ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

FERNANDO FROTA REDÍGOLO

Proposta de um Agente CNM para o gerenciamento *web* de um *backbone* ATM

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia.

São Paulo
2001

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

FERNANDO FROTA REDÍGOLO

Proposta de um Agente CNM para o gerenciamento *web* de um *backbone* ATM

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia.

Área de Concentração:

Sistemas Digitais

Orientadora:

Profa. Dra. Tereza Cristina Melo de Brito Carvalho

São Paulo
2001

AGRADECIMENTOS

Aos meus pais, agradeço pelo carinho e incentivo durante toda a minha vida. Por todo o sacrifício que fizeram para que eu pudesse ter condições de crescer e atingir os meus objetivos, a minha eterna gratidão.

À Carine, minha esposa, um muito obrigado pela compreensão, carinho e paciência ao longo de todos estes anos juntos.

A meu irmão Eduardo, obrigado pela força e companheirismo.

À Prof. Dra. Tereza Cristina, pelo grande apoio, pela enorme confiança depositada em mim e em todos os meus trabalhos e pela valiosa orientação.

Ao Prof. Dr. Wilson Ruggiero, por acreditar em mim desde o início e pela oportunidade de trabalhar em um excelente ambiente de pesquisa.

Aos meus colegas do LARC, pela amizade nesses anos de convivência. Em particular, a Cíntia e Oscar, pela ajuda nos mais diversos momentos.

Enfim, o meu sincero obrigado a todos que estiveram comigo e colaboraram para a realização de mais esta conquista.

SUMÁRIO

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

LISTA DE ABREVIATURAS

AAL	*ATM Adaptation Layer*
ABR	*Available Bit Rate*
API	*Application Programming Interface*
ARP	*Address Resolution Protocol*
ASN.1	*Abstract Syntax Notation One*
ASP	*ActiveX Server Pages*
ATM	*Asynchronous Transfer Mode*
BER	*Basic Encoding Rules*
BUS	*Broadcast and Unknown Server*
CBR	*Constant Bit Rate*
CDVT	*Cell Delay Variation Tolerance*
CGI	*Common Gateway Interface*
CIM	*Common Information Model*
CIMOM	*Common Information Model Object Manager*
CLIP	*Classical IP over ATM*
CMIP	*Common Management Information Protocol*
CNM	*Customer Network Management*
CPCS	*Common Part Convergence Sublayer*
DEN	*Directory-Enabled Networking*
DMTF	*Distributed Management Task Force (antigo Desktop Management Task Force)*
DTD	*Document Type Definition*
DWDM	*Dense Wavelength Division Multiplexing*
ELAN	*Emulated LAN*
FDDI	*Fiber Distributed Data Interface*
HEC	*Header Error Control*

HTML	*HyperText Markup Language*
HTTP	*Hypertext Transfer Protocol*
IAB	*Internet Architecture Board* (antigo *Internet Activities Board*)
ICMP	*Internet Control Message Protocol*
IEEE	*Institute of Electrical and Electronic Engineers*
IETF	*Internet Engineering Task Force*
ILMI	*Integrated Local Management Interface*
IP	*Internet Protocol*
IPoA	*IP over ATM*
ISAPI	*Information Server Application Programming Interface*
ISO	*International Organization for Standardization*
ITU	*International Telecommunication Union*
ITU-T	*ITU Telecommunication Standardization Sector*
JMX	*Java Management eXtensions*
LAN	*Local Area Network*
LANE	*LAN Emulation*
LDAP	*Lightweight Directory Access Protocol*
LEC	*LAN Emulation Client*
LECS	*LAN Emulation Configuration Server*
LES	*LAN Emulation Server*
LIS	*Logical IP Subnetwork*
LLC	*Logical Link Control*
MAN	*Metropolitan Area Network*
MBS	*Maximum Burst Size*
MCR	*Minimum Cell Rate*
MIB	*Management Information Base*
MPOA	*MultiProtocol over ATM*

MTU	*Maximum Transfer Unit*
NMS	*Network Management System*
NNI	*Network Node Interface / Network to Network Interface*
nrt-VBR	*non-real-time Variable Bit Rate*
NSAPI	*Netscape Server Application Programming Interface*
OID	*Object IDentifier*
OSI	*Open System Interconnection*
OSPF	*Open Shortest-Path First*
PCR	*Peak Cell Rate*
PDH	*Plesiochronous Digital Hierarchy*
PDU	*Protocol Data Unit*
PHP	*PHP Hypertext Preprocessor*
PLCP	*Physical Layer Convergence Procedure*
PNNI	*Private Network-Network Interface / Private Network Node Interface*
PVC	*Permanent Virtual Connection*
QoS	*Quality of Service*
RFC	*Request for Comment*
RIP	*Routing Information Protocol*
RMAV-SP	*Rede Metropolitana de Alta Velocidade de São Paulo*
RMON	*Remote MONitoring*
rt-VBR	*real-time Variable Bit Rate*
SCR	*Sustainable Cell Rate*
SDH	*Synchronous Digital Hierarchy*
SDU	*Service Data Unit*
SGML	*Standard General Markup Language*
SLA	*Service Level Agreement*
SMI	*Structure of Management Information*

SNMP	*Simple Network Management Protocol*
SONET	*Synchronous Optical NETwork*
SPVC	*Semi-Permanent Virtual Connection*
SSL	*Secure Sockets Layer*
SVC	*Switched Virtual Connection*
TCP	*Transmission Control Protocol*
TMN	*Telecommunications Management Network*
UBR	*Unspecified Bit Rate*
UDP	*User Datagram Protocol*
UNI	*User-Network Interface*
UPS	*Uninterruptible Power Supply*
VBR	*Variable Bit Rate*
VC	*Virtual Channel*
VCC	*Virtual Channel Connection*
VCI	*Virtual Channel Identifier*
VCL	*Virtual Channel Link*
VP	*Virtual Path*
VPC	*Virtual Path Connection*
VPI	*Virtual Path Identifier*
VPL	*Virtual Path Link*
WAN	*Wide Area Network*
WBEM	*Web-Based Enterprise Management*
XML	*eXtensible Markup Language*
XSL	*eXtensible Style Language*

RESUMO

O surgimento de novas aplicações multimídia tem estimulado a utilização de redes de alta velocidade com garantia de qualidade de serviço. A tecnologia ATM (*Asynchronous Transfer Mode*) tem se destacado, dadas as suas altas taxas de *throughput* e sua versatilidade em suportar aplicações com diferentes requisitos de atraso, banda, entre outros. Sua versatilidade, no entanto, traz consigo uma grande complexidade, dificultando o gerenciamento de uma rede baseada nesta tecnologia.

Para um serviço de comunicação multimídia baseado em ATM, o provedor deste serviço pode compartilhar informações e até mesmo funções de gerenciamento com seus clientes, através de um serviço adicional denominado gerenciamento CNM (*Customer Network Management).*

O presente trabalho tem por objetivo definir uma arquitetura modular e extensível para um sistema capaz de prover um serviço de gerenciamento CNM para usuários de um *backbone* ATM, aplicando técnicas de gerenciamento *web* para este serviço.

Para definir os requisitos necessários para esta arquitetura, são feitos o levantamento e a análise dos principais padrões de gerenciamento para redes ATM existentes, bem como dos padrões para gerenciamento via *web.* Os requisitos de uma arquitetura para um sistema de gerenciamento CNM são definidos a partir desta análise, sendo a arquitetura via *web* proposta baseada nestes requisitos.

A implementação de um protótipo desta arquitetura via *web* complementa o trabalho, permitindo a verificação da viabilidade da arquitetura proposta.

ABSTRACT

The multimedia applications have caused the increasing use of high-speed networks that can guarantee quality of service. The ATM (Asynchronous Transfer Mode) technology has been attracting attention, due to its high throughput and its versatility in supporting applications with different requirements, such as delay and bandwidth requirements, among others. This versatility, however, also brings complexity, making the management of networks based on this technology much harder than traditional ones.

For an ATM-based, multimedia-communication service, its provider can share information and even management functions with its customers, by an additional service called Customer Network Management, or CNM.

The main goal of this work is the definition of an extensible, modular architecture for a system capable of providing CNM service for the users of an ATM backbone. The CNM service provided by the proposed architecture applies web-based management techniques.

In order to define the requirements of this architecture, this work includes a survey and analysis of the main ATM management standards, as well as of the web-based management standards. From this analysis, the requirements of an architecture for a CNM management system are defined, and a web-based architecture is proposed according to these requirements.

The implementation of a prototype of the web-based architecture fulfills this work, allowing the verification of the viability of the proposed architecture.

1. Introdução

A tecnologia ATM (*Asynchronous Transfer Mode*) tem sido bastante empregada como base para redes metropolitanas (MANs – *Metropolitan Area Networks*) ou de longa distância (WANs – *Wide Area Networks*) que necessitam de altas taxas de transmissão e garantia de qualidade de serviço para aplicações multimídia. O gerenciamento destas redes ATM é o foco deste trabalho.

1.1. Objetivos

Um serviço de gerenciamento CNM (*Customer Network Management*) permite a comunicação entre o sistema de gerenciamento de uma instituição usuária de uma rede ATM ou *Frame Relay* pública (*customer*) e o sistema de gerenciamento desta própria rede pública, permitindo aos administradores das redes destas instituições um gerenciamento limitado da rede pública, correspondente à parte por eles utilizada. Tal gerenciamento pode ser tanto passivo (o administrador supervisiona a utilização do *backbone*) como ativo (permitindo ao administrador um certo controle sobre o *backbone*).

O presente trabalho tem como objetivo principal o desenvolvimento de uma arquitetura capaz de oferecer um serviço de gerenciamento CNM de um *backbone* ATM a instituições usuárias deste *backbone*. Tal arquitetura visa obter os seguintes benefícios:

- Possibilitar aos administradores destas instituições o acesso a interfaces e ferramentas de gerenciamento ATM, evitando que haja sistemas complexos de gerenciamento ATM em suas redes;
- Diminuir os custos de gerenciamento para as instituições, evitando que eles necessitem de novas ferramentas ou plataformas para o gerenciamento ATM;
- Evitar ou diminuir a necessidade de treinamento para que os administradores das redes privativas destas instituições possam gerenciar a sua parte do *backbone*;

- Controlar o que cada administrador pode fazer em termos de gerenciamento, evitando que a operação de um administrador interfira no comportamento global do *backbone*;
- Prover uma infra-estrutura escalar para o serviço, capaz de agregar novas funcionalidades quando necessário.

Para melhor avaliar a adequação do agente frente aos objetivos propostos, o agente será implementado na infra-estrutura ATM existente da RMAV-SP (Rede Metropolitana de Alta Velocidade de São Paulo) [15].

