

O protocolo CMIP não impõe nenhuma restrição sobre quais os tipos de dados ASN.1 que podem ser usados para representar a sintaxe dos atributos, notificações e ações dos objetos. O definição de uma classe de objetos pode conter qualquer um dos 26 tipos de dados.

O protocolo SNMP, propositadamente, limita os tipos de dados ASN.1 que o projetista da MIB pode usar: INTEGER, OCTET STRING, NULL e OBJECT IDENTIFIER. O lado positivo disso é que um agente SNMP necessita somente fornecer o subconjunto definido dos tipos de dados ASN.1. No lado negativo, sobrecarregar esses tipos de dados com semânticas, para dados mais complicados, implica que o gerente e agente devem permitir esta codificação, tornando-os mais complexos; eles não possibilitam as vantagens na auto-descrição das características dos dados fornecidos por ASN.1 e BER. Essa ligação entre simplicidade e flexibilidade é muito subjetiva. Entretanto, o protocolo SNMP não fornece algumas requisições comuns.

Alguns tipos de dados ASN.1 são particularmente úteis nas requisições de endereçamento internacional e de empresas. São eles:

- *Character Strings*;
- *Time*;
- SEQUENCE e SEQUENCE OF;
- SET e SET OF.

A notação ASN.1 fornece para uma variedade de tipos que representam conjuntos de caracteres, faixas que vão desde de um conjunto mínimo, *PrintableString*,

até um conjunto mais geral, *GeneralString*. No protocolo SNMP, *character string* poderia ser codificado como OCTET STRING. Desde que OCTET STRING não carrega identificação explícita do conjunto de caracteres que está sendo transportado, o gerente e o agente devem compreendê-la, a priori, numa variável por uma base de variáveis.

Similarmente, a notação ASN.1 define duas representações para tempo, *GeneralizedTime* e UTCTIME. O protocolo SNMP não tem um método padrão para codificação de valores de tempo.

A notação ASN.1 fornece tipos de dados para representar estruturas de dados. O tipo de dado SEQUENCE é usado para agregar um número fixo de dados. O tipo SEQUENCE OF é uma variação que permite um número de elementos variáveis, cada elemento sendo do mesmo tipo. Os tipos de dados SET e SET OF são como suas contrapartidas SEQUENCE e SEQUENCE OF, exceto a ordem dos elementos que não é importante.

O protocolo SNMP permite que os dados sejam agregados em forma de tabelas. De fato, toda tabela é definida como uma SEQUENCE OF *Entry*, e *Entry* é definida como uma SEQUENCE. Entretanto, o protocolo SNMP não permite que tabelas ou entradas de tabelas sejam transferidas como simples unidades. Aos invés disso, o gerente de aplicação deve operar em cada coluna ou entrada separadamente. Embora essa limitação simplifique a tarefa do agente de coletar e empacotar dados, ela complica a lógica do gerente, porque o mesmo deve combinar múltiplas variáveis em diferentes operações. Na prática, isso também libera o gerente e agente de sincronizar os dados e garantir sua consistência.

2.3.4. Comunicação

Os protocolos CMIP e SNMP têm demanda mínima de serviços de comunicação. Uma diferença chave entre os dois é que o protocolo CMIP requer um serviço orientado a conexão e o protocolo SNMP requer um serviço sem conexão. Outra diferença chave é que o SNMP assume o paradigma do *polling*, enquanto o CMIP assume um paradigma dirigido a eventos (*event-driven*).

2.3.4.1. Comunicação no protocolo SNMP

O protocolo SNMP opera sobre um serviço sem conexão fornecido pelo UDP (*User Datagram Protocol*) e o IP (*Internet Protocol*).

O tamanho de um pacote UDP é limitado pelas características operacionais da rede de roteamento Internet. O padrão IP requer que as implementações sejam capazes de transmitir, sob condições normais, pacotes de 484 bytes ou mais. Entretanto, um sistema de envio pode assumir, sob condições normais, que pacotes de 484 bytes ou menos serão recebidos pelo sistema receptor.

Como um serviço sem conexão IP e, por extensão UDP, não fornece nenhuma indicação, nem para o remetente nem para o destinatário, os pacotes podem ser perdidos ou descartados por uma série de razões, tais como:

- destinatário não disponível;
- pacote muito grande a ser transmitido pelo sistema intermediário;
- congestionamento da rede;
- perda de conectividade.

Porque os *traps* são eventos não-confirmados e são enviados como datagramas, um agente não tem a indicação de que o gerente não está disponível e que ele poderia mudar, para enviar seu *trap* para outro gerente.

O modelo sem conexão não fornece nenhuma maneira para controlar o fluxo de dados entre um gerente e um agente. Os gerentes que não são habilitados para processar grandes volumes de traps, não tem outra opção, senão descartá-los. O protocolo SNMP parcialmente resolveu esse problema usando os traps ponderadamente, para que um gerente saiba que algum evento ocorreu. Ele está acima do gerente para requerer mais informação sobre o evento com operações GET e GET-NEXT. O modelo sem conexão não fornece a um gerente e um agente a oportunidade para trocar e negociar informações tais como capacidades, parâmetros de segurança, e outros parâmetros operacionais.

Porque o modelo SNMP permanece com *polling*, o gerenciamento baseado em eventos não é possível. As aplicações que necessitam de informações periódicas, tais como aquelas para gerenciamento de desempenho, devem periodicamente "convidar" cada agente para coletar tais informações.

O protocolo SNMP adapta-se facilmente para operar sobre muitos outros protocolos, incluindo protocolos orientados à conexão, sendo esta uma de suas grandes vantagens.

2.3.4.2. Comunicação no protocolo CMIP

O protocolo de comunicação CMIP é definido para operar sobre o serviço de apresentação orientado à conexão OSI e exige a presença de dois outros elementos de serviço que são: ACSE (*Association Control Service Element*) e ROSE (*Remote Operations Service Element*), além do CMISE. O serviço da camada de apresentação

OSI fornece serviços de representação de dados, incluindo codificação e decodificação ASN.1 em BER.

Embora a camada de sessão OSI seja funcionalmente muito rica, CMIP, ROSE e ACSE usam somente poucas funções: procedimentos de segmentação/remontagem e de estabelecimento/encerramento de conexão. Os elementos CMIP, ACSE e ROSE não tem uma unidade de tamanho máximo definida; as camadas de sessão, transporte e rede fornecem o serviço de segmentação/remontagem para unidades de dados muito grandes. O protocolo CMIP é, também, prontamente adaptável a outros serviços de comunicação.

O serviço ACSE é usado pelas aplicações do gerente e do agente durante a fase de estabelecimento de associação para trocar e negociar parâmetros de acordo, incluindo contextos de aplicação e unidades funcionais. É também usado durante a fase de encerramento de conexão para liberar os recursos alocados em ambos, gerente e agente. O elemento ROSE é uma máquina de protocolo simples que é usada para estruturar requisições e respostas pelo protocolo CMIP. Na tabela 2.3 estão algumas características dos protocolos quanto a aspectos de comunicação.

Característica	**SNMP**	**CMIP**
Perda de dados Sinalizada	Não	Sim
Corrupção dados Sinalizada	Sim	Sim
Negociação	Nenhuma	Permitida e poderosa
Contexto de Aplic. Múltipla	Não permitido	Permitido
Limitações de tamanho	484 bytes	Sem limitação
Paradigma	Sem Conexão	Orientado a Conexão
Modelo de Interação	Polling	Dirigido a Eventos
Adaptável a outros modelos	Sim	Sim

Tabela 2.3. - Diferenças entre CMIP e SNMP quanto à comunicação

Capítulo 3. Modelo Funcional

Um aspecto crucial do desenvolvimento de aplicações de gerenciamento integrado é a criação de funções comuns para gerenciar diferentes tipos de recursos. Conforme descrito no capítulo 2, os protocolos CMIP e SNMP, diferem consideravelmente quanto às funções de gerenciamento existentes.

O alcance de uma funcionalidade é obtido pela modelagem de objetos de apoio (suporte). Os esquemas (produtos) de gerenciamento nem sempre abrangem as cinco áreas funcionais definidas pela ISO (falhas, configuração, desempenho, segurança, contabilização). Esta deficiência é decorrente da inexistência de objetos que possam dar apoio (suporte) a uma determinada funcionalidade. Por exemplo, para a área funcional de gerência de segurança, o modelo OSI define três funções de suporte: Auditoria, Controle de Acesso e Relatório de Alarme de Segurança. Cada uma destas funções requer um conjunto de classes de objetos que formam uma base de suporte para a definição de aplicações de segurança. Como ilustração, considere-se os objetos "*Target*" definidos na função de controle de acesso. Tais objetos não possuem classes correspondentes no modelo Internet. Isto implica na inexistência da funcionalidade de segurança (ou controle de acesso) na grande maioria dos produtos de gerenciamento que seguem o padrão SNMP versão 1.

Ao projetar-se uma plataforma de gerenciamento que integre os dois modelos (OSI e Internet), deve-se levar em conta que algumas das funcionalidades serão obtidas apenas em partes da rede, ou seja, apenas naqueles domínios onde o produto de gerenciamento oferece tal suporte. Em outros domínios, uma requisição de informação específica pode resultar em respostas do tipo: "impossível de se obter" ou respostas com vários campos incompletos.

A especificação de uma Aplicação Gateway entre os dois modelos deverá fazer um estudo exaustivo dos objetos de suporte às áreas funcionais definidas no modelo OSI, buscando a correspondência no modelo Internet e modelando a transformação das respostas desejadas (considerando respostas incompletas ou impossíveis de serem obtidas), ou seja, quando não houver a funcionalidade em determinado domínio da rede, o Gateway modelará as respostas recebidas para o formato da plataforma OSI.

As funções comuns, chamadas funções de gerenciamento de sistemas, são definidas como partes separadas do ISO/IEC 10164-X. Atualmente, essas funções de gerenciamento de sistemas estão em processo de padronização. São elas:

- Gerenciamento de Objetos;
- Gerenciamento de Estados;
- Atributos para Representação de Relacionamentos;
- Gerenciamento de Relatório de Alarmes;
- Gerenciamento de Relatório de Eventos;
- Gerenciamento de *Log;*
- Gerenciamento de Relatório de Alarmes de Segurança;
- Gerenciamento de Dados de Auditoria de Segurança;
- Gerenciamento de Objetos e Atributos para Controle de Acesso;
- Gerenciamento de *Accounting Meter;*
- Gerenciamento de Monitoração de Carga;
- Gerenciamento de Testes;
- Gerenciamento de Sumarização de Medidas.

Em [Brisa93] e [10164-X] encontram-se definidas as tarefas executadas pelas funções de gerenciamento citadas anteriormente, bem como algumas outras características do modelo de gerenciamento OSI.

A integração entre modelos de gerenciamento que seguem filosofias diferentes, deve levar em consideração as características de cada um desses modelos. Como descrito no capítulo 2, existem muitas diferenças entre os modelos que se pretende integrar.

A dificuldade de integração deve-se, principalmente, às diferenças de funcionalidades fornecidas em cada um dos modelos. Enquanto o modelo OSI é mais completo e elaborado, com funções como *scoping, filter* e *synchronization*, o modelo Internet propõe a utilização do protocolo SNMP, somente com funções básicas de gerenciamento. A versão 2 do protocolo SNMP já apresenta algumas características que aumentam a sua funcionalidade, minimizando os esforços de integração com o protocolo CMIP.

Com isso, o que se observa é que a integração entre os modelos de gerenciamento OSI e Internet é mais complexa e trabalhosa quando se utiliza a versão 1 do SNMP, e que esta tarefa é facilitada na versão 2, visto que ela apresenta uma maior funcionalidade, aproximando-se mais do modelo OSI. No entanto, a maioria dos produtos/ferramentas de gerenciamento utiliza a versão 1 do protocolo, razão pela qual este trabalho utilizará esta versão para a integração.

Capítulo 4. Gateway CMIP-SNMP

Os dois modelos para o gerenciamento de redes (Modelo OSI e Modelo Internet) foram apresentados no capítulo 2. O alcance de um gerenciamento integrado é obtido através da utilização de apenas um padrão ou de um gerenciamento unificado dos vários padrões existentes em uma rede. Esta última solução é a mais adequada, em vista da situação real das redes atuais que são compostas de produtos de diversos fabricantes. Neste capítulo é apresentada uma discussão sobre o gerenciamento de redes pertencentes ao domínio Internet e ao domínio OSI, desenvolvendo um paradigma que alcança o gerenciamento uniforme de redes (usando uma Aplicação Gateway). Pretende-se também, apresentar um guia para o projeto de uma Aplicação Gateway.

4.1.Definição do Paradigma

Para alcançar a integração entre sistemas é importante ter um bom paradigma que forma a base para a interoperabilidade. A seguir serão descritos três paradigmas que facilitam tal integração entre os protocolos CMIP e SNMP [Kalyan93]. Desses três paradigmas um foi escolhido para pesquisa. As razões que levaram a tal escolha são também explicitadas.