1.2. Motivação

Atualmente, há uma grande tendência de integração de aplicações de voz, dados e vídeo sobre uma única infra-estrutura de rede. A tecnologia ATM tem recebido bastante atenção, por ser uma solução viável para esta integração, oferecendo garantia de qualidade de serviço (QoS – *Quality of Service*) necessária para estas aplicações. No entanto, a complexidade da tecnologia ATM impõe alguns desafios para o seu gerenciamento. Como fatores que dificultam o gerenciamento de uma rede ATM, pode-se citar [1][3]:

- Diversas infra-estruturas de transmissão na camada física. Como exemplo, podem-se citar as redes SONET/SDH (*Synchronous Optical NETwork/ Synchronous Digital Hierarchy*), PDH (*Plesiochronous Digital Hierarchy*) ou mesmo DWDM (*Dense Wavelength Division Multiplexing*);
- Em cada enlace físico, há diversos "enlaces virtuais", formados pela combinação de VPLs (*Virtual Path Links*) e VCLs (*Virtual Channel Links*). Estes enlaces constituem circuitos virtuais, que podem ser do tipo ponto-a-ponto ou ponto-multiponto; além disso, os circuitos virtuais podem ser permanentes (PVCs – *Permanent Virtual Circuits*), semi-permanentes (SPVCs – *Semi-Permanent Virtual Circuits*) ou comutados (SVCs – *Switched Virtual Circuits*);

- Cada circuito virtual possui associado a si um tipo de serviço, com requisitos distintos de QoS (tais como atraso e taxa de perda aceitáveis e banda necessária), bem como perfis de tráfego também diferenciados. Como exemplos desses tipos de serviço, pode-se citar: ABR (*Available Bit Rate*), UBR (*Unspecified Bit Rate*), rt-VBR (*Real-Time Variable Bit Rate* –), nrt-VBR (*Non-Real-Time Variable Bit Rate*) e CBR (*Constant Bit Rate*);

- Os circuitos virtuais comutados podem ter um tempo de vida bastante curto, podendo ser criados e encerrados freqüentemente, dificultando a sua monitoração.

Como a rede opera em dois níveis (rede física e rede lógica), o gerenciamento deve levar em conta as diferenças de funcionamento dos dois níveis. Por exemplo, enquanto que a monitoração de desempenho e falhas dos enlaces físicos é semelhante ao das redes tradicionais (deve-se garantir que os enlaces físicos operem na maior taxa de transmissão possível, aproveitando-se ao máximo a banda disponível no enlace e minimizando o *downtime*), a monitoração de circuitos virtuais, no entanto, apresenta desafios adicionais. Em um dado enlace físico, há tráfego de diferentes circuitos virtuais e, para uma dada transmissão entre dois elementos da rede, podem ser utilizados diferentes circuitos virtuais, de acordo com as condições da rede (por exemplo, a queda de um enlace pode causar a reconfiguração dos circuitos virtuais que dele se utilizavam). Sendo assim, o sistema de gerenciamento deve ser capaz de acompanhar estas mudanças dos circuitos virtuais através de sua monitoração. O sistema deve ser capaz, também, de monitorar parâmetros de transmissão (tais como células transmitidas e recebidas, atrasos, banda alocada e utilizada) não só para os enlaces físicos como para cada circuito virtual.

Em função destes fatores e da velocidade com que opera, uma rede ATM pode gerar um volume muito grande de informações de gerenciamento, tais como estatísticas e alarmes, em intervalos de tempo muito curtos. Torna-se necessário, então, a utilização de ferramentas que possam auxiliar na tarefa de gerenciamento de uma rede ATM, apresentando aos administradores da rede somente informações significativas. Tais ferramentas exigem

plataformas computacionais capazes de manipular, de forma rápida, o grande volume de dados passíveis de serem gerados pela rede.

A utilização de um *backbone* ATM público para a interconexão de redes privativas apresenta, também, novos desafios na área de gerenciamento de redes. Quando uma rede corporativa é formada por diversas redes locais distribuídas geograficamente e interconectadas através de enlaces ponto-a-ponto, os administradores desta rede podem detectar facilmente problemas tanto na camada física como na camada de enlace. Se, no entanto, estas redes forem interconectadas através de uma WAN ATM, a dificuldade na detecção das causas de problemas passa a ser significativamente maior, uma vez que, para se gerenciar os enlaces, é necessário que os administradores tenham acesso a um número muito maior de variáveis (dada a própria complexidade das redes baseadas nesta tecnologia). Além disso, muitas destas variáveis são parâmetros de funcionamento do próprio *backbone* ATM, do qual os administradores nem sempre têm acesso, dificultando ainda mais o gerenciamento.

Para melhor abordar a questão do gerenciamento de redes ATM, o ATM Forum (uma das entidades de padronização da tecnologia ATM) define uma arquitetura de gerenciamento. Para atender às necessidades de administradores de redes corporativas que se utilizam dos serviços de um *backbone* ATM público, a arquitetura de gerenciamento do ATM Forum define que as redes públicas podem oferecer um serviço de gerenciamento do *backbone* ATM a seus usuários. Tal serviço é conhecido como CNM (*Customer Network Management*) e oferece às instituições usuárias (*customers*) desta rede pública acesso a informações e operações de gerenciamento que, normalmente, são exclusivas dos gerentes do *backbone* público.

A necessidade de um gerenciamento CNM não é exclusiva do contexto de um *backbone* ATM, aparecendo, também, em *backbones* que utilizam tecnologias como *Frame Relay* ou mesmo *backbones* IP. Além disso, o gerenciamento CNM não é exclusivo de redes públicas. Um serviço CNM pode ser necessário em qualquer *backbone* WAN ou MAN, cujo uso é compartilhado por grupos ou departamentos diferentes que necessitam de um acesso restrito a funções e informações de gerenciamento deste *backbone*. Nestas redes, geralmente um

determinado departamento é responsável pelo gerenciamento do *backbone* WAN, porém há a necessidade de compartilhar algumas informações deste gerenciamento entre os administradores de outros departamentos que compartilham o uso deste *backbone*. As redes acadêmicas de grandes universidades podem se encaixar neste perfil de redes, pois em geral possuem um departamento responsável pelo gerenciamento da rede do campus e/ou das conexões entre os campi, sendo que normalmente as escolas e institutos que compõem uma universidade possuem administradores próprios para suas redes internas (que cuidam também do acesso ao *backbone* do campus).

Apesar da necessidade de um serviço de gerenciamento CNM, são poucas as soluções de gerenciamento capazes de implementá-lo. Além disso, as implementações existentes são pertinentes a soluções de redes (tanto de equipamento como de plataforma de gerenciamento) de um único fabricante, impedindo o desenvolvimento de um serviço de gerenciamento CNM para uma rede heterogênea.

1.3. Organização do Trabalho

Este trabalho encontra-se organizado em 4 capítulos principais, além do capítulo introdutório:

- Capítulo 2: Gerenciamento de Redes ATM

  Apresenta um levantamento sobre as arquiteturas envolvidas no gerenciamento de redes ATM, bem como outras especificações de sistemas de gerenciamento pertinentes ao trabalho. Encontra-se dividido em 3 partes principais:

  - Apresentação da arquitetura de gerenciamento SNMP (*Simple Network Management Protocol*), seus principais elementos e protocolos;
  - Descrição do modelo de gerenciamento de redes ATM definido pelo ATM-Forum, detalhando-se as interfaces de gerenciamento (interfaces M1-M5). A interface M3, responsável pelo serviço de gerenciamento CNM, é apresentada com maiores detalhes, uma vez que esta interface define os requisitos de um agente CNM;

- Discussão a respeito de soluções de gerenciamento via *web*, apresentando-se vantagens e desvantagens, bem como os padrões de gerenciamento via *web* em desenvolvimento;

- Capítulo 3: Padrões de Gerenciamento ATM:

  Contém uma análise dos padrões existentes para o gerenciamento de redes ATM, que deverão ser utilizados pelo agente. É dividido em 2 partes:

  - Discussão das entidades de padronização responsáveis pela definição das MIBs padronizadas para o gerenciamento ATM: o IETF (*Internet Engineering Task Force*) e o ATM-Forum;
  - Discussão das principais MIBs (*Management Information Bases*) definidas por estas entidades de padronização;

- Capítulo 4: Requisitos para um agente CNM

  Discute os requisitos de uma arquitetura para o gerenciamento CNM de uma rede ATM, visando atender os objetivos propostos. É dividido em três partes principais:

  - Especificação dos requisitos de um serviço CNM;
  - Especificação dos requisitos da arquitetura de uma solução para gerenciamento CNM, bem como dos protocolos de comunicação;
  - Especificação dos requisitos funcionais de gerenciamento da solução proposta;

- Capítulo 5: Agente CNM via Web

  Analisa e avalia a arquitetura para o gerenciamento CNM proposta no trabalho. É dividido em duas partes principais:

  - Discussão das características da solução definida no trabalho, tais como a arquitetura da ferramenta, linguagens e plataformas computacionais adotadas;
  - Discussão da implementação da solução em uma rede de produção e da experiência prática resultante da implementação dessa solução;

- Capítulo 6: Considerações Finais

  Apresenta uma análise crítica do trabalho, avaliando vantagens e desvantagens das soluções e contribuições trazidas pelo trabalho e discute possíveis desenvolvimentos futuros.

1.4. Terminologia Utilizada

Alguns termos são utilizados dentro de um escopo específico no decorrer deste trabalho; para um melhor entendimento, os principais termos são descritos a seguir.

A tecnologia ATM é utilizada tanto para MANs e WANs privativas (pertencentes a uma determinada empresa, centro de pesquisa ou organização, por exemplo) como para MANs e WANs públicas (pertencentes a uma empresa de telecomunicações, que oferece serviços de conectividade ATM). Estas redes ATM são denominadas genericamente de ***backbone* ATM** no decorrer do trabalho. Os *switches* e roteadores da rede, bem como os servidores responsáveis por serviços ATM são denominados genericamente de **elementos** da rede ATM. Um **serviço ATM** é um protocolo que é utilizado para complementar as funcionalidades da rede ATM; como exemplo pode-se citar o *Classical IP over ATM* (CLIP) e o *LAN Emulation* (LANE).

Entende-se por **instituições usuárias de uma rede ATM,** ou simplesmente **instituições,** as empresas, centros de pesquisas, institutos, organizações, entre outros, que são interligadas através de uma MAN ou WAN ATM, cujo gerenciamento e manutenção é de responsabilidade de terceiros. Sendo assim, para redes ATM públicas, as instituições são os clientes do provedor de conectividade, enquanto que, para redes ATM privativas, as instituições são grupos administrativos diferentes do responsável pela manutenção da rede (por exemplo, universidades conectadas a uma WAN ATM acadêmica ou hospitais interligados por uma MAN ATM comum a centros de saúde).

O termo **entidade** é utilizado em duas situações distintas:

- Organizações responsáveis pela especificação de padrões e normas, são denominadas **entidades de padronização.** Estas organizações são compostas por

diferentes grupos de indivíduos, responsáveis pelos diferentes padrões. Estes grupos são denominados de **grupos de trabalho**;

- Software responsável pela implementação das funcionalidades de uma pilha de protocolos de comunicação.

A arquitetura do sistema de gerenciamento CNM proposto neste trabalho é descrita em termos de **módulos**, que são *softwares* responsáveis pela implementação das funcionalidades do sistema. O agrupamento destes módulos recebe o nome de **componentes funcionais** ou simplesmente **componentes.** Por fim, o conjunto das funcionalidades do sistema de gerenciamento CNM define os **níveis funcionais** do sistema.

2. Arquiteturas para gerenciamento de redes ATM

São especificadas pelas entidades de padronização diversas arquiteturas que definem como deve ser feito o gerenciamento de redes. Para monitorar e controlar uma rede ATM (*Asynchronous Transfer Mode*) devem ser estudadas as diversas arquiteturas que podem ser utilizadas no seu gerenciamento.

São três as arquiteturas de gerenciamento a serem vistas:

- Arquitetura SNMP(*Simple Network Management Protocol*): tal arquitetura é bastante popular, podendo-se encontrar um grande número de ferramentas, bem como suporte a esta arquitetura nos mais diversos equipamentos de rede. É arquitetura mais difundida para gerenciamento de redes corporativas;
- Arquitetura para gerenciamento via *web*: estas arquiteturas têm sido bastante discutidas nos últimos anos, havendo um grande esforço no seu desenvolvimento. Trazem diversos benefícios para soluções de gerenciamento, especialmente na área de segurança (devido à grande utilização de sistemas baseados na *web* em aplicações que exigem segurança como, por exemplo, sistemas de comércio eletrônico);
- Modelo de gerenciamento do ATM Forum: neste modelo são definidos os requisitos para gerenciamento fim-a-fim de uma rede ATM, de acordo com a visão do ATM Forum.

Baseadas nestas arquiteturas foram definidos diversos padrões para o gerenciamento de redes ATM, que devem ser vistos a seguir.

2.1. Arquitetura SNMP

A arquitetura de gerenciamento para redes TCP/IP (*Transmission Control Protocol/Internet Protocol*) é denominada de arquitetura SNMP (*Simple Network Management Protocol*). É uma arquitetura bastante popular, havendo uma grande disponibilidade de produtos baseados em seus padrões.

Assim como os padrões relacionados à arquitetura TCP/IP, a arquitetura SNMP é definida através de especificações denominadas RFCs (*Request for Comments)*. As RFCs são, em sua grande maioria, desenvolvidas por grupos de trabalho do IETF (*Internet Engineering Task Force*), entidade responsável pelas padronizações relacionadas à arquitetura TCP/IP. A base da arquitetura SNMP é definida pelo conjunto de RFCs apresentadas na Tabela 2-1 **[34][51]**.

RFC	***Status***	**Nome**
1157	*Standard – STD 15*	*A Simple Network Management Protocol (SNMP)* **[16]**
1155	*Standard – STD 16*	*Structure and Identification of Management Information for TCP/IP-Based Internets* **[48]**
1212	*Standard – STD 16*	*Concise MIB Definitions*
1213	*Standard – STD 17*	*Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II* **[49]**

Legenda: IP *Internet Protocol*
MIB *Management Information Base*
SNMP *Simple Network Management Protocol*
TCP *Transmission Control Protocol*

Tabela 2-1: RFCs contendo as especificações básicas da arquitetura SNMP

Há muitas outras especificações relacionadas à arquitetura SNMP, que estendem as funcionalidades definidas pelas especificações básicas. No entanto, a maioria encontra-se em processo de padronização, não tendo sido completamente aprovada pelo IETF. As especificações da Tabela 2-1 foram aprovadas pelo IETF, constituindo-se em padrões efetivos para as redes TCP/IP. A coluna de *status* indica a identificação destas especificações na relação de protocolos padronizados.

1.1.1 Arquitetura Geral

Na arquitetura SNMP, a estação de trabalho responsável pelo gerenciamento da rede é denominada de estação de gerenciamento. Ela contém um conjunto de aplicativos através do qual o administrador da rede pode interagir com os elementos da rede, tais como:

- Aplicativos para análise de dados, recuperação de falhas, entre outros;
- Interface gráfica para monitoração e controle da rede;

- Processo para comunicação com os elementos da rede, traduzindo as operações requisitadas pelo administrador em operações nos elementos da rede. Este processo é denominado **gerente SNMP**.

Para se monitorar ou controlar um dado dispositivo da rede, é necessário que seja executado no dispositivo um processo responsável pelas tarefas de gerenciamento. Este processo é denominado **agente SNMP**, tendo como funções executar operações de gerenciamento sobre os dispositivos, mediante solicitação do gerente, e notificá-lo de eventos importantes ocorridos no dispositivo (por exemplo, a queda de um enlace). Para o gerenciamento da rede, os seus elementos-chave devem, portanto, ser equipados com um agente SNMP. A Figura 2-1 ilustra um exemplo de uma rede onde somente alguns elementos são gerenciáveis (um servidor de impressão, um roteador e um servidor).

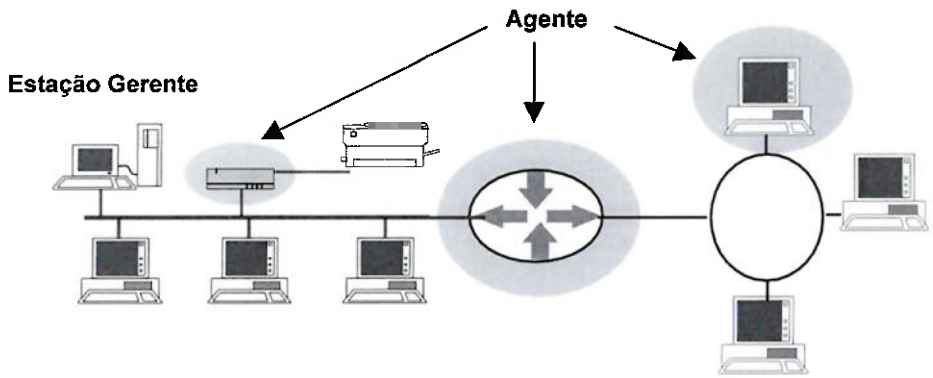


Figura 2-1: Gerente e Agentes em uma rede gerenciável via SNMP

Para o gerenciamento de um dispositivo, são executadas operações equivalentes a amostragens: os recursos a serem gerenciados são representados através dos valores de um conjunto de variáveis. Em outras palavras, através do comportamento de um conjunto de variáveis associadas a um dispositivo, pode-se representar o seu comportamento como um todo. A recíproca também é verdadeira, ou seja, alterando-se o valor de uma variável altera-se o comportamento do dispositivo. Este conjunto de variáveis recebe o nome de **MIB**

(*Management Information Base*), sendo as variáveis denominadas objetos da MIB. A Figura 2-2 representa esta relação entre recursos gerenciáveis e objetos da MIB.

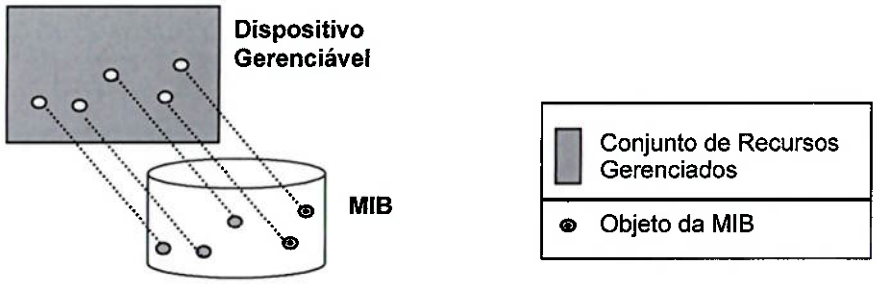


Figura 2-2: Representação de Recursos Gerenciados através de uma MIB

As MIBs devem possuir duas características básicas:

- Cada recurso a ser gerenciado deve estar representado, de forma única, nas diversas MIBs da rede. Em outras palavras, dois elementos da rede devem representar um dado recurso através do mesmo conjunto de objetos de suas MIBs;
- Os objetos devem estar organizados de forma única nas diferentes MIBs.

Estes dois itens garantem a interoperabilidade entre os gerentes e os agentes: o primeiro item garante que o gerente sempre sabe quais são os objetos que ele deve utilizar para gerenciar um dado recurso, independente do equipamento a ser gerenciado, enquanto que o segundo item garante que só existe uma única maneira do gerente referir e manipular estes objetos. Para atingir estes objetivos, o modelo SNMP padroniza os objetos que uma MIB deve ter, assim como as regras de estruturação destes objetos na MIB. Há diversos padrões de objetos para MIBs, de maneira a atender particularidades das diferentes tecnologias (tais como Ethernet ou Token Ring) e tipos de equipamentos (tais como *hubs, switches* ou roteadores) a serem gerenciados.

Para a comunicação entre o gerente e o agente, é utilizado um protocolo de gerenciamento, o **protocolo SNMP**. Ele opera sobre o protocolo TCP/IP, mais especificamente sobre o protocolo de transporte UDP (*User Datagram Protocol*). O agente deve, portanto, implementar não

somente o protocolo SNMP, como também o UDP e os protocolos das camadas inferiores, como os protocolos de rede: IP, ARP (*Address Resolution Protocol*) e ICMP (*Internet Control Message Protocol*). Caso um determinado dispositivo não suporte estes protocolos, ou parte deles, pode-se utilizar mecanismo de *proxy*: um ***proxy* SNMP** é um agente capaz de receber requisições SNMP do gerente direcionadas ao dispositivo não-SNMP, mapeá-las para o protocolo proprietário do dispositivo e converter as respostas do dispositivo em respostas SNMP direcionadas ao gerente.

A arquitetura de protocolos envolvida no modelo de gerenciamento SNMP é exibida na Figura 2-3 **[51]**.

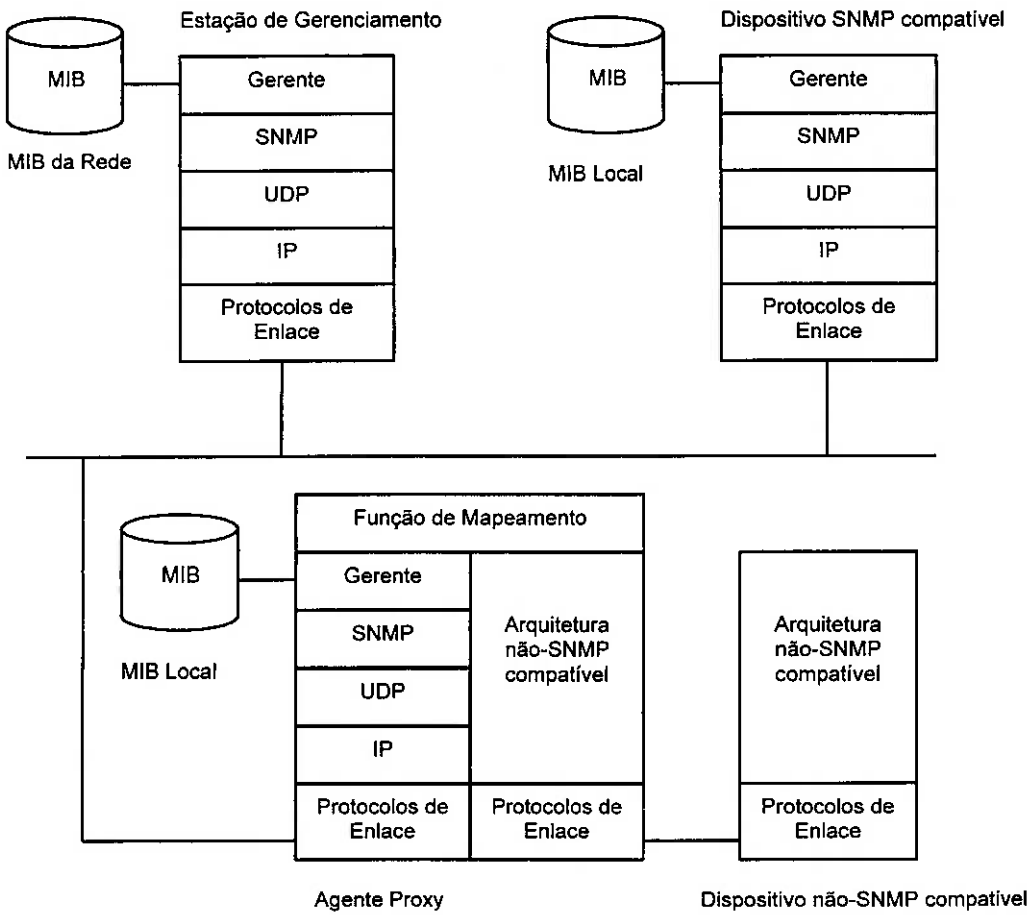


Figura 2-3: Protocolos utilizados na arquitetura SNMP e agente *Proxy*

2.1.1 SMI

A estruturação dos objetos de uma MIB é definida por uma especificação denominada SMI (*Structure of Management Information*), padronizada pela RFC 1155 **[48]**. Ela define aspectos importantes da MIB, tais como:

- Características dos objetos, tais como tipos permitidos, estruturas de dados, sintaxe, valores, entre outros;
- A organização dos objetos na MIB, visando a sua identificação;
- As regras para codificação dos objetos da MIB na transmissão entre gerente e agente.

A filosofia básica da SMI é definir uma estrutura simples e extensível para a MIB, visando facilidade de implementação, interoperabilidade entre produtos de diversos fabricantes e simplicidade do protocolo de gerenciamento.

Para atingir estes objetivos, a SMI exige que todos os objetos da MIB sejam definidos através de um subconjunto da linguagem formal para a descrição de dados ASN.1 (*Abstract Sintax Notation* 1), padronizada pelas normas ISO 8824 e ITU-T X.208. Através da ASN.1, a forma e o conteúdo dos objetos são definidos de maneira não-ambígua, garantindo-se que não haja interpretações diferentes para as suas especificações.

2.1.1.1 Estrutura da MIB

Para a identificação dos objetos de uma MIB, é definida uma árvore de nomeação global. Os objetos da MIB constituem uma estrutura em árvore, possuindo cada objeto um identificador único na mesma. Esta árvore de identificadores é administrada de forma conjunta pela ISO e pelo ITU, que delegam a outras instituições o controle dos identificadores nos nós intermediários da árvore, de acordo com a estrutura hierárquica entre estas instituições.

É interessante que se verifique a posição em que se encontra a árvore da MIB dentro da árvore de identificadores ISO-ITU. Partindo da raiz desta árvore saem três nós, denominados *iso*, *itu* e *joint-iso-itu*. Um dos nós da subárvore *iso*, denominado *org*, é reservado para o uso de outras

organizações; sob o nó *org* foi alocado um nó para o Departamento de Defesa norte-americano, denominado *dod* (*Department Of Defense*), e sob este há o nó *internet*, de responsabilidade do IAB (*Internet Architecture Board*). Esta organização pode ser vista na Figura 2-4 **[51]**.

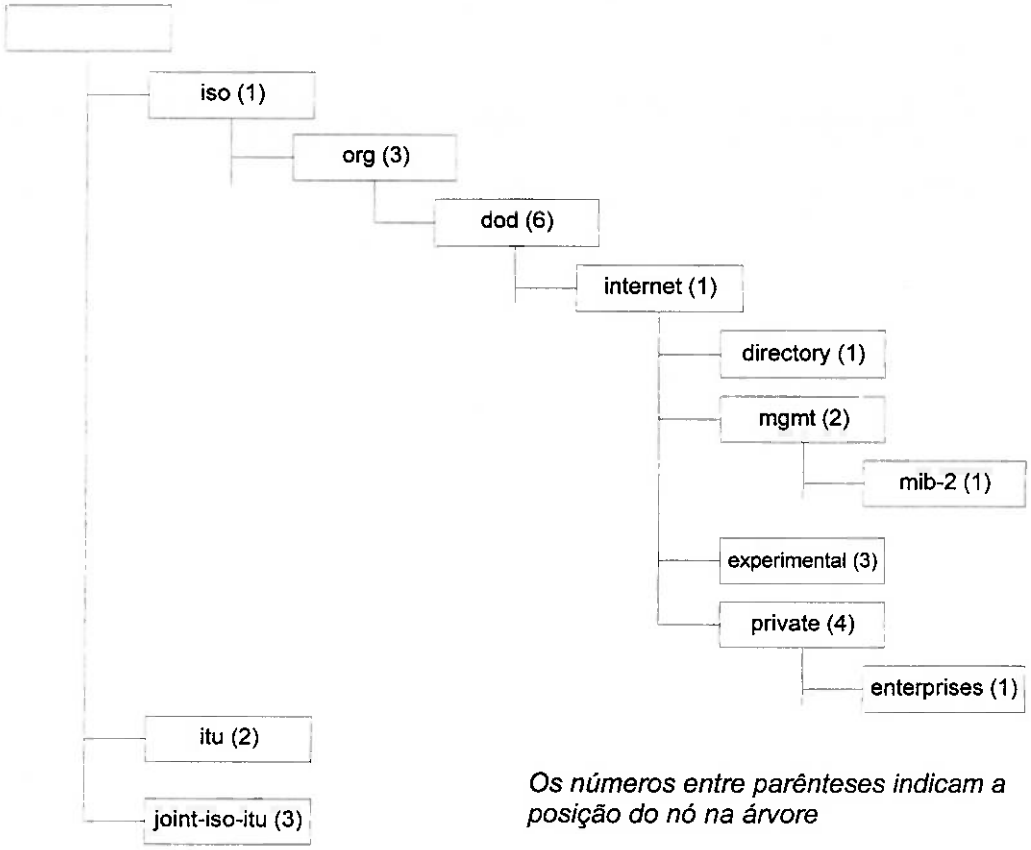


Figura 2-4: Árvore de identificadores da ISO-ITU, com o caminho até a subárvore mib-2

Por fim, o IAB definiu quatro nós sob a sua subárvore:

- *Directory:* reservado para o serviço de diretório X.500;
- *Mgmt:* utilizado para objetos relacionados a gerenciamento;
- *Experimental:* utilizado para objetos empregados em experiências com novos protocolos na Internet;

- *Private:* utilizado para identificar objetos de caráter privado. Possui uma subárvore importante, denominada *enterprises*, onde cada nó é de responsabilidade de uma empresa. Estes nós são utilizados pelas empresas para a extensão de objetos da MIB, visando representar recursos particulares de seus produtos.

A subárvore mgmt contém as definições das MIBs já aprovadas pelo IAB; atualmente a versão corrente é a MIB-II, cujas definições de objetos ficam dentro da subárvore mib-2.

A divisão da subárvore *internet* fornece flexibilidade na definição de novos objetos na MIB: caso uma nova tecnologia seja criada, os seus objetos devem ficar sob a subárvore *experimental* para testes, podendo ser movidos para a subárvore oficial de gerenciamento (*mgmt*) quando necessário. Analogamente, as extensões proprietárias da MIB por parte dos fabricantes ficam nos seus respectivos nós sob *enterprises*, sem interferir na MIB oficial.

Um último ponto a ser observado nesta estrutura de identificadores é o identificador numérico dos nós. À medida em que os nós são definidos, eles recebem um identificador numérico, correspondente à ordem em que eles foram criados (tais identificadores numéricos podem ser vistos na Figura 2-4).

O identificador único de um dado objeto é definido pela sua posição nesta árvore de nomeação e é denominado OID (*Object Identifier*) . Sendo assim, o OID do nó *mib-2* é:

iso . org . dod . internet . mgmt . mib-2

ou então, utilizando os identificadores numéricos dos nós:

1 . 3 . 6 . 1 . 2 . 1

2.1.1.2 Objetos

Um ponto importante na definição de uma MIB é o conceito de objeto. Os objetos de uma MIB são variáveis com tipos, valores e atributos rigidamente definidos pela SMI. No entanto, apesar da denominação, a MIB SNMP não é orientada a objetos (ou seja, sobre os objetos da MIB não se aplicam conceitos como herança, polimorfismo, métodos, entre outros); tal nomenclatura é

derivada do modelo OSI, onde a MIB correspondente possui uma estrutura extremamente complexa, orientada a objetos.

A SMI define apenas duas estruturas de dados para os objetos dentro da MIB: escalares ou matrizes bidimensionais de escalares (tabelas de escalares). No caso das tabelas, cada coluna é tratada como um objeto, recebendo o nome de objeto colunar.

Além das estruturas de dados, a SMI padroniza os tipos dos objetos permitidos, definindo como devem ser os valores de cada tipo, bem como o seu comportamento em relação à implementação da MIB (a Tabela 2-2 contém os principais tipos especificados na SMI). Podem-se definir novos tipos para os objetos, utilizando-se os tipos especificados pela SMI como base. Além disso, pode-se restringir os valores de um objeto, como por exemplo, definir uma lista de valores permitidos (ou seja, uma enumeração) ou definir o número máximo de caracteres de uma *string* **[48][51]**.

Tipo	**Descrição**
Integer	Objeto contém um valor inteiro.
Octectstring	Objeto contém uma *string* de caracteres.
Null	Objeto contém um valor nulo ou vazio.
Object identifier	Objeto contém o OID de um objeto.
Sequence	Objeto contém outros objetos; sendo utilizado para a criação de tabelas.
Ipaddress	Objeto contém um endereço IP, representado através de 32 bits.
Counter	Objeto contém um valor inteiro não-negativo que pode ser incrementado mas não decrementado. Quando valor máximo é atingido, o valor do objeto retorna a zero e começa a crescer novamente.
Gauge	Objeto contém um valor inteiro não-negativo que pode ser incrementado e decrementado. Quando valor máximo é atingido, o valor do objeto fica travado neste valor até ser reinicializado.
Timeticks	Objeto contém um valor inteiro não-negativo que conta os centésimos de segundo a partir de um dado evento (por exemplo, desde a última inicialização do dispositivo).
Opaque	Objeto contém um dado arbitrário.

Legenda: IP *Internet Protocol*
OID *Object Identifier*

Tabela 2-2: Principais tipos permitidos pela SMI para os objetos de uma MIB

Cada objeto possui alguns atributos associados. Entre os principais atributos definidos pela SMI estão o *status* de implementação do objeto e o modo de acesso ao objeto. Para objetos colunares, há um atributo importante, que indica se os valores da coluna são únicos, devendo ser utilizados como índices para linhas da tabela.