No primeiro paradigma, mostrado na figura 4.1, a NMS (*Network Management Station*) é composta por três camadas. A primeira possui as aplicações de gerenciamento com interfaces para o usuário. A segunda (NMLS - *Network Management Language System*) fornece uma interface para comandos padrão de gerenciamento para as máquinas de protocolo. Na última camada tem-se as máquinas de protocolo. O usuário especifica comandos na NML; esses comandos passam através da NMLS e são recebidos pelas máquinas de protocolo em um formato de comando padrão. As máquinas de protocolo convertem esse comando padrão para o seu formato específico.

Essas funções adicionam complexidade ao software gerente e, se implementado num simples nó da rede, poderia levar a requisições muito grandes de espaço e tempo pela NMS; se a implementação for distribuída, então uma interface separada necessita ser projetada para implementar as interfaces distribuídas de uma forma consistente. Com esse paradigma, os gerentes não podem alcançar a integração entre sistemas sem as funções mencionadas e também concentra muita complexidade na NMS.

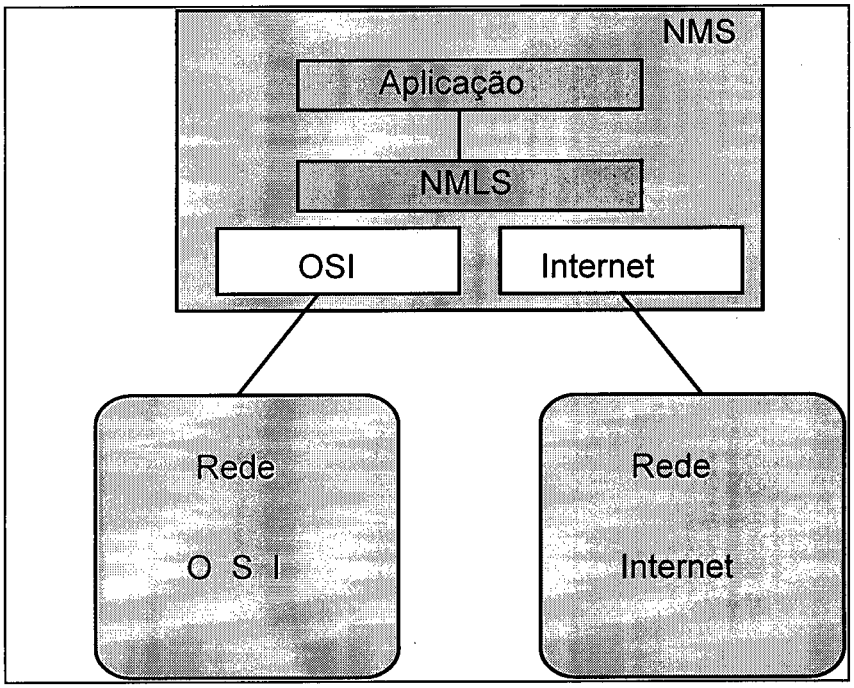


Figura 4.1 - Gerenciamento unificado de Redes Heterogêneas

O segundo paradigma, mostrado na figura 4.2, tenta a integração usando o conceito de duplicação das MIBs. Essa aproximação mapeia objetos na MIB Internet correspondentes aos objetos da MIB OSI (por exemplo, MIB-II é mapeada em OIM-II) [RFC1214]. Nesse paradigma, cada nó tem uma implementação de um agente SNMP, bem como de um agente CMIP. A MIB Internet e sua contrapartida mapeada OSI, estão presentes em cada nó Internet. Uma requisição de um gerente OSI é passada ao agente OSI, assim como uma requisição de um gerente SNMP é encaminhada ao agente SNMP. Os problemas apresentados por este paradigma são:

- a tradução da MIB é feita manualmente; conseqüentemente, mudanças nos objetos da MIB envolvem uma retradução; traduzir todas as MIBs existentes dessa forma poderia ser uma tarefa muito grande;
- necessita ter uma versão de uma MIB e um agente, que gerencia a MIB para cada tipo de rede sendo interconectada;
- a presença de mais do que um agente pode levar a problemas de múltiplos leitores-escritores.

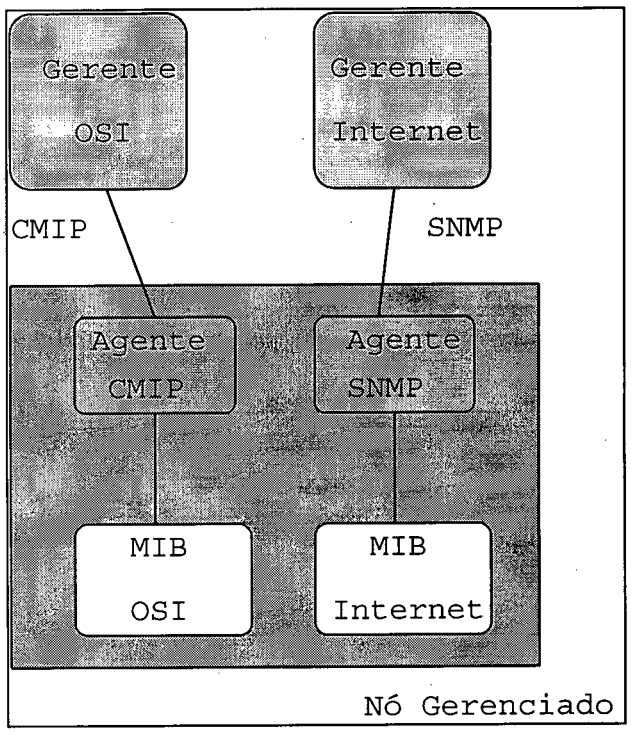


Figura 4.2 - Mapeamento de MIBs e agentes separados

O paradigma final a ser apresentado, mostrado na figura 4.3, usa o conceito de uma Aplicação Gateway. Nessa aproximação existem nós especiais projetados como gateways que implementam a tradução entre os protocolos CMIP e SNMP. O resto dos nós na rede implementam sua própria funcionalidade. Este paradigma permite a nomeação simplificada e o mapeamento de objetos é usado, independente de mudanças na MIB. O mapeamento usa as características da MIB Internet em lugar de um mapeamento de objetos um-a-um. Ou seja, não existe a tradução da MIB Internet. O gerente OSI possui uma visão dos objetos existentes em determinado agente SNMP.

Esse paradigma não requer mais do que uma implementação do agente, uma vez que o agente CMIS pode fazer a tradução entre protocolos e somente algoritmos de tradução necessitam ser adicionados. Com tal esquema, o gerente no domínio OSI pode gerenciar nós Internet sem qualquer mudança. Isso definitivamente é a maior vantagem do lado OSI, uma vez que adicionar maior funcionalidade ao gerente significa que isso tem que ser adicionado a todos os nós, pois qualquer nó poderia funcionar como um gerente.

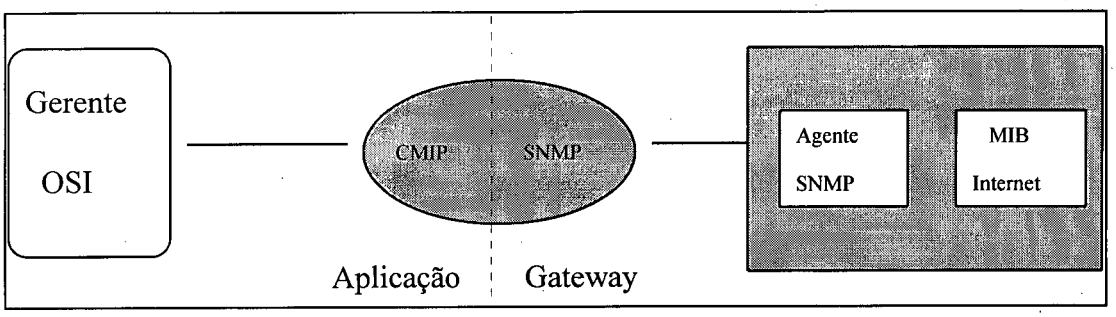


Figura 4.3 - Aplicação Gateway

A estrutura de Aplicações Gateways no domínio OSI de um lado e no domínio Internet de outro é mostrada na Figura 4.3. Um gateway recebe requisições CMIP e envia requisições SNMP; ele recebe respostas/traps SNMP e responde no lado OSI com respostas CMIP para alcançar a interoperabilidade. Se uma interface conveniente pode ser gerada, então várias traduções diferentes poderiam ser obtidas pelo agente, apenas permitindo ao agente escolher a tradução adequada. Esse paradigma garante a confiabilidade, uma vez que a perda de um gateway não resulta na perda do controle do gerente. Pela escolha de um gateway alternativo, o gerente pode continuar desimpedido. Se um gateway alternativo não está disponível, ainda assim, somente a parte da rede conectada ao gateway com problema não poderá ser gerenciada. Outra vantagem é que somente as traduções requisitadas necessitam estar presentes em cada um desses gateways. Todas essas características levaram à escolha deste paradigma para a pesquisa de interoperabilidade CMIP/SNMP.

4.2. Projeto de Aplicação Gateway

Os esquemas de nomeação e mapeamento de instâncias são muito importantes, pelo fato dos dois domínios de gerenciamento (OSI e Internet) possuírem esquemas inteiramente diferentes. Conforme descrito no capítulo 2, o esquema Internet define dois tipos de objetos gerenciados: **escalares**, que tem somente uma instância e são estáticos, e **tabelas**. Os objetos escalares são nomeados usando *Object Identifiers* (OIDs) que são únicos. A instância é identificada usando um sufixo padrão para esses OIDs. Os objetos **tabela** são definidos em termos de linhas, onde cada linha de objeto é constituída de vários campos colunas. Cada campo coluna é nomeado usando um OID. A unicidade de uma entrada na linha é alcançada pelo uso de uma seqüência de campos chamados **índices**. O índice é normalmente composto de valores de entradas nas linhas pertencentes a colunas especificadas. Uma variável coluna numa linha é identificada unicamente acrescentando o valor do índice ao OID do campo coluna que necessita ser modificado.

O esquema OSI oferece uma aproximação onde um conjunto de linhas gerais são fornecidas para a definição do objeto e é permitido a um usuário especificar a definição das classes de objetos baseadas nessas linhas gerais[10165-4]. Na estrutura OSI, classes de objetos são nomeadas usando OIDs. Cada atributo de um objeto é identificado usando um OID. Entretanto, a identificação da instância usa o conceito de *containment tree* e é independente dos OIDs usados para nomear as classes e atributos. A *containment tree* é uma instância de uma árvore dinâmica que representa os relacionamentos *containment* entre instâncias de objetos gerenciados. Não existe conceito equivalente no lado Internet. No modelo OSI, cada instância de objeto na árvore está contida em outra instância chamada **pai**. Todo **filho** de um dado pai é identificado unicamente através de um *Relative Distinguished Name* (RDN) que é um par de valores de atributo. Um *Distinguished Name* (DN) é obtido pela concatenação

dos RDNs desde a raiz da árvore da instância até o nó que representa a instância de objeto que está sendo considerada.

Visto que os dois esquemas são radicalmente diferentes, encontrar um método claro e eficiente de mapeamento de objetos de um esquema para outro é de grande importância para alcançar a integração de sistemas. Tal mapeamento garantirá que os domínios de gerenciamento sejam capazes de endereçar os objetos e instâncias de objetos que estão mantidos no esquema local, transpondo as diferenças. Em suma, mapeamento de nomes deve ser simples e fácil de automatizar. Além disso, deve ser genérico o suficiente para que mudanças nos objetos gerenciados não levem a requisições de novos esquemas de gerenciamento. O esquema deve capturar as características essenciais das MIBs.

4.3. Mapeamento Funcional

Ao invés de alcançar a interoperabilidade dos dois domínios de gerenciamento fornecendo diferentes serviços, é necessário que os serviços ou funções fornecidas por um domínio sejam traduzidos para o outro. Tal tradução é alcançada pelo mapeamento funcional. O mapeamento funcional permite, então, que uma operação executada num domínio seja mapeada para uma ou mais operações no outro domínio.

O principal problema no mapeamento funcional surge da diferença na complexidade das funções oferecidas pelos dois domínios. Uma função poderosa num domínio pode ser mapeada em várias funções simples no outro. Assim, pode não ser possível mapear todas as funções complexas fornecidas por um domínio, devido a menor funcionalidade fornecida pelo outro.

Os serviços definidos no CMIS permitem consultar e modificar atributos dos objetos, relatar eventos e criar e remover instâncias de objetos. Em adição, o serviço

CMIS define três outras funções que são: *scoping, filtering* e *synchronization*. Essas funções adicionam complexidade no mapeamento para o domínio Internet, visto que não existe contrapartida dessas funções. Assim, qualquer esquema de mapeamento funcional entre os protocolos CMIP e SNMP deve possibilitar a execução de tais funções. Esses mapeamentos devem ser aplicados em conjunção com o mapeamento básico entre serviços nos dois domínios. Por exemplo, um M-SET pode ser mapeado para um simples Set em um objeto. Entretanto, se *scoping* e *filtering* resultam na seleção de mais de um objeto gerenciado, então o mesmo serviço M-SET poderá ser mapeado para um ou mais Sets, se a técnica do melhor esforço para sincronização for desejada. Se a técnica atômica for utilizada, a realização de um Set pode não ser possível, e neste caso, deve haver a possibilidade de retornar à situação anterior.