O modo de acesso ao objeto indica se o gerente pode ou não acessar o objeto, e como pode ser feito este acesso. Os modos permitidos podem ser vistos na Tabela 2-3.

Modo de Acesso	**Descrição**
Read-only	Valor do objeto pode ser lido, porém não pode ser alterado.
Read-write	Valor do objeto pode ser lido e alterado.
Write-only	Valor do objeto pode ser alterado somente, não podendo ser lido.
Not-accessible	Valor do objeto não pode ser lido nem alterado (tipicamente representa um nó na árvore).

Tabela 2-3: Modo de acesso a um objeto definido pela SMI

Se um agente SNMP implementar uma dada MIB, o atributo de *status* de implementação do objeto indica quais objetos o agente deve suportar de forma obrigatória ou opcional.

***Status* de Implementação**	**Descrição**
Mandatory	O agente deve obrigatoriamente suportar o objeto.
Optional	O agente pode ou não suportar o objeto.
Obsolete	O agente não deve suportar o objeto. Refere-se a um objeto que era obrigatório ou opcional em versões anteriores da MIB.
Deprecated	O agente deve suportar o objeto, porém provavelmente deve ser removido na próxima versão da MIB.

Legenda: MIB *Management Information Base*

Tabela 2-4: *Status* de implementação de um objeto definido pela SMI

Por fim, a SMI exige que os objetos da MIB sejam codificados de forma especial nos pacotes de dados para a transmissão, utilizando-se as regras definidas pela especificação BER (*Basic Encoding Rules*), padronizada pelas normas ISO 8825 e ITU-T X.209. A BER contém um conjunto de regras que especificam como devem ser convertidos os elementos descritos pela sintaxe abstrata ASN.1 em um conjunto de bytes para transferência entre o gerente e o agente.

Estas regras padronizam a representação dos dados na transmissão, independente de como estes dados são representados internamente nos dispositivos (independente da arquitetura interna destes dispositivos).

2.1.2 MIB-II

A RFC 1213 **[49]** contém a definição da MIB-II, a segunda versão da base de dados de gerenciamento da arquitetura TCP/IP. Assim como a primeira versão, a MIB-II define quais os objetos são essenciais em um dispositivo TCP/IP e devem ser suportados em todos os dispositivos gerenciáveis via SNMP.

Na MIB-II, os objetos encontram-se divididos em 10 grupos, de acordo com a afinidade de suas funções (Tabela 2-5) **[14][48][51]**.

Grupo	Descrição
system	Informações gerais sobre o sistema. Por exemplo, o objeto *SysDescr* contém uma descrição do dispositivo gerenciado, definida pelo fabricante.
interfaces	Informações de cada interface de rede do sistema. Por exemplo, o objeto *IfInOctects* contém o número total de bytes recebidos por uma dada interface de rede.
at (address translation)	Informações para o mapeamento de endereços IP das interfaces em endereços físicos. Sua implementação foi considerada desnecessária pela MIB-II, sendo mantida em compatibilidade com a MIB-I. Por exemplo, o objeto *atPhysAddress* contém um endereço físico relacionado a um dado endereço IP.
ip	Informações relacionadas à implementação e execução do protocolo IP no sistema gerenciado. Por exemplo, *ipForwarding* indica se o equipamento está atuando como um roteador ou não.
icmp	Informações relacionadas à implementação e execução do protocolo ICMP no sistema gerenciado. Por exemplo, *icmpInMsgs* contém o número total de mensagens ICMP recebidas.
tcp	Informações relacionadas à implementação e execução do protocolo TCP no sistema gerenciado. Por exemplo, *tcpCurrEstab* contém o número total de conexões TCP estabelecidas.
udp	Informações relacionadas à implementação e execução do protocolo UDP no sistema gerenciado. Por exemplo, *udpInDatagrams* contém o número total de pacotes UDP recebidos.
egp	Informações relacionadas à implementação e execução do protocolo EGP (*Exterior Gateway Protocol*) no sistema gerenciado. Por exemplo, *egpInMsgs* contém o número total de mensagens EGP recebidas sem erros.
transmission	Informações relacionadas a esquemas de transmissão e protocolos de acesso aos meios físicos presentes nas interfaces do sistema. Por exemplo, a subárvore *dot3* contém objetos relacionados ao gerenciamento de interfaces Ethernet.
snmp	Informações relacionadas à implementação e execução do próprio protocolo SNMP no sistema gerenciado. Por exemplo, *snmpInPkts* contém o número total de mensagens SNMP recebidas.

Legenda:

EGP	*Exterior Gateway Protocol*
IP	*Internet Protocol*
ICMP	*Internet Control Message Protocol*
MIB	*Management Information Base*
SNMP	*Simple Network Management Protocol*
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*

Tabela 2-5: Grupos de objetos da MIB-II

Pode-se perceber que cada grupo de objetos é responsável pelo gerenciamento de um conjunto de recursos correlacionados. Se um elemento da rede implementa um dado conjunto de recursos, os grupos correspondentes são considerados, de acordo com a definição da MIB-II, de implementação essencial, assim como os seus objetos. Por exemplo, no caso de um dado equipamento implementar o protocolo TCP, sua MIB deve possuir todos os objetos pertencentes ao grupo *tcp*; já a MIB de uma ponte não precisaria possuir os grupos *tcp e egp*, pois ela não possui os protocolos correspondentes implementados.

A Figura 2-5 mostra a posição dos grupos de objetos dentro da MIB-II **[14][48][51]**.

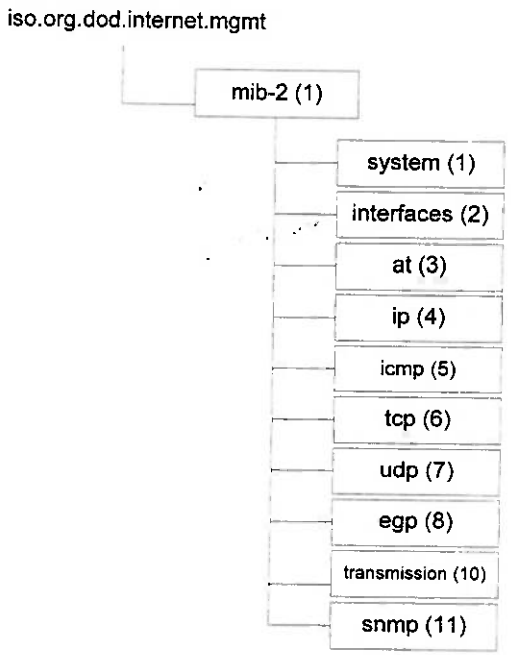


Figura 2-5: Árvore de grupos da MIB-2 com os respectivos identificadores numéricos

2.1.2.1 Extensões da MIB

A MIB-II contém objetos que são comuns a qualquer sistema gerenciável via SNMP que implemente parte da arquitetura de protocolos TCP/IP. No entanto, estes objetos nem sempre são suficientes para o gerenciamento das funcionalidades dos diversos elementos de uma

rede. Não há, na MIB-II, objetos para a monitoração de características físicas de equipamentos de rede ou em nível de camada de aplicação como, por exemplo:

- Um *no-break* (ou UPS – *Uninterruptible Power Supply*), apesar de não ter nenhuma função de comunicação na rede, é um elemento importante. É interessante poder monitorar a sua carga e a sua autonomia (o tempo que ele pode fornecer energia aos equipamentos a ele ligados em caso de falta de luz), entre outros;
- Em um servidor, é interessante saber quanto de espaço em disco há disponível ou o nível de processamento sendo exigido deste servidor;
- Em um servidor de *e-mails* pode-se monitorar estatísticas de maneira a atestar o seu funcionamento, tais como: número de mensagens recebidas e enviadas ou mensagens que não foram enviadas.

Para atender a estes requisitos, foram definidas diversas extensões para a MIB-II; cada uma destas extensões define uma nova MIB. Um elemento gerenciável deve obrigatoriamente suportar a MIB-II e pode, opcionalmente, suportar algumas extensões. A Tabela 2-6 exibe algumas MIBs definidas em grupos de trabalho do IETF **[34][51]**.

RFC	Descrição
1253	*OSPF Version 2 Management Information Base.*
1493	*Definitions of Managed Objects for Bridges.*
1512	*FDDI MIB.*
1514	*Host Resources MIB.*
1628	*UPS MIB.*
1643	*Definitions of Managed Objects for the Ethernet-like Interface Types.*
1696	*Modem MIB.*
1697	*Relational Database Management System MIB.*
1724	*RIP Version 2 MIB Extension.*
1749	*IEEE 802.5 Token Ring MIB.*
1759	*Printer MIB.*
2249	*Mail Monitoring MIB.*

Legenda: FDDI *Fiber Distributed Data Interface*
IEEE *Institute of Electrical and Eletronics Engineers*
MIB *Management Information Base*
OSPF *Open Shortest-Path First*
RIP *Routing Information Protocol*
UPS *Uninterruptible Power Supply*

Tabela 2-6: Exemplos de MIBs definidas pelo IETF

Após a publicação da RFC 1213, decidiu-se que os grupos da MIB-II seriam atualizados de forma independente por grupos de trabalhos distintos dentro do IETF. Alguns grupos de objetos, tais como o grupo *interfaces* e o *ip*, possuem extensões visando atender a evolução dos recursos por eles representados **[51]**.

1.1.1.1 MIBs Privativas

Tanto na MIB-II como nas várias outras MIBs padronizadas pelo IETF são definidos objetos que endereçam recursos genéricos de elementos de rede. No entanto, cada fabricante coloca, geralmente, em seus equipamentos recursos exclusivos e proprietários; para gerenciá-los deve-se definir uma MIB que os represente. Estas MIBs são denominadas MIBs privativas e devem ser definidas dentro do nó *iso.dod.internet.private.enterprises,* onde há uma subárvore de nomeação para cada fabricante que implementa uma extensão proprietária. A Figura 2-6 exibe a hierarquia de nomeação para MIBs privativas, com a posição de alguns fabricantes na subárvore correspondente **[14][33]**.

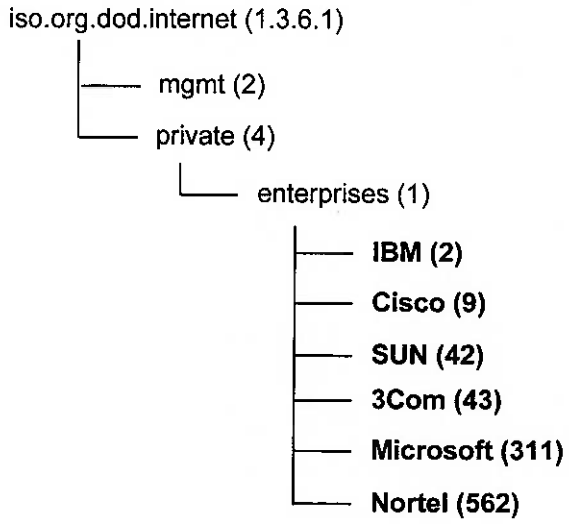


Figura 2-6: Hierarquia de nomeação para MIBs privativas

Para a manipulação dos objetos de uma MIB, o gerente necessita saber como ela está definida, a localização de seus objetos na árvore, os modos de acesso e tipo de dados de cada objeto, entre outros. No entanto, normalmente o gerente não conhece as MIBs privativas dos fabricantes, sendo necessário "aprender" as definições destas MIBs. Este procedimento é denominado de compilação da MIB: o gerente converte a descrição formal da MIB privativa (em ASN.1) em um formato interno, "aprendendo" a sua estruturação e sendo, então, capaz de manipular os seus objetos.

1.1.2 Protocolo SNMP

O protocolo SNMP é o protocolo de gerenciamento utilizado para a comunicação entre um gerente e os agentes a ele subordinados. Ele é um protocolo simples, fornecendo um mecanismo básico para esta comunicação.

1.1.2.1 Comunidades

Dentro do modelo SNMP, o conjunto formado por agentes e seus respectivos gerentes é denominado uma **comunidade SNMP**. O conceito de comunidade está relacionado à

segurança em relação ao acesso dos objetos de uma MIB: um agente só permite o acesso aos objetos de sua MIB a gerentes da mesma comunidade à qual ele pertence.

Associada a uma comunidade SNMP existe uma política de acesso, denominada de Perfil da Comunidade. São duas as características associadas ao perfil da comunidade **[14][51]**:

- Visão da MIB: corresponde a um subconjunto de objetos da MIB que pode ser acessado pelos gerentes da comunidade;
- Modo de Acesso: especifica o modo de acesso (*read-only* ou *read-write*) permitido para gerentes da comunidade.

Como um agente pode pertencer a mais de uma comunidade, através do perfil de comunidade, pode-se restringir as funcionalidades que cada gerente pode executar sobre o dispositivo. Pode-se definir, por exemplo, um gerente com acesso total ao dispositivo e outro com um acesso restrito.

2.1.2.2 Primitivas SNMPv1

Como as operações sobre elementos gerenciados são apenas sobre os valores dos objetos de suas respectivas MIBs, o protocolo de gerenciamento não necessita ser complexo; sendo assim, o protocolo SNMP possui apenas cinco primitivas, conforme pode ser visto na Tabela 2-7 **[16][51]**:

Primitiva	**Descrição**
GetRequest	Utilizada pelo gerente para obter o valor de um ou mais objetos da MIB do agente.
GetNextRequest	Utilizada pelo gerente para obter o valor de um ou mais objetos da MIB do agente. Similar à primitiva *GetRequest*, o agente deve retornar não o valor dos objetos requisitados pelo gerente, mas o valor dos objetos seguintes aos objetos requisitados.
SetRequest	Utilizada pelo gerente para requisitar a um agente a modificação do valor de um ou mais objetos da MIB.
GetResponse	Utilizada pelo agente para devolver ao gerente o resultado das operações *GetRequest*, *GetNextRequest* e *SetRequest*.
Trap	Utilizada pelo agente para notificar o gerente sobre a ocorrência de um evento significativo (geralmente um problema na rede)

Legenda: MIB *Management Information Base*

Tabela 2-7: Primitivas do protocolo SNMPv1

Todas estas primitivas possuem como parâmetros: o nome da comunidade a qual pertence o gerente e o nome dos objetos aos quais se referem a primitiva, bem como os seus valores (no caso das primitivas SetRequest, *GetResponse* e *Trap*). Para objetos colunares, o gerente deve fornecer também o valor do objeto (ou objetos) índice da tabela.

Através destas primitivas, o protocolo SNMP possui três funções básicas: leitura de dados, atualização de dados e notificação. A relação entre as primitivas, bem como as suas funções podem ser vistas na Figura 2-7 **[14]**.

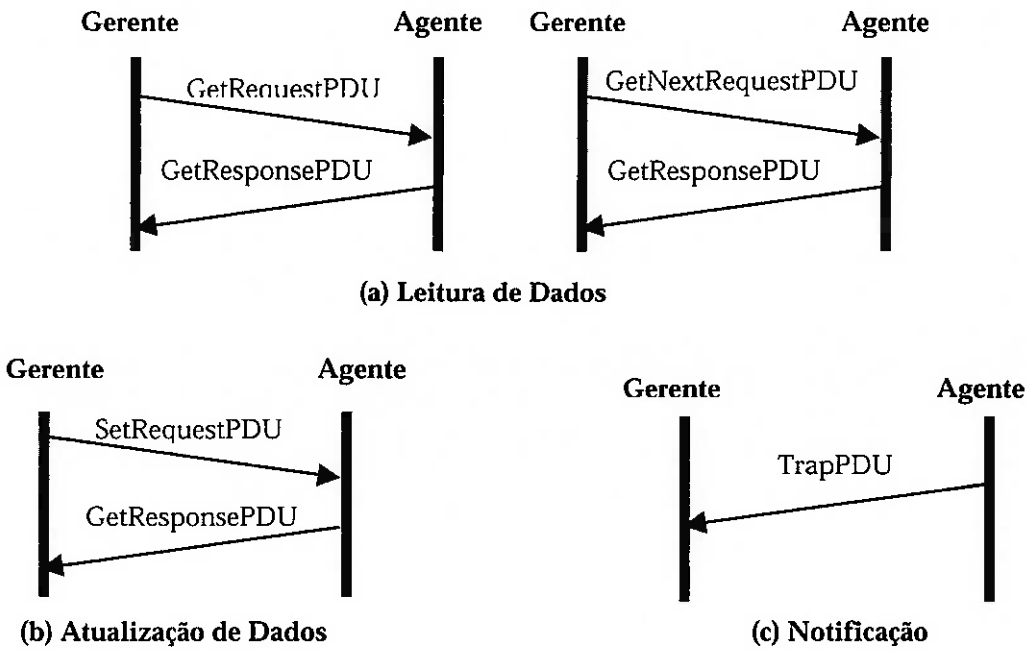


Figura 2-7: Relação entre as primitivas do protocolo SNMP

As primitivas *GetRequest* e *GetNextRequest* são utilizadas na leitura de informações da MIB do agente, que devolve ao gerente, como resposta, uma primitiva *GetResponse.* Enquanto, em uma requisição *GetRequest,* o agente devolve o valor dos objeto especificados da MIB, em uma *GetNextRequest* o agente devolve o nome e o valor dos objetos seguintes aos especificados. A sua importância é bastante sutil, sendo usada em duas situações:

- Quando o gerente não conhece a MIB à qual ele se refere;
- Em tabelas, quando o gerente não conhece o seu conteúdo (ou seja, os índices da tabela).

Utilizando a primitiva *GetNextRequest,* o gerente pode, recursivamente, percorrer a MIB ou uma tabela, descobrindo os seus objetos.

Para a atualização dos dados do agente, o gerente utiliza a primitiva *SetRequest.* Através da resposta do agente (que responde com um *GetResponse*), o gerente pode verificar se o valor foi alterado ou não.

Por fim, para a notificação de eventos, o agente utiliza a primitiva *Trap,* não havendo nenhuma requisição ou confirmação do gerente.

O mecanismo básico das operações do SNMP baseia-se na realização de *polling* **[14][51]**. Caso o gerente deseje obter informações dos recursos gerenciados, ele deve, periodicamente, requisitar o valor dos objetos da MIB que os representem e, baseado nestes valores, decidir se o dispositivo está operando normalmente.

O gerente pode enviar, em uma única mensagem, uma lista dos objetos a serem lidos ou alterados. No entanto, a operação é atômica, ou seja, se um dos objetos da lista não puder ser lido ou alterado, nenhum dos objetos da lista é lido ou alterado.

Através deste conjunto de primitivas pode-se perceber o caráter simplista do SNMP (por exemplo, não há primitivas para a criação e remoção de objetos de uma MIB, sendo os objetos fixos na MIB). Enquanto que esta "pobreza" de funções pode limitar o seu uso, a estrutura definida pelo protocolo é bastante genérica, capaz de ser aplicada a uma gama enorme de equipamentos, das mais variadas arquiteturas.

2.1.3 Comentários sobre o SNMP

Dado o seu caráter simplista, a arquitetura SNMP original apresenta alguns problemas. Um dos principais está relacionado ao desempenho em redes com muitos elementos gerenciados. A utilização do mecanismo de *polling* (operações de *get* e *set*) é inadequada para redes muito grandes: como o gerente deve periodicamente monitorar os valores dos objetos considerados críticos, nestes tipos de rede o tráfego de gerenciamento pode tornar-se excessivo, acarretando problemas de desempenho na rede.

O protocolo apresenta-se, também, inadequado para transferências de grandes volumes de dados. Para a transferência de uma tabela com muitas linhas, deve-se transferir uma linha por vez, através do *GetNextRequest,* gerando um *overhead* de requisições e confirmações trafegados na rede.

Apesar do protocolo permitir a utilização de diversos gerentes, não há suporte à comunicação gerente-gerente. Sendo assim, para viabilizar a correlação das informações de diversos elementos da rede dentro da arquitetura SNMP, o gerenciamento deve ser feito de forma centralizada, impondo à estação de gerenciamento uma carga grande de processamento. Além disso, centralizando-se o gerenciamento em uma rede grande, com diversos *sites* interligados por uma rede de longa distância (WAN – *Wide Area Network*), o tráfego de *polling* do gerente pode, facilmente, sobrecarregar os enlaces da WAN.

Um outro problema está relacionado com a baixa confiabilidade das *traps*. Como as *traps* utilizam o protocolo UDP e não há nenhuma primitiva que indique para o agente que o gerente recebeu a *trap* enviada, não há nenhuma garantia que o gerente tomou conhecimento de um evento importante. Sendo assim, o gerente não pode confiar exclusivamente no envio de *traps* para detectar de forma rápida anomalias na rede, devendo para isso executar o *polling* em intervalos pequenos, aumentando o tráfego de gerenciamento.

Por fim, a segurança definida pela arquitetura é bastante fraca, havendo apenas um mecanismo de autenticação baseado no nome da comunidade que é enviado junto com as primitivas. Não há, no entanto, nenhum mecanismo de criptografia associado ao SNMP, sendo que o nome de comunidade é passado aberto e pode ser facilmente observável através de analisadores de protocolo. Descobrir os nomes de comunidade associados aos agentes da rede pode dar a um invasor poderes equivalentes ao do administrador da rede (dada a capacidade da arquitetura em alterar, de forma remota, a configuração dos equipamentos), o que pode resultar em uma ameaça bastante séria. Para limitar os riscos de segurança inerentes da arquitetura SNMP, em muitas redes os administradores configuram os agentes para não aceitar operações de *SetRequest*, restringindo o protocolo apenas à monitoração dos elementos.

Para resolver estas limitações, foram definidas outras versões do protocolo, bem como versões novas da SMI e extensões da MIB-II. Entre estas extensões da MIB-II, destacam-se as MIBs RMON1 e RMON2 (*Remote MONitoring*) **[51]**. Os agentes que as implementam, denominados

probes RMON, são capazes de coletar informações periódicas do tráfego da rede, sem necessitar de *pollings* periódicos do gerente; além disso, são capazes de analisar os dados coletados e enviar para o gerente apenas o resultado destas análises, quando necessário. Com isso, os *probes* RMON diminuem não só o tráfego de informações de gerenciamento na rede como também aliviam a demanda de recursos computacionais da estação de gerenciamento. No entanto, como o RMON e o RMON2 foram definidos dentro de um escopo específico, seus benefícios são sentidos somente na monitoração de tráfego e na resolução de problemas detectados a partir desta monitoração.