O tamanho dos pacotes e a largura de banda podem apresentar problemas durante a função de mapeamento. As restrições no tamanho dos pacotes podem forçar que uma requisição no lado CMIP tenha que ser "quebrada" em mais de uma requisição no lado SNMP. Assim, uma Aplicação Gateway terá que manter a informação de estado para tal requisição. Isso pode adicionar complexidade ao gateway.

A transformação funcional pode resultar em um dos dois possíveis casos: uma simples operação no sistema pode resultar em uma ou mais operações no outro sistema; várias operações num sistema podem ser mapeadas para uma simples operação no outro sistema. Em outro caso existe uma distribuição desigual dos pacotes resultantes da transformação funcional. Isso poderia ocasionar um aumento na largura de banda do sistema no qual múltiplas operações são solicitadas. O esquema da transformação funcional deve garantir que a transformação resultante seja eficiente em tempo e largura de banda.

Fica claro que existem grandes diferenças nos esquemas adotados pelos dois domínios para acessar e atuar sobre objetos gerenciados e percorrer a árvore de

gerenciamento. Qualquer transformação funcional adotada deve ser simples, clara, eficiente e fácil de automatizar. Os parágrafos seguintes apresentam o projeto dos aspectos funcionais de um gateway. As transformações funcionais são unidirecionais, isto é, do domínio OSI para o domínio Internet, visto que o modelo OSI é mais abrangente e funcional que o modelo Internet.

No modelo OSI, a geração de DN(*Distinguished Name*)/RDN(*Relative Distinguished Name*) é importante pois o gerente precisa especificar o DN quando requisita uma operação num objeto gerenciado. Quando acessa um objeto escalar, o DN é o da instância do objeto escalar ao qual o atributo requisitado pertence. As instâncias dos objetos escalares têm RDNs fixos. O gateway deve realizar um mapeamento desses identificadores de objetos (DNs) de forma a identificá-los no lado Internet.

O esquema do DN possibilita a um gateway extrair somente as entradas que combinam com o do agente SNMP ao invés de solicitar várias entradas e só então filtrá-las. Isso permite a economia de tempo e largura de banda. O gateway possui tabelas internas que permitem que a informação da MIB seja armazenada. Essas tabelas contém características dos objetos gerenciados Internet e são usadas para interpretar e traduzir os DNs recebidos nas requisições do protocolo CMIP.

A execução da função *Scoping* pode resultar na seleção de múltiplos objetos. Isso significa que uma operação especificada (como M-GET) deve ser executada em todo ou num subconjunto dos objetos selecionados. No lado Internet, *scoping* pode ser mapeado pela repetição da operação requisitada para cada objeto selecionado. Essa repetição é implementada como parte de um mapeamento funcional de uma Aplicação Gateway. Essa Aplicação Gateway mantém a informação da MIB mapeada e usa isso para decidir quais objetos necessitam ser selecionados para a operação requisitada.

As seguintes escolhas são possíveis numa requisição *scoping*:

1. somente o objeto base;

2. n níveis subordinados ao objeto base;

3. o objeto base e todos seus subordinados.

No caso 1, o gateway faz o mapeamento da função *scoping* como uma simples requisição Get para recuperar o objeto base. Na situação 2, o gateway tem que percorrer a MIB e retornar com todos os objetos gerenciados que estão no nível definido pelo objeto especificado. Por exemplo, se todos objetos estão no nível 2 do grupo de objetos IP especificados pela função *scoping*, então o gateway terá que retornar todas a linhas de objetos, conforme mostrado na figura 4.4. Os objetos do nível 1 são os escalares e instâncias de objetos **Tabela**. O gateway usará a operação GetNext para extrair a informação sobre todos os objetos no nível especificado. No caso 3, a função do gateway é similar, com a diferença que o gateway extrairá informação de todos os objetos definidos na MIB abaixo do objeto especificado. Por exemplo, se a função *scoping* atua sobre todos os objetos do grupo IP, então o gateway usará o operador GetNext para extrair todos os objetos definidos abaixo do grupo de objetos **IP** usando o OID do grupo IP.

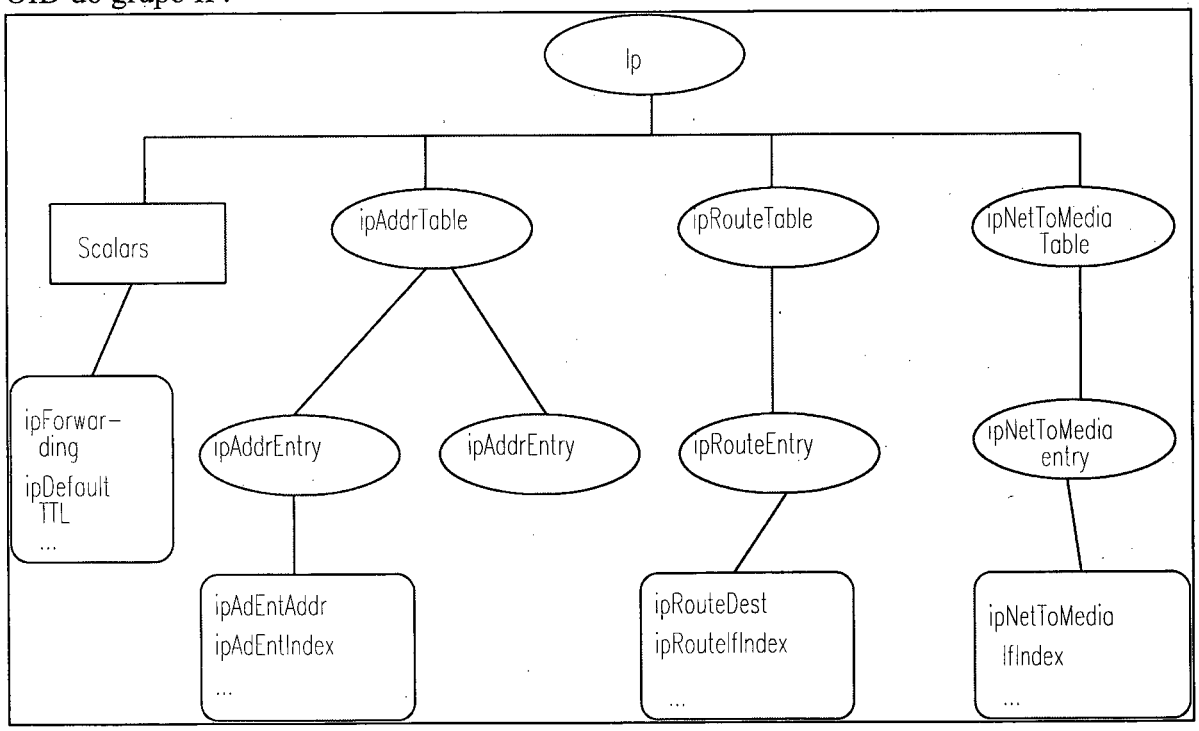


Figura 4.4 - MIB Internet mapeada

Assim, o gateway simulará as ações requisitadas pelo gerente SNMP para operar em vários objetos e, ao mesmo tempo, preservar a interface no lado OSI. Tal simulação garante a operação transparente do gerente OSI.

Um filtro especifica uma condição que deve ser satisfeita num objeto para que ele seja selecionado para realizar uma operação específica. A função *filtering* impõe restrições aos objetos selecionados pela função *scoping* para uma operação. Enquanto a função *scoping* usa a hierarquia de objetos na árvore de instâncias para selecionar um objeto, a função *filtering* usa o estado de uma instância de objeto.

Ao invés de implementar a função *filtering* no lado Internet, um gateway, ao receber uma requisição no lado OSI, armazena os filtros associados com a requisição. O gateway então aplica a função *scoping* para selecionar objetos nos quais os filtros especificados necessitam ser aplicados. Os objetos gerenciados selecionados são recuperados do agente SNMP especificado na operação requisitada. Uma vez que os objetos foram recuperados, o gateway aplica os filtros armazenados e executa a operação somente naqueles objetos cujo teste foi positivo para o(s) filtro(s) especificado(s).

O protocolo CMIP não tem restrição quanto ao tamanho do pacote. O protocolo SNMP usa UDP para implementar seus serviços. O protocolo UDP tem uma restrição quanto ao tamanho do pacote que é transmitido. Essa restrição adiciona complexidade à máquina de estados do gateway. Em casos onde uma simples requisição no lado OSI

traduz-se em várias requisições no lado SNMP, o gateway tem que manter o estado de cada requisição no lado OSI, coletar várias respostas do domínio Internet e formatá-las numa simples resposta. Em tal caso, um gateway retém os estados da requisição OSI e coleta as respostas do lado Internet. Uma vez que uma informação completa é obtida do lado Internet, ela é formatada e enviada como resposta para o lado OSI.

O serviço CMIS permite dois tipos de sincronização nos objetos: **atômica** e **melhor esforço**. Como descrito no capítulo 2, a sincronização atômica tem uma semântica todos para um, isto é, a operação é realizada com sucesso ou falha em todos os objetos selecionados. A técnica do melhor esforço tem por objetivo executar a operação em tantos objetos selecionados quanto possível. O protocolo SNMP suporta somente sincronização atômica. A sincronização atômica no lado OSI trata todos os objetos gerenciados como sendo parte de uma simples requisição. A técnica de sincronização do melhor esforço no lado OSI implica que cada requisição gerará requisições que tem um simples objeto contido nelas.

Baseado nas informações acima, um gateway tem que permitir somente sincronização atômica, quando o gerente requer atualização de múltiplos objetos na linha de uma tabela. Tal atualização atômica reflete o comportamento do protocolo SNMP e garante a consistência da consulta na tabela. Entretanto, se um gerente requer uma operação em vários objetos escalares, então o gateway pode implementá-las como sincronização atômica ou como sincronização do melhor esforço.

Capítulo 5. Proposta de um Gateway CMIP-SNMP

A ISO está desenvolvendo padrões para o gerenciamento de redes OSI. Sua abordagem orientada a objetos na identificação de entidades de gerenciamento e o CMIS/CMIP tem-se tornado uma estrutura conceitual importante para o surgimento de soluções de gerenciamento de redes. Do outro lado, existe a IAB (*International Activity Board*) com um conjunto de padrões para o gerenciamento de redes TCP/IP (*Transmission Control Protocol/Internet Protocol*), referenciados como SNMP. Devido a sua simplicidade, o protocolo SNMP é hoje implementado em produtos de praticamente todos os maiores fabricantes de equipamentos de comunicação. Além disso, redes Internet estão hoje, de fato, sendo gerenciadas usando o protocolo SNMP.

Como o protocolo SNMP tem sido largamente implementado, fez-se necessário solucionar o problema cada vez maior de integrar o gerenciamento de redes heterogêneas. Por isso, os conceitos OSI, que podem ser aplicados também para redes não-OSI, parecem ser os mais adequados. Sabe-se porém, que os recursos SNMP ainda necessitarão ser gerenciados no futuro. Por isso, a integração do gerenciamento OSI e Internet é fundamental.

O capítulo anterior introduz um paradigma que usa uma aproximação de um gateway para alcançar a integração entre os domínios OSI e Internet. Vários pontos chave necessários para alcançar tal integração são identificados e discutidos. Baseado nesses pontos chaves, o projeto básico de um gateway que procura alcançar a interoperabilidade usando o paradigma escolhido é sugerido.

5.1. Estrutura de Mapeamento CMIP-SNMP

Com a expansão dos conceitos de gerenciamento OSI surgiu a necessidade de, através de um gerente de rede OSI, gerenciar não apenas redes OSI, mas usar os conceitos OSI para gerenciar, também, redes heterogêneas (não-OSI). Entretanto, deseja-se que o gerente de rede OSI tenha uma visão integrada e completa da rede heterogênea. Dentre as redes não-OSI destaca-se a Internet e que é a utilizada na descrição deste trabalho. Para permitir o gerenciamento OSI é necessário oferecer uma visão dos recursos de acordo com o modelo OSI. Para isso, as variáveis SNMP têm que ser transformadas em objetos gerenciados CMIP. As classes de objetos gerenciados que resultarem estarão de acordo com as classes de objetos gerenciados Internet.

Um gateway CMIP-SNMP deve mapear a informação de gerenciamento do modelo de informação OSI para o modelo de informação Internet e vice-versa. Para isso, deve considerar dois aspectos considerados fundamentais para esta tradução, que são o mapeamento do nome e o mapeamento do serviço.

Para ilustrar, considere o processo de requisição de uma operação **M-Get** do protocolo CMIP num atributo de um objeto gerenciado que corresponde a um contador de valor no protocolo SNMP. Para trazer o contador de valor do agente SNMP o gateway executará as seguintes tarefas:

- Mapeamento do Nome: depois de receber o *M-Get-Indication*, o gateway tem de determinar, através do nome OSI especificado, qual informação de gerenciamento na MIB Internet é referenciada e, consequentemente, executar algum tipo de mapeamento do nome.