2.1.4 SNMPv2

Para tentar resolver alguns dos problemas da arquitetura SNMP, foi definida uma segunda versão da arquitetura, denominada SNMPv2 e especificada pelas RFCs da Tabela 2-8 **[34][51]**. Com esta nova versão, a SMI e o protocolo SNMP definidos na Tabela 2-1 passaram a ser referidos, respectivamente, por SMIv1 e SNMPv1.

RFC	***Status***	**Título**
1901	*Experimental*	*Introduction to Community-Based SNMPv2* ***[17]***
2578	*Standard – STD 58*	*SMI for SNMPv2* ***[18]***
2579	*Standard – STD 58*	*Textual Conventions for SNMPv2* ***[19]***
2580	*Standard – STD 58*	*Conformance Statements for SNMPv2* ***[20]***
1905	*Draft*	*Protocol Operations for SNMPv2*
1906	*Draft*	*Transport Mappings for SNMPv2*
1907	*Draft*	*MIB for SNMPv2*
1908	*Draft*	*Coexistence between SNMPv1 and SNMPv2*

Legenda: MIB *Management Information Base*
SNMP *Simple Network Management Protocol*
SMI *Structure of Management Information*

Tabela 2-8: RFCs da especificação do SNMPv2

Entre as principais mudanças em relação ao SNMPv1, encontram-se:

- Em adição ao gerenciamento centralizado, o SNMPv2 permite a definição de um esquema de gerenciamento distribuído, com elementos atuando ao mesmo tempo

como gerente (de um grupo de agentes) e agente (de um gerente superior), podendo-se definir uma hierarquia de gerentes;

- O protocolo SNMP foi alterado, incorporando-se uma nova primitiva para comunicação entre gerentes, bem como uma primitiva para a transmissão de grandes volumes de dados de uma forma mais eficiente;
- A nova SMI define novos tipos de objetos, bem como refinamentos das especificações existentes;
- A nova MIB apresenta novos objetos, de maneira a suportar as novas funcionalidades do padrão.

As diversas alterações propostas para a arquitetura tornam gerentes e agentes SNMPv2 incompatíveis com os gerentes e agentes que utilizam as especificações do SNMPv1. Devem, portanto, ser definidas estratégias para a migração do SNMPv1 para o SNMPv2, bem como para uma coexistência entre estes protocolos. De acordo com a RFC 1908, as seguintes estratégias podem ser utilizadas para este fim:

- Gerentes bilíngües, capazes de se comunicar com os agentes utilizando tanto o SNMPv1 como o SNMPv2;
- Agentes *proxy*, capazes de converter requisições SNMPv2 de gerentes para agentes SNMPv1.

2.1.4.1 SMIv2

Através de novos elementos da ASN.1, a SMIv2 **[18]** permite a descrição formal de uma MIB de forma mais clara e concisa, eliminando não só ambigüidades que podem ocorrer com descrições feitas a partir da SMIv1, como também permitindo que a descrição da MIB seja mais inteligível para seres humanos.

Entre as alterações introduzidas, visando melhorar a documentação da MIB e facilitar o seu entendimento por seres humanos, têm-se:

- Definições de módulo: através da macro ASN.1 MODULE-IDENTITY, documentam-se, na MIB, informações de caráter geral, tais como: entidade e indivíduo responsável pela mesma, o histórico de revisões, descrição das funcionalidades, entre outros;
- Convenções Textuais: através do elemento ASN.1 TEXTUAL-CONVENTION, é possível criar novos tipos para os objetos da MIB similares aos definidos na SMI, porém com nome diferente, sintaxe similar e semântica mais precisa **[19]**.

Além destas alterações, foram definidas duas macros para a especificação de objetos e notificações (*traps*), denominadas respectivamente OBJECT-TYPE e NOTIFICATION-TYPE. Tais macros definem a sintaxe e a semântica destes elementos de forma concisa, permitindo não só uma leitura fácil como uma fácil interpretação pelos aplicativos de gerenciamento na compilação da MIB.

Além de eliminar ambigüidades, os tipos dos dados permitidos foram alterados também para atender aos requisitos de novas tecnologias. A Tabela 2-9 exibe os tipos definidos pela SMIv2, comparando-os com os tipos definidos pela SMIv1 **[18][51]**.

Tipo SMIv1	**Tipo SMIv2**	**Comentário**
Integer	*Integer* *Integer32*	Os tipos *Integer* e *Integer32* são definidos dentro do intervalo [2^{31}, 2^{31} –1] na SMIv2. O tipo *integer32* é igual ao *Integer*.
Octectstring	*Octectstring*	O tipo *OctectString* pode ter um tamanho máximo 2^{16} –1 caracteres na SMIv2.
Null	——	O tipo *Null foi r*emovido da SMIv2.
Counter	*Counter32* *Counter64*	Os tipos *Counter32* e *Counter64* possuem o mesmo comportamento do *Counter*. A diferença ocorre no valor máximo: o *Counter32* e o *Counter* possuem um valor máximo de 2^{32} –1, enquanto o do *Counter64* é de 2^{64} –1.
——	*Unsigned32*	O tipo *Unsigned32 é* definido para valores inteiros dentro do intervalo [0, 2^{32} –1]. Foi adicionado pela SMIv2.
——	*BITS*	O tipo BITS contém uma enumeração de bits. Foi adicionado pela SMIv2.
Gauge	*Gauge32*	O tipo Gauge32 é equivalente ao Gauge.
Timeticks	*Timeticks*	Os tipos *Timeticks*, *Object identifier*, *Sequence* e *Ipaddress* são iguais aos tipos de mesmo nome definidos na SMIv1.
Object identifier	*Object identifier*	
Sequence	*Sequence*	
Ipaddress	*Ipaddress*	
Opaque	*Opaque*	O tipo Opaque foi mantido para garantir compatibilidade com a SMIv1.

Legenda: SMI *Structure of Management Information*

Tabela 2-9: Comparação entre tipos definidos na SMIv2 e na SMIv1

A SMIv2 permite também que linhas em uma tabela sejam criadas e removidas dinamicamente pelo gerente; tal característica não era especificada pela SMIv1. Esta funcionalidade, associada a uma definição mais precisa das notificações, demandaram alterações no atributo de modo de acesso ao objeto, conforme pode ser visto na Tabela 2-10 **[18][51]**.

SMIv1	**SMIv2**	**Comentário**
Read-only	*Read-only*	Modos de acesso mantidos da definição da SMIv1.
Read-write	*Read-write*	
Write-only	——	Modos de acesso removidos da SMIv2.
Not-accessible	——	
——	*Read-create*	Modo de acesso adicionado pela SMIv2. Indica que o objeto pode ser lido, alterado e criado, sendo utilizado para a criação de linhas em tabelas.
——	*Accessible-for-notify*	Modo de acesso adicionado pela SMIv2. Indica que o objeto não pode ser acessado por operações de *Get* ou *Set*, sendo o seu valor acessado somente através de uma notificação.
Legenda:	SMI	*Structure of Management Information*

Tabela 2-10: Modos de acesso a um objeto definidos pela SMIv2

Para definir se um objeto deve ou não ser implementado, foram introduzidas na arquitetura declarações de conformidade (*Conformance Statements*) **[20]**. Através destas declarações, os objetos e as notificações de uma MIB são reunidas em grupos de conformidade, definindo-se assim níveis de implementação. Portanto, a implementação de uma MIB deve indicar quais grupos de conformidade foram implementados e, com esta informação, o gerente pode determinar a capacidade do agente.

Com as declarações de conformidade, o atributo de *status* de implementação de um objeto passa a conter informações relacionadas à validade do objeto, ou seja, se ele é um objeto válido para a implementação atual ou se ele é um objeto histórico, que não deve ser implementado pelo agente. A Tabela 2-11 apresenta os *status* permitidos pela SMIv2.

SMIv1	SMIv2	Comentário
Mandatory	——	Atributos de *status* removidos da SMIv2; estas informações são de responsabilidade das declarações de conformidade da MIB.
Optional	——	
——	*Current*	Atributo de *status* adicionado pela SMIv2, indica que o objeto é valido para implementação.
Obsolete	*Obsolete*	Atributo de *status* mantido da SMIv1, indica que o objeto não deve ser implementado pelo agente.
Deprecated	*Deprecated*	Atributo de *status* mantido da SMIv1, indica objetos que devem ser implementados somente para fins de interoperabilidade com implementações antigas.
Legenda:	*SMI*	*Structure of Management Information*

Tabela 2-11: *Status* de implementação de um objeto, definidos pela SMIv2

2.1.4.2 SNMPv2-MIB

Para suportar as alterações introduzidas no protocolo, foram feitas também alterações em alguns grupos da MIB-II **[14] [51]**:

- *system*: expansão do grupo *system* da MIB-II, inclui objetos relativos a recursos que podem ser configurados dinamicamente pelos gerentes (por exemplo, o objeto *sysORLastChange*, que indica o instante em que ocorreu a última configuração do agente);
- *snmp*: diversos objetos do grupo *snmp* da MIB-II original foram removidos, por não trazerem benefícios concretos em ambientes de produção (por exemplo, o objeto *snmpOutgetRequests*, que indica o número de primitivas *GetRequests* enviadas pelo agente); foram, também, incluídos novos objetos, relativos a funcionalidades adicionais do SNMPv2 (por exemplo, *snmpProxyDrops*, que indica o número de primitivas SNMP que foram descartadas por um agente *proxy* SNMPv2, devido a problemas na conversão entre protocolos executada pelo *proxy*).

Além destes grupos, foi criado o grupo *snmpMIBOBjects,* que contém objetos relacionados à utilização de *traps* e à coordenação de *SetRequests* enviados por diversos gerentes.

2.1.4.3 Primitivas SNMPv2

No protocolo SNMPv2, foram incluídas duas novas primitivas, *GetBulkRequest* e *InformRequest,* além de se alterar o formato das outras primitivas. A Tabela 2-12 lista as primitivas do SNMPv2 **[14][51]**:

Primitiva	Descrição
GetRequest	Idêntica à *GetRequest* do SNMPv1.
GetNextRequest	Idêntica à *GetNextRequest* do SNMPv1.
GetBulkRequest	Utilizada para a transmissão de grande volume de dados: dado um objeto, o agente deve retornar o valor dos *n* objetos seguintes ao objeto requisitado, sendo o valor de *n* especificado pelo gerente.
InformRequest	Utilizada pelo gerente para requisitar a outro gerente o valor de um ou mais objetos da MIB do gerente.
Response	Equivalente à *GetResponse* do SNMPv1, é utilizada como resposta às operações *GetRequest, GetNextRequest, GetBulkRequest, SetRequest* e *InformRequest.*
SNMPv2-Trap	Equivalente à *Trap* do SNMPv1, porém com um formato de PDU diferente. É utilizada pelo agente para notificar o gerente de eventos inesperados (utiliza as mesmas definições de eventos do SNMPv1).