- Mapeamento do Serviço: outro problema que deve ser tratado pelo gateway surge das diferenças entre os protocolos de gerenciamento CMIP e SNMP, cujos serviços tem de ser mapeados. No exemplo, o M-Get será mapeado para o serviço Get correspondente no SNMP. Se uma cláusula de escopo, filtro ou sincronização estivesse sendo usada poderiam ser gerados mais de um Get no protocolo SNMP.

Assim sendo, o gateway deverá conter um módulo tradutor de PDUs e um outro módulo que controle as PDUs enviadas/recebidas para fornecer uma informação consistente ao gerente. O gateway funciona basicamente segundo o esquema mostrado na figura 5.1, ou seja, verifica se os comandos de gerenciamento são para um recurso OSI ou Internet. Se for para um recurso OSI, o gateway faz o estabelecimento da associação, utilizando os serviços fornecidos pelo ACSE, antes de enviar operações de gerenciamento, visto que o modelo é orientado a conexão, conforme descrito no capítulo 2. No caso do exemplo, as camadas abaixo do ROSE formam a arquitetura da Rede UFSC, que possui uma arquitetura Internet. Caso o comando de gerenciamento seja para um recurso SNMP, o gateway então, ao receber a primitiva solicitando o estabelecimento de conexão, enviará uma PDU com um comando Get para um objeto qualquer do recurso que se pretende gerenciar. Se retornar uma resposta, o gateway responde ao ACSE com o *A-Associate-Confirm*, caso ocorra *Time-out,* o gateway envia *A-Associate-Reject* ao ACSE. Desta forma, com a arquitetura proposta é possível fazer o gerenciamento de recursos SNMP através de uma plataforma de gerenciamento OSI.

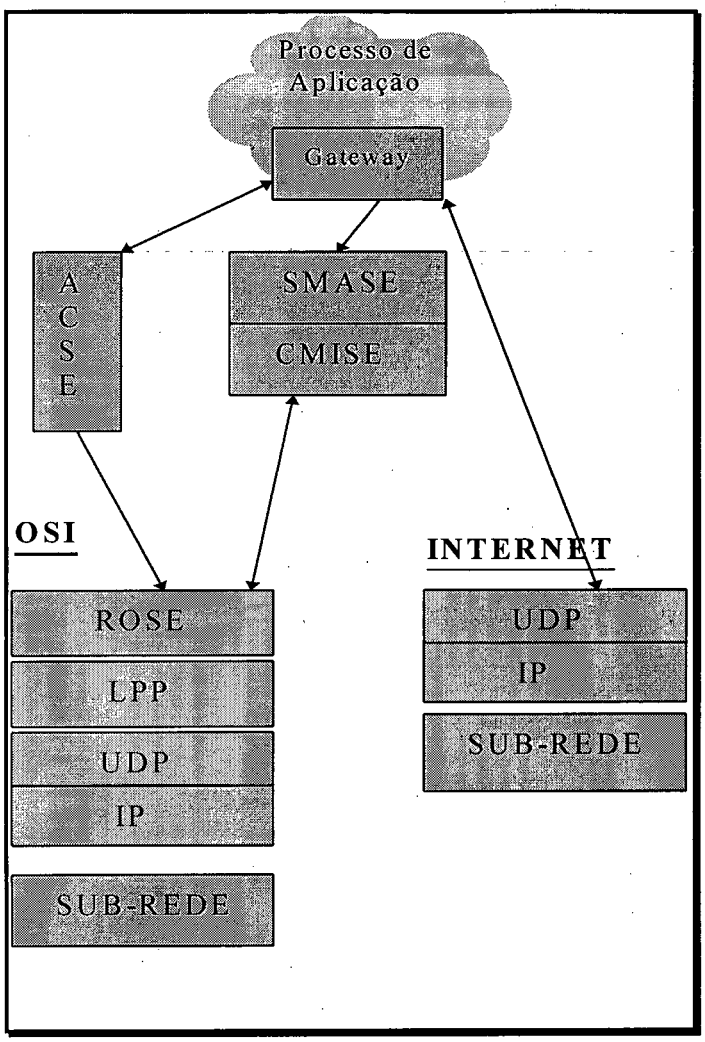


Figura 5.1 - Localização do Gateway na Plataforma de Gerenciamento

Para efetuar a tradução das PDUs é necessário conhecer o seu formato, tanto no modelo OSI quanto no modelo Internet. No anexo A encontram-se definidos os formatos, em ASN.1, das PDUs utilizadas pelos protocolos CMIP e SNMP.

5.2. Definição das operações fornecidas pelo Gateway

O gateway é mais um processo de aplicação que ficará à disposição do usuário para o gerenciamento dos recursos que compõem uma rede de computadores. Este gerenciamento é feito através de uma plataforma que, entre outros componentes, possui uma interface gráfica que mostra o estado da rede. Com esta interface, o gerente

seleciona um agente e, em seguida, define qual operação de gerenciamento deve ser executada, e em qual objeto daquele agente.

O processo gateway definido é uma aplicação que faz interface com o ACSE, o SMASE e o UDP (visto que a rede UFSC possui a estrutura definida na figura 5.1). No processo de aplicação estão definidas operações de gerenciamento que podem ser efetuadas sobre os objetos gerenciados.

A execução de uma operação de gerenciamento no ambiente OSI passa pelas seguintes fases:

- fase de estabelecimento de associação entre os processos gerente e agente;
- fase de troca e execução das operações e notificações de gerenciamento;
- fase de liberação de associação.

O processo de aplicação utiliza diretamente os serviços do ACSE para a fase de estabelecimento/liberação da associação. O protocolo CMIP especifica os elementos de protocolo que são usados para prover os serviços de operação e notificação contidos no CMISE. A CMIPM aceita as primitivas de pedido e resposta do serviço CMISE e emite CMIPDUs iniciando a transferência de operações/notificações de gerenciamento.

Como mostrado na figura 5.1, o gateway possui interface com o SMASE. Os serviços providos pelo SMASE são identificados através das unidades funcionais correspondentes às funções de gerenciamento mencionadas. O SMASE também identifica a informação de gerenciamento específica a ser trocada entre os processos de gerenciamento através de MAPDUs. Assim, o gateway envia uma operação de gerenciamento para o SMASE (Get ou Set, por exemplo), que se utiliza dos serviços PASS-THROUGH para mapear seus serviços sobre o CMISE. Em [Brisa93] encontra-se definido o mapeamento descrito.

Os procedimentos de protocolo somente indicam como interpretar os vários campos da CMIPDU e não indicam o que um usuário do serviço CMISE deve fazer com a informação recebida, nem como deve processá-la.

5.2.1. Estrutura proposta para o Gateway CMIP-SNMP

O processo de Aplicação Gateway possui dois módulos principais. O primeiro é o módulo **Tradutor de Pdus**, visto que a função principal do Gateway é permitir a integração entre modelos de gerenciamento diferentes, e a tradução de PDUs é uma tarefa fundamental para alcançar tal objetivo. O outro módulo é definido como **Unidade de Controle**. Este módulo permite que, ao enviar/receber operações de gerenciamento, o Gateway repasse ao gerente somente informações consistentes. Um exemplo disso é a execução de uma operação de gerenciamento com sincronização atômica. A unidade de controle verifica se todas as respostas, referentes a uma determinada requisição foram recebidas. Em caso afirmativo, estas respostas são repassadas ao gerente; caso contrário, o gateway deve retornar, no caso de uma operação Set não completada, a um estado anterior e informar um código de erro ao processo gerente. Além dos módulos apresentados, deve existir outra aplicação que defina a tradução das MIBs, atualizando a MIB OSI, a estrutura de identificação dos agentes e a estrutura de mapeamento.

Devido à limitação imposta pelo protocolo SNMP, que só permite a utilização das operações *GetRequest*, *GetNextRequest*, *SetRequest*, *GetResponse* e *Trap*, nem todos os serviços CMIS são possíveis de serem mapeados. A operação M-CREATE, por exemplo, não é mapeada porque não existe um parâmetro nesta operação que contenha o nome Internet do recurso, parâmetro esse que permitiria tal mapeamento. Além disso, o protocolo SNMP não suporta a criação de objetos dinamicamente. Da mesma forma as outras operações, com exceção do M-GET e M-SET, não têm contrapartida no modelo Internet. Apesar disso, a operação M-DELETE, que tem a função de apagar os objetos selecionados da MIB do agente e da visão da

MIB que o gerente possui, poderia ser mapeada de forma que o objeto fosse apagado somente da estrutura de mapeamento. Com isso, o recurso deixa de existir para a plataforma de gerenciamento OSI, sem entretanto, apagá-lo da MIB Internet, porque, como foi dito, o agente SNMP não permite a operação de exclusão de objetos.

A idéia inicial para o mapeamento da operação Delete era que bastava excluir o objeto da estrutura de mapeamento. Existe o problema que, se ele não fosse encontrado na tabela, a aplicação gateway interpretaria que o objeto pertence ao domínio OSI. Com isso, teríamos um situação de erro/inconsistência no modelo definido. Uma solução para o problema seria a criação de um campo **Estado** na estrutura de mapeamento. Assim, o objeto existe na estrutura, fazendo com que o gateway reconheça que o objeto é SNMP, e marcando esse campo com *, por exemplo, estaria definido que o objeto foi apagado.

Nesta situação, o objeto existe na visão do gerente e existe na MIB do agente. Apenas na estrutura de mapeamento é que está indicado que ele não existe, ou seja, foi apagado. Assim, o gerente pode ficar enviando requisições para este objeto a todo instante e sempre recebendo uma mensagem do gateway informando que o objeto não existe.

Além disso, o agente permanece atualizando os objetos da MIB. Se algum objeto for excluído da estrutura de mapeamento, mas não for da MIB do agente, o agente estará realizando um esforço em vão. Como a operação M-DELETE permite as funções *scoping*, *filtering* e *synchronization*, existe um esforço muito grande para a implementação de uma função que, na verdade, não excluiria o objeto da MIB do agente. Por isso, quando o gateway identificar que a operação Delete pretende atuar sobre um objeto SNMP, ele deve informar à aplicação de gerenciamento que não é possível realizar esta operação.

Além das operações do CMIS, o tradutor emula o estabelecimento de associação no lado Internet, visto que o protocolo SNMP não é baseado em conexão.

As funções implementadas pelo tradutor, que emulam as funcionalidades existentes no serviço CMIS são as seguintes:

- G-ASSOCIATE;
- G-GET;
- G-SET.

5.3. Descrição do Funcionamento das Atividades da Aplicação Gateway

Conforme já foi dito, receber solicitações da aplicação de gerenciamento Gerente (através de uma chamada de função anteriormente definida) e identificar se a operação de gerenciamento deve ser executada em um recurso OSI ou SNMP (através da pesquisa na estrutura de mapeamento) são as tarefas iniciais a serem executadas pelo Gateway.

5.3.1. Descrição da Fase de Estabelecimento de Associação efetuada pelo Gateway

O processo de aplicação Gerente, ao iniciar uma operação de gerenciamento, deve, primeiramente, estabelecer uma associação com o processo de aplicação Agente desejado. Isto ocorre através da chamada da função G-ASSOCIATE(), fornecida pelo Gateway. Se o gerente não quiser utilizar o gateway, porque tem certeza que não existem recursos SNMP na rede, então ele deve utilizar os elementos de serviço ACSE, ROSE, SMASE e CMISE diretamente; neste caso supõe-se a existência de um ambiente OSI completo.

A função G-ASSOCIATE() possui os mesmos parâmetros definidos para a função **ACSE-Association-Request**, implementada em [Andrea95].

Para identificar a origem do recurso, deve-se utilizar o conteúdo do parâmetro CAPT (*Called AP-Title* - título do processo de aplicação chamado), que conterá a identificação do agente com o qual se deseja estabelecer a associação. Deve existir uma estrutura (que na verdade funcionará como um serviço de diretório) que permitirá verificar se o processo agente chamado é OSI ou não.

A Aplicação Gateway possui uma tabela de identificação de agentes, formada por dois campos: identificação e endereço IP, conforme mostrado na tabela 5.1. O campo identificação possui os "nomes" dos agentes. É através deste campo, que servirá como chave primária de pesquisa, que o recurso é identificado. Na tabela, a coluna endereço IP contém o endereço IP do agente.

Identificação	**Endereço IP**
"Hub1"	150.162.1.2
"Hub2"	150.162.1.3

Tabela 5.1. Tabela de Identificação dos Agentes

Esta tabela contém apenas os nomes dos agentes SNMP. Isto porque, se o agente for OSI, o controle da operação é repassado para os elementos de serviço ACSE e ROSE.

Assim, ao receber um pedido de associação, a Aplicação Gateway pesquisa na tabela de identificação dos agentes a existência ou não do CAPT passado como parâmetro. Se ele não estiver na tabela, então o agente é OSI e o Gateway faz uma

chamada do serviço fornecido pelo ACSE (*request*), confirmando ou não a associação para o gerente. Se o CAPT estiver na tabela, então o agente é SNMP, e, neste caso, o Gateway deve verificar se o agente está ativo ou não. Para isso, deve enviar uma PDU Get, com o identificador de algum objeto presente naquele agente (*System*, por exemplo). Se ocorrer *time-out*, o gateway informa ao gerente, através do envio de uma PDU do ACSE, rejeitando o pedido de associação, senão deve confirmar a associação e, a partir disso, fica apto a receber operações de gerenciamento.

Algoritmo do G-ASSOCIATE

INÍCIO

1. **O gerente faz a chamada do G-ASSOCIATE passando os mesmos parâmetros que ACSE-ASSOCIATION.REQUEST;**

2. **O Gateway verifica através do parâmetro CAPT, se o agente é SNMP:**
   2.1. Se o CAPT estiver na tabela o agente é SNMP, caso contrário é OSI.

3. **Se o agente for OSI então:**
   3.1. Repassa pedido para o ACSE
   3.2. Aguarda resposta do ACSE
   3.3 Confirma ou rejeita associação para o gerente

4. **Senão, o agente é SNMP:**
   4.1. Envia PDU Get com identificador de algum objeto presente naquele agente
   4.2. Se o agente responder, confirma associação; senão (ocorre *time-out*) rejeita associação.

FIM

Uma vez estabelecida a associação, o processo de aplicação inicia a fase de emissão de operações de gerenciamento sobre os objetos gerenciados. O gateway permitirá a execução das seguintes operações de gerenciamento sobre um agente SNMP:

- G-Get();
- G-Set().

Após a execução das operações de gerenciamento, o Gateway deverá solicitar a liberação da associação. No caso do ambiente OSI basta utilizar os serviços do ACSE, e no caso do SNMP, como a associação foi apenas emulada, o gateway, ao receber o pedido de liberação da associação, deverá apenas atualizar seus mecanismos de controle, não permitindo o envio de operações de gerenciamento para aquele agente.

O anexo A apresenta a especificação das PDUs CMIP e SNMP em ASN.1. Dentre as PDUs SNMP existe a PDU TRAP que possibilita ao agente informar ao gerente de algumas condições previamente estabelecidas. A Aplicação Gateway ao receber Traps enviados pelos agentes SNMP deverá mapeá-los em notificações CMIP, definidas de acordo com [10165-4].

Para prover estas operações, o gateway manterá a seguinte estrutura:

Nome-OSI	**Nome-Internet**	**Nível**
Objeto1	Variável X	1
Atributo1	VariávelY	-1
Objeto2	VariávelZ	2

5.2 - Estrutura de Mapeamento

Nome-OSI: *OBJECT IDENTIFIER* do recurso no domínio OSI (ou seja, o identificador - DN - na MIB-OSI);

Nome-Internet: *OBJECT IDENTIFIER* do recurso no domínio Internet (ou seja, o identificador na MIB-Internet);
Nível: um número inteiro que permitirá a execução da função *Scoping*.

Esta estrutura contém a identificação de todos os recursos SNMP e seu correspondente identificador no ambiente OSI. Quando da inclusão de um recurso SNMP no ambiente de gerenciamento OSI, esta estrutura deve ser atualizada, ou seja, a operação CREATE deve, além de incluir o objeto na MIB-OSI, atualizar também a estrutura.

A seguir são descritos os procedimentos efetuados pelo Gateway quando da solicitação de execução de operações de gerenciamento sobre um recurso SNMP.

5.3.2. Descrição da operação GET

Em [Gisele93] encontram-se explicitados os parâmetros que cada uma das operações de gerenciamento possuem, bem como exemplos das cláusulas *scoping*, *filter* e *synchronization* nestas operações. Quando a PDU CMIP contiver a identificação de uma instância e um ou vários atributos, o processo Gateway deve procurar na estrutura de mapeamento o identificador da instância, informado através do parâmetro *Base Object Instance*. Ao encontrar, caso o recurso seja SNMP, verifica-se o conteúdo do parâmetro *Attribute Identifier List*. Se ele estiver vazio, então todos os atributos da classe serão mapeados para o formato SNMP. Caso contrário, somente os atributos presentes no parâmetro serão mapeados.

A grande vantagem do modelo OSI, conforme foi discutido no capítulo 2, está na existência das funções *scoping*, *filter* e *synchronization*, que não possuem

correspondência no modelo Internet. Entretanto, o Gateway permite a execução destas funções sobre agentes SNMP, emulando-as através da utilização das operações Get e Set, existentes no modelo SNMP.

O Gateway, ao identificar a existência de um filtro, além de efetuar a geração das PDUs, conforme descrito anteriormente, também deve aplicar tal filtro sobre as respostas que chegarem para ele. Assim, ao receber as respostas, o Gateway repassará para o processo gerente somente aquelas que satisfizerem a condição imposta pelo filtro. Esta verificação/aplicação do filtro sobre as PDUs é feita pela unidade de controle, módulo que, juntamente com tradutor de PDUs, compõem o gateway.

Como a estrutura utilizada para o mapeamento contém um campo NÍVEL, a função *Scoping* pode ser aplicada em tempo de mapeamento de PDUs e não após o mapeamento, como na função *Filter*. Para isso, basta verificar se o(s) objeto(s) definido(s) no parâmetro *Object Instance* e *Attribute List*, está(ão) dentro do escopo especificado no parâmetro *Scope*. Assim, somente os objetos que estão dentro do escopo definido serão selecionados para a execução da operação de gerenciamento solicitada.

A execução da função Sincronização é parecida com a da função Filtro. Caso seja definida a sincronização Melhor Esforço no parâmetro *Synchronization*, o Gateway poderá informar ao processo gerente uma resposta, independente se esta resposta está completa ou não. No caso da sincronização atômica, a unidade de controle somente repassará ao gerente uma resposta, se esta for completa. Caso isto não seja possível, um erro é retornado.

Algoritmo da operação de gerenciamento G-Get

INÍCIO

1. O gerente faz a chamada da função G-Get passando os mesmos parâmetros que o M-Get definido em [ISO9595].

2. O gateway identifica, através do parâmetro *Base Object Instance*, se o objeto no qual se deseja executar uma operação de gerenciamento é SNMP.

2.1. Se o objeto estiver presente na estrutura de mapeamento, o objeto é SNMP, caso contrário é OSI.

3. Se o objeto for OSI então:

3.1. Faz a chamada do serviço M-GET definido pelo modelo OSI;

3.2. Repassa ao gerente a resposta recebida.

4. Senão, o objeto é SNMP e:

4.1. Efetua a tradução, gerando PDUs SNMP dos atributos definidos no parâmetro *Attribute Identifier List*;

4.2. Aplica o escopo definido no parâmetro *Scope*;

4.3. Envia a(s) PDU(s) para o agente associado;

4.4. Recebe as respostas enviadas pelo agente;

4.5. Aplica o filtro definido no parâmetro *Filter,* bem como a sincronização definida no parâmetro *Synchronization.*

4.6. Monta a resposta no formato CMIP e a envia para o gerente.

FIM

5.3.3. Descrição da operação SET

A Aplicação Gateway efetua procedimentos semelhantes àqueles utilizados na operação G-GET. Primeiramente, procura o objeto base na tabela de mapeamento. Se não for encontrado repassa a operação para o SMASE. Caso contrário, realiza o mapeamento do nome para o objeto base e seus atributos, de acordo com o parâmetro *Modification List*. Ou seja, serão mapeados apenas aqueles atributos que estiverem contidos neste parâmetro. Neste caso, só será possível a operação *Replace*. As outras variações da operação SET, *add values*, *remove values* e *set to default*, permitidas pelo modelo OSI, não possuem contrapartida no protocolo SNMP. Portanto, a interface da aplicação de gerenciamento não deve permitir que o usuário selecione esse tipo de operação.

Caso o parâmetro *scoping* tenha sido especificado, o tradutor realiza o mapeamento do nome para cada um dos objetos selecionados por este parâmetro, da mesma forma como no G-GET. Cada objeto do escopo corresponde a uma PDU *SetRequest* do protocolo SNMP. O campo *variable-bindings* é composto pelos atributos mapeados e os seus respectivos valores, definidos no parâmetro *Modification list*.

A sincronização é emulada da mesma forma que no G-GET. Quando uma PDU *SetRequest* chega ao processo que se comunica com o agente SNMP, este verifica o tipo de sincronização requerida. Se a sincronização for atômica, a PDU é repassada para o agente. Se todas as variáveis puderem ser modificadas pelos valores especificados, um *GetResponse* é retornado para o Gateway, das mesma forma que foi enviado o *SetRequest*. Caso contrário, se pelo menos uma variável não permitir a atualização requerida, nenhuma delas é modificada e o campo *error-status* e *error-index*, na PDU de resposta, indicarão o erro que ocorreu. Quando a sincronização é a de melhor esforço, o processo receptor da PDU *SetRequest*, quebra a PDU em várias partes, uma para cada variável do campo *variable-bindings*. Assim, se a variável não

puder ser modificada pelo agente, este processo insere no lugar do valor requerido para a modificação, o valor NULL. Depois de receber todas as respostas, por parte do agente, uma PDU *GetResponse* é montada para ser enviada ao Gateway. Dessa forma, o Gateway, no caso de sincronização de melhor esforço, consegue identificar quais as variáveis que foram modificadas e quais não foram. Aquelas que tiverem seu valor no campo *variable-bindings* igual a NULL, não foram atualizadas e, aquelas que retornaram com o mesmo valor especificado na requisição, tiveram a operação de gerenciamento executada com sucesso.

O filtro também é mapeado como na operação G-GET; primeiro é enviado um *GetRequest* com os atributos contidos no parâmetro *filter* mapeados. Depois que a resposta retornar ao Gateway, é verificado o filtro para cada PDU de resposta (caso em que o escopo seleciona mais de um objeto). Para aqueles que a condição de filtro satisfizer, é enviado um *SetRequest* com os atributos do parâmetro *Attribute identifier list* mapeados compondo o campo *variable-bindings*.

Algoritmo da operação de gerenciamento G-Set

INÍCIO

1. O gerente faz a chamada da função G-Set passando os mesmos parâmetros que o M-Set definido em [ISO9595].

2. O gateway identifica, através do parâmetro *Base Object Instance*, se o objeto no qual se deseja executar uma operação de gerenciamento, é SNMP.

2.1. Se o objeto estiver presente na estrutura de mapeamento, o objeto é SNMP, caso contrário é OSI.

3. Se o objeto for OSI então:

3.1. Faz a chamada do serviço M-SET definido pelo modelo OSI;

3.2. Repassa ao gerente a resposta recebida.

4. Senão, o objeto é SNMP e:

4.1. Efetua a tradução, gerando PDUs SNMP dos atributos definidos no parâmetro *Modification List*;

4.2. Aplica o escopo definido no parâmetro *Scope*;

4.3. Envia a(s) PDU(s) para o agente associado;

4.4. Recebe as respostas enviadas pelo agente;

4.5. Aplica o filtro definido no parâmetro *Filter,* bem como a sincronização definida no parâmetro *Synchronization.* Neste caso, é necessário obter-se os valores dos atributos através da operação Get, para efetuar a operação Set somente naqueles atributos que satisfizerem a condição imposta pelo filtro. No caso da sincronização o funcionamento é semelhante. O Gateway deve obter os valores dos atributos/variáveis, caso seja necessário retornar ao estado anterior, se no caso de uma sincronização atômica ocorrer algum problema.

4.6. Monta a resposta no formato CMIP e a envia para o gerente.

FIM

5.3.4. Descrição do comportamento do Gateway quanto aos Traps

Conforme já foi discutido, uma das operações que podem ser recebidas pelo Gateway é o Trap. Assim, um algoritmo que descreve o comportamento do Gateway ao receber tal operação é:

Algoritmo da operação de gerenciamento Trap

INÍCIO

1. Receber a PDU Trap do agente SNMP, no formato definido em [RFC1157];

2. Mapear a PDU Trap recebida do agente SNMP em uma notificação CMIP definida de acordo com [10165-4].

3. Enviar esta notificação para o gerente OSI.

FIM

Uma vez que o número de Traps definido no protocolo SNMP é limitado (existem sete Traps), o gateway utilizará uma classe de notificação definida como NotificaçãoTrap, que servirá para indicar ao gerente que se trata de uma informação provinda do ambiente SNMP. Um exemplo desta notificação pode ser:

```
NotificaçãoTrap NOTIFICATION
BEHAVIOUR notificationTrapBehaviour
WITH INFORMATION SYNTAX  Notification-ASN1Module
AND ATTRIBUTE IDS tipoTRAP TipoTrap
REGISTERED AS {Smi2Notification Exemplo}
notificationTrapBehaviour BEHAVIOUR
        DEFINED AS "Esse tipo de notificação é usado para relatar a chegada de um
        TRAP na aplicação Gateway."
```

6. Considerações Finais

Atualmente a maioria dos produtos de gerenciamento de redes são implementados sobre a arquitetura Internet, utilizando o protocolo SNMP. Isto ocorreu devido a sua simplicidade e facilidade de implementação. Esta simplicidade ocorre porque a tarefa do gerenciamento de rede completo, ou seja, permitindo acessar informações fornecidas pelas cinco áreas funcionais definidas pela ISO, não é trivial. Os agentes do modelo Internet não fornecem a funcionalidade necessária para o gerenciamento das áreas funcionais definidas pela ISO. Para suprir tais necessidades de gerenciamento, ou se implementam os agentes com a funcionalidade necessária, ou se obtém a integração entre modelos de gerenciamento diferentes através de uma Aplicação Gateway. Este trabalho teve por objetivo propor um modelo de Gateway, que permitirá a integração entre os dois modelos de gerenciamento mais difundidos, que são o modelo OSI e o modelo Internet de gerenciamento de redes.

Durante o desenvolvimento deste trabalho procurou-se conhecer as principais características dos dois modelos de gerenciamento mais difundidos atualmente para verificar quais eram as suas principais diferenças. Feito isso, identificou-se três propostas de funcionamento da Aplicação Gateway. Na primeira concentrava-se toda a complexidade do gateway em um nó, e utilizava-se uma linguagem chamada de padrão, através da qual os comandos passavam. Esta solução causava um *overhead* muito grande, visto que os comandos eram traduzidos deste formato padrão para o formato dos modelos de gerenciamento, para então serem executados. Em uma segunda solução pensou-se na duplicação das MIBs. Toda MIB Internet seria traduzida para a MIB OSI e em cada nó estariam presentes os dois agentes (CMIP e SNMP). Neste paradigma, além do trabalho de tradução das MIBs, existe também a necessidade de uma versão da MIB e do agente em cada nó para cada tipo de rede (com seu modelo de gerenciamento) que fosse interconectada. Por fim, estudou-se o paradigma da Aplicação Gateway, onde as requisições do gerente OSI são encaminhadas ao gateway que traduz para o modelo Internet, encaminha ao agente SNMP, e ao receber a resposta, envia ao gerente no formato OSI. Esta tradução leva em consideração dois aspectos fundamentais que são o mapeamento funcional e o mapeamento dos nomes. Como os dois modelos possuem

diferenças no modelo funcional e no esquema de nomeação, a maneira como as PDUs devem ser traduzidas é de fundamental importância.

Devido a heterogeneidade das redes de computadores está havendo um grande esforço de fabricantes e pesquisadores no sentido de desenvolver soluções que possibilitem formas de integração entre equipamentos e ferramentas automatizadas. A Aplicação Gateway proposta neste trabalho é uma destas soluções de integração entre modelos de gerenciamento que seguem filosofias diferentes. Esta dissertação procurou fazer o levantamento dos requisitos funcionais necessários para o desenvolvimento de uma Aplicação Gateway entre os protocolos CMIP e SNMP. Entretanto, esses requisitos foram especificados se uma maneira informal. Durante a definição desses requisitos, um trabalho de grande valia foi a sua implementação definida em [Gisele96]. Neste trabalho, conseguiu-se verificar inconsistências nos requisitos definidos. O ideal seria a definição dos requisitos, com a utilização de técnicas de descrição formal, que permitiriam a validação e verificação da especificação definida neste trabalho. Por ser considerado um ponto fundamental, este trabalho começou a ser feito e é tema de outra dissertação de mestrado.

Além da solução proposta neste trabalho existem outras tais como, o desenvolvimento de agentes CMIP para gerenciar recursos SNMP, discutida no capítulo 4. Existe também outra tendência que é o aprimoramento do protocolo SNMP, exemplo disso é a versão 2 deste protocolo, que adicionou algumas das funcionalidades definidas no modelo OSI. A partir do momento que os dois protocolos apresentam características mais semelhantes a integração entre eles se torna mais trivial.

Uma outra característica que pode ser adicionada ao Gateway é a funcionalidade de segurança. Pode-se inserir procedimentos no Gateway que possibilitem o controle de acesso às suas funções.

Baseado no estudo descrito, espera-se que com a utilização de uma plataforma de suporte de gerenciamento OSI, e mais a Aplicação Gateway proposta neste trabalho, seja possível gerenciar não somente recursos OSI, mas também recursos não-OSI, que hoje são a maioria no mercado. Ou seja, ter-se-ia o gerenciamento integrado de recursos que seguem filosofias de gerenciamento diferentes.

7. Referências Bibliográficas

[10164-1] - ISO/IEC DIS 10164-1 - *Information Technology - Open Systems Interconnection - System Management - Part 1: Object Management Function, International Organization for Standardization/International Electrotecnical Commission, March 1991.*

[10164-2] - ISO/IEC DIS 10164-2 - *Information Technology - Open Systems Interconnection - System Management - Part 2: State Management Function, International Organization for Standardization/International Electrotecnical Commission, March 1991.*

[10164-3] - ISO/IEC DIS 10164-3 - *Information Technology - Open Systems Interconnection - System Management - Part 3: Attributes for Representing Relationships, International Organization for Standardization/International Electrotecnical Commission, March 1991.*

[10164-4] - ISO/IEC DIS 10164-4 - *Information Technology - Open Systems Interconnection - System Management - Part 4: Alarm Reporting Function, International Organization for Standardization/International Electrotecnical Commission, March 1991.*

[10164-5] - *ISO/IEC DIS 10164-5 - Information Technology - Open Systems Interconnection - System Management - Part 5: Event Report Management Function, International Organization for Standardization/International Electrotecnical Commission, March 1991.*

[10164-6] - ISO/IEC DIS 10164-6 - *Information Technology - Open Systems Interconnection - System Management - Part 6: Log Control Function, International Organization for Standardization/International Electrotecnical Commission, March 1991.*

[10164-7] - ISO/IEC DIS 10164-7 - *Information Technology - Open Systems Interconnection - System Management - Part 7: Security Alarm Reporting Function, International Organization for Standardization/International Electrotecnical Commission, March 1991.*

[10164-8] - ISO/IEC DIS 10164-8 - *Information Technology - Open Systems Interconnection - System Management - Part 8: Security Audit Trail Function, International Organization for Standardization/International Electrotecnical Commission, March 1992.*

[10164-9] - ISO/IEC DIS 10164-9 - *Information Technology - Open Systems Interconnection - System Management - Part 9: Objects and Attributes for Access Control [for CCITT Applications], International Organization for Standardization/International Electrotecnical Commission, October 1991.*

[10164-10] - ISO/IEC DIS 10164-10 - *Information Technology - Open Systems Interconnection - System Management - Part 10: Accounting Meter Function, International Organization for Standardization/International Electrotecnical Commission, January 1991.*

[10164-11] - ISO/IEC DIS 10164-11 - *Information Technology - Open Systems Interconnection - System Management - Part 11: Worload Monitoring Function, International Organization for Standardization/International Electrotecnical Commission, October 1990.*

[10164-12] - ISO/IEC DIS 10164-12 - *Information Technology - Open Systems Interconnection - System Management - Part 12: Test Management Function, International Organization for Standardization/International Electrotecnical Commission, July 1991.*

[10164-13] - ISO/IEC DIS 10164-13 - *Information Technology - Open Systems Interconnection - System Management - Part 13: Summarization Function [for CCITT Applications], International Organization for Standardization/International Electrotecnical Commission, July 1991.*

[10165-1] ISO/IEC DIS 10165-1, *Information Technology - Open System Interconnection- Structure of Management Information - Part 1: Management Information Model, International Organization for Standardization / International Electrotecnical Commission , March 1991.*

[10165-2] ISO/IEC DIS 10165-2, *Information Technology - Open System Interconnection- Structure of Management Information - Part 2: Definition of Management Information, International Organization for Standardization / International Electrotecnical Commission , March 1991.*

[10165-3] ISO/IEC DIS 10165-3, *Information Technology - Open System Interconnection- Structure of Management Information - Part 3: Definition of Management Attributes, International Organization for Standardization / International Electrotecnical Commission , May 1989.*

[10165-4] ISO/IEC DIS 10165-4, *Information Technology - Open System Interconnection - Structure of Management Information - Part 4: Guidelines for the definition of managed objects, International Organization for Standardization / International Electrotecnical Commission , March 1991.*

[Andrea95] - BORGES, Andrea Luiza V. M. "Especificação e Implementação de Elementos de Serviço do modelo OSI - ACSE e ROSE". Projeto de conclusão do curso de Ciências da Computação. UFSC - Departamento de Informática e de Estatística. Dezembro/95.

[Bean93] - BEAN, Angelo et alli. *"Specifying Goal-Oriented Network Management Systems". IEEE Communications Magazine. May 1993.*

[Brisa93] - CARVALHO, Tereza Cristina Melo de Brito et alli. Gerenciamento de Redes - Uma Abordagem de Sistemas Abertos. BRISA (Sociedade Brasileira para Interconexão de Sistemas Abertos) Makron Books, 1993.

[Gering93] - GERING, Michael. *"CMIP versus SNMP". Integrated Network Management - III, Elsevier Science Publishers, 1993.*

[Gisele96] - SILVA, Gisele Dircksen da. "Um Tradutor CMIP-SNMP". Projeto de conclusão do curso de Ciências da Computação. UFSC - Departamento de Informática e de Estatística. Agosto/96.

[Hayes93] - HAYES, Stephen. *"Analyzing Network Performance Management". IEEE Communications Magazine. May 1993.*

[Hunter92] - HUNTER, H. Bruce et alli. *"Surveying Far-Flung Networks". Byte - August 1992.*

[ISO8824] ISO/IEC 8824, *Information Processing - Open System Interconnection-Specification of Abstract Syntax Notation One (ANS.1), April 1990.*

[ISO9595] - *Recommendation X.710 and ISO/IEC 9595, Information Technology - Open Systems Interconnection - Common Management Information Service Definition for CCITT Applications, 1991.*

[ISO9596] - *Recommendation X.711 and ISO/IEC 9596, Information Technology - Open Systems Interconnection - Common Management Information Protocol Specification for CCITT Applications, 1991.*

[Nance92] - NANCE, Barry. *"Netware grows lean, not mean". Byte - March 92.*

[Kalyan93] - KALYANASUNDARAM, Pramod e SETHI, Adarshpal S. *" An Application Gateway Design for OSI-Internet Management". Integrated Network Management - III, Elsevier Science Publishers*, 1993.

[Oliveira96] - OLIVEIRA, Alexandre V. et alli. "Definição de um Gateway CMIP-SNMP". Anais VI Semana de Informática da UFBA. 1996.

[Pyle93] - PYLE, H. Raymond. *"OSI Network Management Systems". IEEE Communications Magazine. May 1993.*

[Raman93] - RAMAN, Lakshmi. *"CMISE Functions and Services". IEEE Communications Magazine. May 1993.*

[RFC1095] IAB - *Common Management Information Services and Protocol over TCP/IP: CMOT, RFC 1095, April 1989.*

[RFC1155] IAB - *Structure and Identification of Management Information for TCP/IP - based Internets, RFC 1155, May 1990.*

[RFC1157] IAB - *Simple Network Management Protocol (SNMP), RFC 1157, May 1990.*

[RFC1213] IAB - *Management Information Base for Network Management of TCP/IP: MIB II, RFC 1213, March 1991.*

[RFC1214] IAB - *OSI Internet Management: Management Information Base, RFC 1214, April 1991.*

[Stallings90] - STALLINGS, William. *"Managing the Well-Tempered LAN". Byte - April 90.*

[Subrata93] - MAZUMDAR, Subrata, BRADY Stephen, and LEVINE, David W. *" Design of Protocol Independent Management Agent to Support SNMP and CMIP Queries". Integrated Network Management - III, Elsevier Science Publishers, 1993.*

[Thiry94] - THIRY, Marcelo Comicholi da Costa. Uma plataforma de suporte ao gerenciamento da rede UFSC. Anais do 12º Simpósio Brasileiro de Redes de Computadores. Curitiba, maio 1994.

[Udell92] - UDELL, Jo. *"Trends in Network Management". Byte - October 92.*

[Yemini93] - YEMINI, Yechiam. *"The OSI Network Management Model". IEEE Communications Magazine. May 1993.*

Anexo A. Definição das PDUs CMIP e SNMP em ASN.1

M-Get OPERATION

ARGUMENT **GetArgument**

RESULT **GetResult**

ERRORS (**accessDenied, classInstanceConflict, complexityLimitation, getListError, invalidFilter, invalidScope, noSuchObjectClass, noSuchObjectInstance, operationCancelled, processingFailure, syncNotSupported**)

LINKED (**m-Linked-Reply**)

::= **localValue 3**

GetArgument ::= *SEQUENCE* (*COMPONENTS OF* BaseManagedObjectId;

accessControl	[5] AccessControl *OPTIONAL*,
synchronization	[6] *IMPLICIT* CMISSync *DEFAULT* bestEffort,
scope	[7] Scope *DEFAULT* baseObject,
filter	CMISFilter *DEFAULT* and {},
attributeIdList	[12] *IMPLICIT SET OF* AttributeId *OPTIONAL*)

GetResult::= *SEQUENCE*{managedObjectClass ObjectClass *OPTIONAL*,

managedObjectInstance ObjectInstance *OPTIONAL*,

currentTime [5] *IMPLICIT* GeneralizedTime *OPTIONAL*,

attributeList [6] *IMPLICIT SET OF* Attribute *OPTIONAL*}

Attribute::= *SEQUENCE* { attributeId **AttributeId**,

attributeValue *ANY DEFINED BY* **AttributeId**}

AttributeId ::= *CHOICE* (globalForm [0]*IMPLICIT OBJECT IDENTIFIER*,

localForm [1] *IMPLICIT INTEGER*)

AttributeIdError ::= *SEQUENCE*

{errorStatus *ENUMERATED* { acessDenied (2),
noSuchAttribute (5) }
attributeId **AttributeId** }

BaseManagedObjectId ::= *SEQUENCE* (baseManagedObjectClass **ObjectClass**,
baseManagedObjectInstance **ObjectIntance**)

CMISFilter ::= *CHOICE* (item [8] **FilterItem**,
and [9] *IMPLICIT SET OF* **CMISFilter**,
or [10] *IMPLICIT SET OF* **CMISFilter**,
not [11] **CMISFilter**)

CMISSync ::= *ENUMERATED* (bestEffort (0),
atomic (1))

FilterItem ::= *CHOICE* (
equality [0] *IMPLICIT* **Attribute**,
substrings [1] *IMPLICIT SEQUENCE OF CHOICE* {
initialString [0] *IMPLICIT SEQUENCE* {
attributeId **AttributeId**,
string *ANY DEFINED BY* **attributeId**),
anyString [1] *IMPLICIT SEQUENCE* {
attributeId **AttributeId**,
string *ANY DEFINED BY* **attributeId**),
finalString [2] *IMPLICIT SEQUENCE* {
attributeId **AttributeId**,
string *ANY DEFINED BY* **attributeId**),
greaterOrEqual [2] *IMPLICIT* **Attribute** - declarar valor >= 2
lessOrEqual [3] *IMPLICIT* **Attribute** - declarar valor <=2
present [4] **AttributeId**

subsetOf [5] *IMPLICIT* **Attribute**

supersetOf [6] *IMPLICIT* **Attribute**

nonNullSetIntersection [7] *IMPLICIT* **Attribute** }

Scope:=*CHOICE* (INTEGER{ baseObject (0),
firstLevelOnly (1),
wholeSubtree (2) },
individualLevels [1] *IMPLICIT* INTEGER
basetoNhLevel [2] *IMPLICIT* INTEGER

accessDenied ERROR
::=localValue 2

classInstanceConflict ERROR
PARAMETER BaseManagedObjectId
::=localValue 19

complexityLimitation ERROR
PARAMETER ComplexityLimitation (opcional)
::=localValue 20

ComplexityLimitation ::= *SET* (scope [0] Scope *OPTIONAL*,
filter [1] CMISFilter *OPTIONAL*,
sync [2] CMISSync *OPTIONAL*)

duplicateManagedObjectInstance ERROR
PARAMETER ObjectInstance
::= localValue 11

getListError ERROR

PARAMETER GetListError

::=localValue 7

invalidAttributeValue ERROR

PARAMETER Attribute

::= localValue 6

GetListError::= *SEQUENCE* { managedObjectClass **ObjectClass** *OPTIONAL*,
managedObjectInstance **ObjectInstance** *OPTIONAL*,
currentTime [5] *IMPLICIT* GeneralizedTime *OPTIONAL*,
getInfoList [6] *IMPLICIT SET OF* **GetInfoStatus** }

GetInfoStatus ::= *CHOICE* {attributeIdError [0] *IMPLICIT* **AttributeIdError**,
attribute [1] *IMPLICIT* **Attribute** }

ObjectClass::= *CHOICE* (globalForm [0] *IMPLICIT OBJECT IDENTIFIER*,
localForm [1] *IMPLICIT* INTEGER)

ObjectInstance ::= *CHOICE*{distinguishedName [2] *IMPLICIT* **DistinguishedName**,
nonSpecificForm [3] *IMPLICIT OCTET* STRING
localDistinguishedName [4] *IMPLICIT* **RDNSequence** }

invalidFilter ERROR

PARAMETER CMISFilter

::=localValue 4

invalidObjectInstance ERROR

PARAMETER ObjectInstance

::=localValue 17

invalidScope ERROR

PARAMETER Scope

::=localValue 16

missingAttributeValue ERROR

PARAMETER SET OF AttributeId

::= localValue 18

noSuchAttribute ERROR

PARAMETER AttributeId

::= localValue 5

noSuchObjectClass ERROR

PARAMETER ObjectClass

::=localValue 0

noSuchObjectInstance ERROR

PARAMETER ObjectInstance

::=localValue 1

noSuchReferenceObject ERROR

PARAMETER ObjectInstance

::= localValue 12

operationCancelled ERROR

::=localValue 23

processingFailure ERROR

PARAMETER ProcessingFailure (*opcional*)

::=localValue 10

ProcessingFailure ::= *SEQUENCE* {managedObjectClass**ObjectClass**,
managedObjectInstance **ObjectInstance** *OPTIONAL*
specificErrorInfo [5] **SpecificErrorInfo** }

SpecificErrorInfo ::= *SEQUENCE* { errorId *OBJECT IDENTIFIER*,
errorInfo *ANY DEFINED BY* **errorId** }

syncNotSupported ERROR
PARAMETER CMISSync
::=localValue 3

m-Linked-Reply OPERATION
ARGUMENT LinkedReplyArgument
::=localValue 2

LinkedReplyArgument::=*CHOICE*(getResult [0] *IMPLICIT* **GetResult**,
getListError [1] *IMPLICIT* **GetListError**,
setResult [2] *IMPLICIT* **SetResult**,
setListError [3] *IMPLICIT* **SetListError**,
actionResult [4] *IMPLICIT* **ActionResult**,
processingFailure [5] *IMPLICIT* **ProcessingFailure**,
deleteResult [6] *IMPLICIT* **DeleteResult**,
actionError [7] *IMPLICIT* **ActionError**,
deleteError [8] *IMPLICIT* **DeleteError**)

M-Set OPERATION
ARGUMENT SetArgument
::= localValue 4

M-Set-Confirmed OPERATION

ARGUMENT SetArgument

RESULT SetResult

ERRORS **(accessDenied, classInstanceConflict, complexityLimitation, invalidFilter, invalidScope, noSuchObjectClass, noSuchObjectInstance, processingFailure, setListError, syncNotSupported)**

LINKED **(m-Linked-Reply)**

::= **localValue 5**

SetArgument::= *SEQUENCE* { *COMPONENTS OF* **BaseManagedObjectID**,
accessControl [5] **AccessControl** *OPTIONAL*,
synchronization [6] *IMPLICIT* **CMISSync** *DEFAULT* bestEffort,
scope [7] Scope *DEFAULT* **baseObject**,
filter **CMISFilter** *DEFAULT* and {},
modificationList [12] *IMPLICIT SET OF SEQUENCE* {
modifyOperator [2] *IMPLICIT* **ModifyOperator** *DEFAULT* replace,
attributeId **AttributeId**,
attributeValue *ANY DEFINED BY* **attributeId** *OPTIONAL*
(ausente para setToDefault)

SetResult ::= *SEQUENCE* { managedObjectClass **ObjectClass** *OPTIONAL*,
managedObjectInstance **ObjectInstance** *OPTIONAL*,
currentTime [5] *IMPLICIT* GeneralizedTime *OPTIONAL*,
attributeList [6] *IMPLICIT SET OF* **Attribute** *OPTIONAL*}

ModifyOperator ::= *INTEGER* {replace (0),
addValues (1),
removeValues (2),
setToDefault (3)}

SetListError::= *SEQUENCE* {managedObjectClass **ObjectClass** *OPTIONAL*,
managedObjectInstance **ObjectInstance** *OPTIONAL*,
currentTime [5] *IMPLICIT* **GeneralizedTime** *OPTIONAL*,
getInfoList [6] *IMPLICIT SET OF* **GetInfoStatus** }

SetInfoStatus ::= *CHOICE* {attributeIdError [0] *IMPLICIT* **AttributeIdError**,
attribute [1] *IMPLICIT* **Attribute** }

M-Create OPERATION
ARGUMENT **CreateArgument**
RESULT **CreateResult**
ERRORS **(accessDenied, classInstanceConflict, duplicateManagedObjectInstance, invalidAttributeValue, invalidObjectInstance, missingAttributeValue, noSuchAttribute, noSuchObjectClass, noSuchObjectInstance, noSuchReferenceObject, processingFailure)**
::= **localValue 8**

CreateArgument ::= *SEQUENCE* (
managedObjectClass **ObjectClass**;
CHOICE{managedObjectInstance **ObjectInstance**,
superiorObjectInstance [8] **ObjectInstance**}*OPTIONAL*
accessControl [5] **AccessControl** *OPTIONAL*,
referenceObjectInstance[6] **ObjectInstance** *OPTIONAL*,
attributeIdList [7] *IMPLICIT SET OF* **AttributeId** *OPTIONAL*
})

CreateResult ::= *SEQUENCE* {managedObjectClass **ObjectClass** *OPTIONAL*,
managedObjectInstance **ObjectInstance** *OPTIONAL*,
currentTime [5] *IMPLICIT* **GeneralizedTime** *OPTIONAL*,
attributeList [6] *IMPLICIT SET OF* **Attribute** *OPTIONAL*}

M-Delete OPERATION

ARGUMENT **DeleteArgument**

RESULT **DeleteResult**

ERRORS **(accessDenied, classInstanceConflict, complexityLimitation, invalidFilter, invalidScope, noSuchObjectClass, noSuchObjectInstance, processingFailure, syncNotSupported)**

LINKED **m-Linked-Reply**

::= **localValue 9**

DeleteArgument ::=*SEQUENCE* (*COMPONENTS OF* **BaseManagedObjectId**;
accessControl [5] **AccessControl** *OPTIONAL*,
synchronization [6] *IMPLICIT* **CMISSync** *DEFAULT* bestEffort,
scope [7] **Scope** *DEFAULT* **baseObject**,
filter **CMISFilter** *DEFAULT* **and {})**

DeleteResult ::= *SEQUENCE* { managedObjectClass **ObjectClass** *OPTIONAL*,
managedObjectInstance **ObjectInstance** *OPTIONAL*,
currentTime [5] *IMPLICIT* GeneralizedTime *OPTIONAL*}

Formato da PDU CMIP

TIPO DE OPERAÇÃO

Inteiro (Get=3, Set=4)

GET ARGUMENT

[0] Identificador Global

[1] Identificador Local

[2] DN

[3] Octet String

[4] LDN

[5] AcessControl

[6] Sincronização { Melhor Esforço(0) ou Atômica(1) }

[7] Escopo { [0] Inteiro (0 - **Objeto Base**

1 - **PrimeiroNívelSomente**

2 - **Toda subárvore)**

[1] Níveis Individuais

[2] BaseNthLevel

[8] **Filtro** { [0] **Igualdade** *SEQUENCE* { *CHOICE* {[0] **Object Identifier**(global)

[1] **Integer** (local) }

ANY DEFINED BY Acima }

[1] **Substring** {*CHOICE*{[0] **initialString** *SEQUENCE* {

CHOICE {[0] **ObjectIdentifier**(global)

[1] **Integer**(local)}

ANY DEFINED BY Acima}}

[1] **anyString** *SEQUENCE*{

CHOICE {[0] **ObjectIdentifier**(global)

[1] **Integer**(local)}

ANY DEFINED BY Acima}}

[2] **finalString** *SEQUENCE* {

CHOICE {[0] **ObjectIdentifier**(global)

[1] **Integer**(local)}

ANY DEFINED BY Acima}}

[2] >=**2** *SEQUENCE* { *CHOICE* {[0] **Object Identifier**(global)

[1] **Integer** (local) }
ANY DEFINED BY Acima }

[3] <=2 *SEQUENCE* { *CHOICE* {[0] **Object Identifier**(global)
[1] **Integer** (local) }
ANY DEFINED BY Acima }

[4] **Presente** *CHOICE* {[0] **Object Identifier** (global)
[1] **Integer** (local)

[5] **SubSetOf** *SEQUENCE* { *CHOICE* {[0] **Object Identifier**(global)
[1] **Integer** (local) }
ANY DEFINED BY Acima }

[6] **SuperSetOf** *SEQUENCE* { *CHOICE* {[0] **Object Identifier** (global)
[1] **Integer** (local) }
ANY DEFINED BY Acima }

[7] **nonNullSetIntersection** *SEQUENCE*{ *CHOICE* {[0] **Object Identifier**
[1] **Integer** }
ANY DEFINED BY Acima }

[9] **and**
[10] **or**
[11] **not**
[12] *Set of CHOICE* { [0] Global
[1] Local

GET RESULT

[0] Global [2] DN [5] Tempo de Criação [6] Lista of CHOICE { [0] e [1] }

[1] Local [3] Octet String

[4] LDN

ERRORS

LINKED

CHOICE [0] **Get Result**

[1] **GetListError**

[2] **SetResult**

[3] **SetListError**

[4] **ActionResult**

[5] **ProcessingFailure**

[6] **DeleteResult**

[7] **ActionError**

[8] **DeleteError**

PDU SNMP

A pdu SNMP é formada pelos campos:

- nome da comunidade
- endereço fonte
- endereço destino
- autenticação.

Para PDUs corretas, GetResponse-PDU é igual, sendo que os valores requisitados pelo Get vão na lista (VarBindList).

Para cada nome no variable-binding, o componente correspondente do GetResponse representa o nome e o valor daquele objeto cujo nome é, em ordem lexicográfica de nomes;

o conjunto de todos os objetos disponíveis para operações Get na MIB, junto com o valor do campo nome de dado componente, o imediato sucessor para aquele componente.

A resposta do SetRequest é o GetResponse.

PDUs::= *CHOICE*{ [0] get-request **GetRequest-PDU**
[1] get-next-request **GetNextRequest-PDU**
[2] get-response **GetResponse-PDU**
[3] set-request **SetRequest-PDU**
[4] trap **Trap-PDU** }

RequestID ::= INTEGER

ErrorStatus ::= INTEGER { **noError (0), tooBig (1), noSuchName(2), badValues(3), readOnly(4), genErr (5)** }

ErrorIndex ::= INTEGER

VarBind ::= *SEQUENCE* { name **OBJECT NAME**,
value **OBJECT SYNTAX** }

VarBindList ::= *SEQUENCE OF* **VarBind**

GetRequest-PDU ::= [0] *IMPLICIT SEQUENCE* {request-id **RequestId**
error-status **ErrorStatus**
error-index **ErrorIndex**
variable-bindings **VarBindList** }
GetNextRequest-PDU::=[1] *IMPLICIT SEQUENCE* {request-id **RequestId**
error-status **ErrorStatus**

error-index **ErrorIndex**
variable-bindings **VarBindList** }

GetResponse-PDU::=[2] *IMPLICIT SEQUENCE* {request-id **RequestId**
error-status **ErrorStatus**
error-index **ErrorIndex**
variable-bindings **VarBindList** }

SetRequest-PDU::=[3] *IMPLICIT SEQUENCE* { request-id **RequestId**
error-status **ErrorStatus**
error-index **ErrorIndex**
variable-bindings **VarBindList** }

Trap-PDU ::= [4] *IMPLICIT SEQUENCE* {enterprise **OBJECT IDENTIFIER**
agent-addr **NetworkAddress**
generic-trap INTEGER {coldStart (0)
warmStart (1)
linkDown (2)
linkUp (3)
authenticationFailure (4)
egpNeighborLoss (5)
enterpriseSpecific (6)
specific-trap INTEGER
time-stamp TimeTicks
variableBindings VaBindList }.

Anexo B - Descrição das Hierarquias nos modelos OSI e Internet

O objetivo deste anexo é apresentar as hierarquias da informação de gerenciamento nos dois modelos apresentados neste trabalho.

Hierarquias da Informação de Gerenciamento

Os objetos gerenciados relacionam-se uns com os outros, destacando-se dois tipos de relacionamentos que são de particular importância para a informação de gerenciamento: o relacionamento de *containment* e o relacionamento de herança. Esses relacionamentos podem ser usados para construir hierarquias de objetos gerenciados. Além disso, existe outra hierarquia definida pelo processo de registro para armazenar identificadores para classes de objetos e atributos.

A Hierarquia de Registro

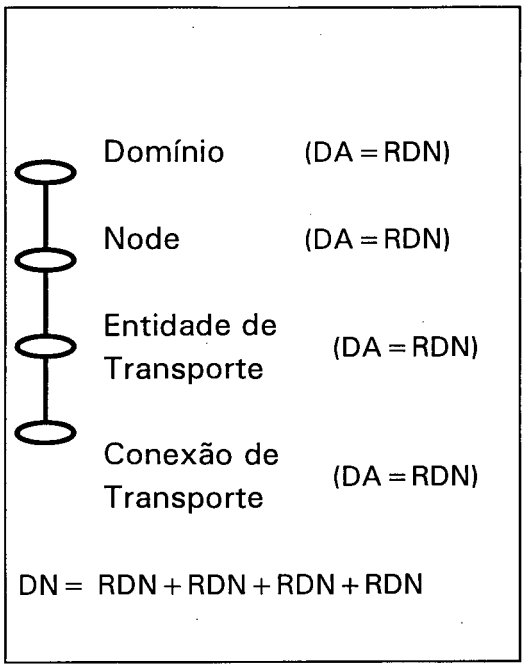
A hierarquia de registro é determinada pela árvore de registro ASN.1 para associar OBJECT IDENTIFIERs (OIDs). Um OID é um nome associado administrativamente, composto de uma série de inteiros representando um endereço desde a raiz da árvore de registro até o nó ou folha a ser identificado. Por exemplo, a seqüência de inteiros { iso(1) standard(0) ips-osi-mips(9596) cmip(2) } (1.0.9596.2) pode ser usada para identificar unicamente o padrão CMIP. Cada nó dessa árvore tem uma autoridade de registro associada que determina quantos números na subárvore definidos para aquele nó são alocados. No contexto do gerenciamento, esses OIDs são usados para identificar classes de objetos e seus atributos. A hierarquia de registro não é baseada em qualquer relacionamento particular entre objetos gerenciados ou entre objetos gerenciados e seus atributos. É independente dos relacionamentos de herança e *containment*. Seu propósito é simplesmente gerar identificadores únicos.

A Hierarquia de *Containment*

A hierarquia de *containment* é construída pela aplicação do relacionamento "está contido em" em objetos e atributos. Os objetos de uma classe podem conter objetos da mesma ou de classes diferentes. Os objetos podem, também, conter atributos. Os atributos não podem conter objetos ou outros atributos. Por exemplo, objetos da classe "Entidade de Transporte" podem conter objetos da classe "Conexão de Transporte"; um objeto da classe "domínio de Gerenciamento" pode conter objetos da classe "Node".

A hierarquia de *containment* é importante porque ela pode ser usada para identificar instâncias de um objeto gerenciado. Por exemplo, considere que existe uma classe de objeto "Domínio" que contém uma classe de objeto "Node", que contém uma classe de objeto "Entidade de Transporte", que por sua vez, contém uma classe de objeto "Conexão de Transporte". Uma instância particular da conexão de transporte pode ser identificada pela concatenação da informação da instância para cada classe de objeto no endereço de *containment* [RFC1095]:

{ domínio="organization", node="herakles", Entidade de Transporte=tp4, Conexão de transporte= <TSAP-AddressA,TSAP-AddresB> }.



O que constitui a informação da instância apropriada para cada classe de objeto é parte da definição daquela classe de objeto e é conhecida como *Distinguished Attribute* (DA). Um DA é composto de um OBJECT IDENTIFIER que dá nome ao atributo e o valor do atributo. Para cada classe de objeto, os DA que diferenciam instâncias daquela classe são coletivamente chamados *Relative Distinguished Names* (RDN). A seqüência de RDNs (um para cada classe no endereço de containment) é o *Distinguished Name* (DN) do objeto gerenciado. O exemplo acima representa o DN de uma conexão de transporte. A hierarquia de *containment* é algumas vezes referenciada como "Árvore de Nomeação", porque ela é usada para nomear uma instância particular de um objeto gerenciado.

O relacionamento de *containment* também define a dependência existente entre seus componentes; um objeto ou atributo pode existir somente se o objeto que o contém também existe. A exclusão de um objeto pode resultar na exclusão de todos os objetos e atributos contidos nele. Alternadamente, dependendo da definição do objeto gerenciado, a exclusão pode ser recusada até que todos os objetos contidos tenham sido apagados.

A Hierarquia de Herança

A hierarquia de herança é construída pela aplicação do relacionamento "herda características de" na classe de objetos. Uma classe de objetos pode herdar características de outra classe de objetos; refinamento é obtido pela adição de características adicionais. Nesse relacionamento, a classe pai é chamada "Superclasse" e a classe que herda é "Subclasse". Por exemplo, a classe "Entidade de Camada" pode ser uma superclasse de "Entidade de Rede", que por sua vez é superclasse de "Entidade de Rede X.25". Atributos definidos para "Entidade de Rede" (por exemplo, número de pacotes enviados) são automaticamente definidos para "Entidade de Rede X.25" sem ter que explicitamente inclui-los na definição para a classe "Entidade de Rede X.25". A classe maior é a classe TOP que não possui características. A hierarquia de herança não tem relevância para nomear as instâncias de objetos.

A Internet SMI

A Internet SMI determina como identificar objetos gerenciados e como defini-los. Define, também, uma subárvore de registro iniciada em {iso(1) org(3) dod(6) internet(1)} como o início do registro dos OBJECT IDENTIFIERs a serem usados para identificar unicamente os objetos gerenciados. A Internet SMI atual especifica o formato para definir objetos em termos de um template *"object type"* e uma macro ASN.1 OBJECT TYPE associada. Uma definição de "*object type*" contém cinco campos: um nome textual, de acordo com o seu OID correspondente; uma sintaxe ASN.1; uma definição da semântica do object type; um acesso (*read-only*, *read-write*, *write-only*, *not_acessible*); e um *status* (obrigatório, opcional, obsoleto). Na descrição da informação de gerenciamento, ela não usa as noções de classe de objetos e atributos encontradas na ISO SMI. Somente os conceitos de *"object type"* e *"object instance"* são usados. A Internet SMI mostra como definir *object types*; ela trata a especificação das instâncias de objetos como um problema específico do protocolo. A estrutura não distingue entre objetos simples e compostos (objeto composto é

aquele que herda características de uma superclasse), ambos são definidos como *object type*. Os objetos que são considerados atributos de outros objetos que os contém são definidos diretamente abaixo deles na árvore de registro de objetos. Isso resulta numa certa falta de flexibilidade, desde que a hierarquia de registro é implicitamente usada para definir a hierarquia de containment. Isso significa que a Internet SMI não contém um mecanismo para definir relacionamentos containment que não coincidem com a hierarquia de registro. Na interpretação da Internet SMI para usar com o CMIP, é necessário suprir essa limitação.

Descreve-se a seguir como as três hierarquias da informação de gerenciamento são compreendidas pela Internet SMI.

A Hierarquia de Registro

A hierarquia de registro é a árvore de registro de objeto global descrita em [RFC1155]. É usada meramente para associar identificadores para as classes de objetos e os atributos.

A Hierarquia de *Containment*

A hierarquia de *containment* é usada para especificar uma instância de objeto. O campo *Names* da definição da classe contém o distinguished attributes para cada classe de objeto. O OBJECT IDENTIFIER que dá nome ao atributo junto com seu valor é chamado *Attribute Value Assertion*(AVA). Um conjunto de AVAs (um para cada *distinguished attribute*) é o RDN associado com aquela classe de objeto. A seqüência de RDNs para cada uma das classes de objeto na hierarquia de containment a qual a classe pertence é o DN do objeto. Uma instância de objeto é totalmente especificada pelo DN.

Considere o exemplo a seguir. Para representar uma instância de uma entrada na tabela de roteamento IP, é necessário começar pelo exame da classe de objeto em questão

(ipRouteEntry) e usar o campo *Superiors* para encontrar a classe superior na hierarquia de containment (ipRouteTable). Esse processo continua até a construção do seguinte endereço de containment: system, ip, ipRoutingTable, ipRoutingEntry. Agora para cada uma dessas classes de objetos, deve-se verificar o campo *Names* para encontrar o DA para cada uma dessas classes. Se nenhum campo *Names* está presente (como o caso para "ip " e "ipRoutingTable"), então nenhuma instância é requisitada naquele nível. "System" e "ipRoutingEntry" tem campos *Names* para mostrar que a informação é esperada naquele nível. Com essa informação, pode-se construir os seguintes DN's especificando uma instância de uma entrada na tabela de roteamento *Ip:*

```
baseManagedObjectInstance {
                distinguishedName {
                relativeDistinguishedName {  --system
                        atributeValueAssertion {
                            attributeType { cmotSystemID  }
                            attributeValue "gateway1.acme.com" }
                                        }
                relativeDistinguishedName {  --ipRouteEntry
                                atributeValueAssertion {
                            attributeType { ipRouteDest }
                            attributeValue  10.0.0.51    }
                                        }
                                    }
}
```

Note que a árvore da instância de objeto pode conter componentes no DN que estão fora do sistema gerenciado (node). Isso permite a referência de objetos através de domínios de gerenciamento (pode existir uma classe de objetos "domínio") e através de uma seleção de nós. Numa rede onde vários gerentes intermediários podem estar envolvidos na requisição, cada gerente intermediário pode usar a parte "*system*" do nome para determinar

onde enviar uma requisição ou resultado. Essa técnica de nomeação trata cada sistema de gerenciamento intermediário como um proxy manager. O *proxy manager* resolve o endereço próximo nó numa cadeia e pode usar um protocolo diferente para transferir a requisição ou resultado. Além disso, a informação da instância *System* pode ser usada para nomear dispositivos sendo gerenciados pelo proxy.

A Hierarquia de Herança

A Internet SMI não usa o relacionamento de herança. O campo *"SubClass of"* está presente na definição da classe de objeto para mostrar como o relacionamento de herança poderia ser representado e permitir futuras extensões.