Tabela 2-12: Primitivas do protocolo SNMPv2

As operações das primitivas do SNMPv2 não são atômicas: o agente deve devolver os valores dos objetos que ele conseguiu ler e indicar quais objetos apresentaram problemas.

2.1.4.4 Comentários a respeito do SNMPv2

Apesar dos benefícios introduzidos pelo SNMPv2, alguns problemas ainda persistiram **[14][51]**:

- Baixa Confiabilidade das *Traps*: o protocolo opera, ainda, sobre UDP e não há confirmação do recebimento das *traps*;
- Segurança Fraca: apesar de terem sido propostos mecanismos de segurança adicionais ao SNMPv2, não houve um consenso quanto aos mecanismos a serem utilizados. O SNMPv2 utiliza, apenas, a autenticação baseada em nome de comunidade, de forma idêntica ao SNMPv1. Devido a este fato, o SNMPv2 é por vezes referenciado como SNMPv2C (onde o C da sigla representa *Community*).

Em relação ao processo de padronização do SNMPv2, apenas as alterações relativas às SMIv2 (incluindo as RFCs relativas a convenções textuais e declarações de conformidade), tornaram-se padrões efetivos (STD 58), devendo as novas MIBs serem descritas de acordo com as suas definições. As alterações relativas ao protocolo e à MIB encontram-se em fase final do processo de padronização e, apesar de se encontrar gerentes compatíveis com as especificações do protocolo SNMPv2, a maioria dos agentes utiliza do protocolo SNMPv1.

2.1.5 SNMPv3

Os principais elementos da arquitetura SNMP incluem **[52]**:

- As regras de definição de dados – SMIv1 / SMIv2;
- Os módulos de informações de gerenciamento – MIB-II;
- As operações do protocolo – SNMPv1 / SNMPv2;
- Segurança e administração.

O foco principal do SNMPv3 foi a incorporação de novos mecanismos de segurança para a arquitetura. Para permitir que diferentes mecanismos de segurança possam coexistir em um dado elemento da rede, bem como para permitir que novos mecanismos de segurança possam ser incorporados sem exigir novas atualizações da arquitetura, o SNMPv3 define uma organização interna para gerentes e agentes. Não são especificadas pela arquitetura SNMPv3 uma nova SMI, novas MIBs ou um novo protocolo; tais elementos são incorporados das especificações do SNMPv2.

A Tabela 2-13 contém a relação das RFCs da arquitetura SNMPv3 **[34][51][52]**.

RFC	*Status*	Nome
2571	*Draft*	*Architecture for SNMP Frameworks* ***[32]***.
2572	*Draft*	*Message Processing and Dispatching.*
2573	*Draft*	*SNMPv3 Applications.*
2574	*Draft*	*User-based Security Model.*
2575	*Draft*	*View-based Access Control Model.*
1905	*Draft*	*Protocol Operations for SNMPv2.*
1906	*Draft*	*Transport Mappings for SNMPv2.*
1907	*Draft*	*MIB for SNMPv2.*
2570	*Informational*	*Introduction to SNMPv3.*

Legenda: MIB *Management Information Base*
SNMP *Simple Network Management Protocol*

Tabela 2-13: RFCs da arquitetura SNMPv3

2.2. *Gerenciamento Web*

A idéia inicial que motivou o gerenciamento *web* é a de que o gerenciamento poderia ser executado através de um *browser*, a partir de qualquer máquina na Internet. No entanto, este conceito expandiu-se, de maneira que o gerenciamento *web* visa também **[14][35]**:

- Utilização de protocolos já existentes na *web*;
- Integração das diversas ferramentas de gerenciamento sob uma única interface, o *browser*.

Entre os principais benefícios da utilização de tecnologias *web* para o gerenciamento pode-se destacar **[14]**:

- Estrutura distribuída e hierárquica: a estrutura de *links* presentes em páginas HTML (*Hypertext Markup Language*) e em aplicações *web* permite a definição de uma hierarquia entre os servidores, de maneira que servidores de gerenciamento, mesmo não suportando gerenciamento distribuído, possam ser acessados de maneira hierárquica via *web*;
- Universalidade de acesso: o *browser* representa uma interface gráfica independente de plataforma, uma vez que há versões de *browsers* para praticamente todos os sistemas operacionais existentes (diversos sistemas operacionais já incluem um *browser* em sua

instalação padrão). Com isso, a console de gerenciamento não fica restrita à estação de gerenciamento local ou a versões de sistema operacional que a suportem;

- Segurança: os servidores e *browsers* possuem diversos mecanismos de segurança associados, evitando que aplicações *web* necessitem implementá-los. Pode-se integrar estes mecanismos de segurança às aplicações *web* de gerenciamento, sem grandes alterações na aplicação em si;

- Integração com Bases de Dados: diversos servidores *web* e mecanismos de aplicações *web* apresentam ferramentas para a integração com bases de dados, evitando-se que as aplicações sejam responsáveis pelo gerenciamento dos dados. Aplicações de gerenciamento *web* podem, então, concentrar-se nas operações de gerenciamento, sem se preocupar com o armazenamento e recuperação dos dados referentes a estas operações.

2.2.1 Segurança

Deve haver, em aplicações de gerenciamento, uma preocupação grande com segurança, uma vez que tais aplicações podem alterar, remotamente, o funcionamento dos dispositivos da rede, impedindo a comunicação em casos de configurações errôneas.

Com a popularização da *web*, há diversos mecanismos de segurança desenvolvidos para o acesso às paginas, sendo alguns destes mecanismos inclusive já incorporados no próprio protocolo HTTP (*Hypertext Transfer Protocol*). Estes mecanismos têm sido bastante utilizados, de maneira que aplicações *web* que deles fazem uso beneficiam-se do fato destes mecanismos estarem sendo exaustivamente testados e estudados em relação a problemas de segurança. Entre os mecanismos temos:

- Mecanismos de Autenticação: dentro do protocolo HTTP podem-se utilizar diversos mecanismos de autenticação, sendo que somente o modo de autenticação básico (baseado em usuário e senha sem criptografia) encontra-se padronizado (o protocolo é extensível, de maneira que novos mecanismos de autenticação podem ser

incorporados). A partir do servidor *web*, definem-se quais páginas necessitam de autenticação e quais são os mecanismos de autenticação a serem utilizados em cada página;

- Listas de controle de acesso: podem-se definir, no servidor *web,* quais páginas cada usuário pode acessar, permitindo-se definir perfis de segurança diferenciados para os diversos usuários;
- Criptografia: a maioria dos servidores suporta o chamado HTTP Seguro (HTTPS), que utiliza um protocolo de segurança, o SSL (*Secure Sockets Layer* **[13]**) para o tráfego dos dados criptografados;
- Certificados: através de certificados digitais, pode-se garantir a autoria de uma dada comunicação.

2.2.2 Tipos de arquiteturas para Gerenciamento *Web*

As arquiteturas de gerenciamento *web* existentes podem ser classificadas em dois tipos **[35] [14]**:

- Gerenciamento *web* com acesso direto;
- Gerenciamento *web* via *proxy*.

2.2.2.1 Acesso Direto

Em arquiteturas de gerenciamento *web* do tipo acesso direto, são embutidos servidores *web* nos dispositivos gerenciáveis. Com isto, páginas HTML e o protocolo HTTP são utilizados em substituição a interfaces do tipo emulação de terminal (por exemplo VT-100) e protocolos como TELNET, que são tradicionalmente utilizados para a configuração dos equipamentos. A Figura 2-8 ilustra o tipo de arquitetura de acesso direto **[14]**.

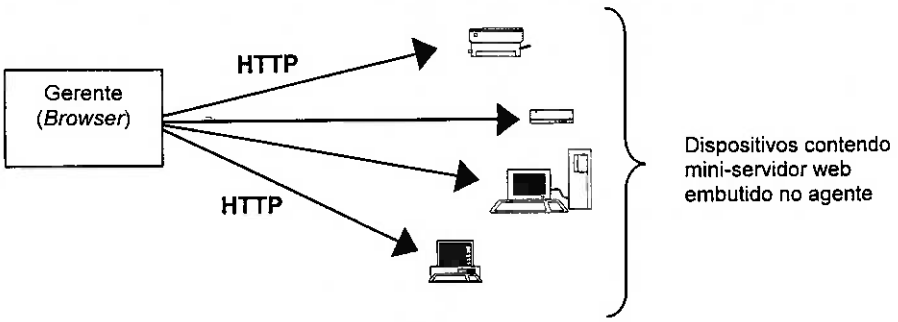


Figura 2-8: Arquitetura de Acesso Direto

Desta maneira, os equipamentos podem ser gerenciados graficamente através de um *browser*, sem necessitar de uma plataforma de gerenciamento ou de uma máquina dedicada para o gerenciamento. Por se acessar os dispositivos diretamente, não é possível obter uma visão global da rede, tendo-se uma visão individualizada da situação da rede. Tal solução pode apresentar também uma demanda maior por recursos computacionais no dispositivo gerenciado, dependendo da forma como são implementadas as aplicações de gerenciamento *web*. Além disso, para dispositivos mais antigos, pode ser necessário atualizá-los para suportar esta forma de gerenciamento.

2.2.2.2 Arquitetura *Proxy*

Em uma arquitetura de gerenciamento *web* com *proxy*, há um servidor dedicado, capaz de converter operações requisitadas através de páginas HTML em operações nos dispositivos gerenciáveis e devolver para o administrador as respostas formatadas de forma conveniente para apresentação em um *browser* (Figura 2-9) **[14]**.

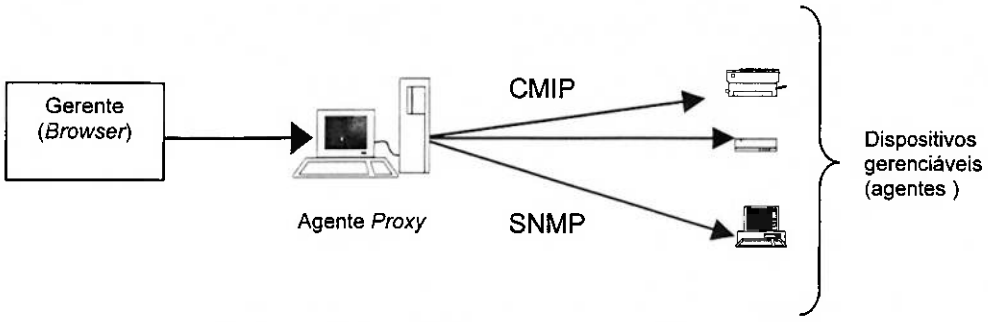


Figura 2-9: Arquitetura de Acesso Via *proxy*

Como as operações de gerenciamento em diversos dispositivos concentram-se no servidor *proxy*, pode-se integrar e correlacionar dados obtidos de diversos dispositivos, obtendo-se uma visão global da situação da rede. Além disso, pode-se integrar diferentes arquiteturas de gerenciamento, bastando para isto que o servidor *proxy* seja capaz de converter as operações requisitadas pelo administrador nas operações necessárias para cada dispositivo, de acordo com a arquitetura de gerenciamento suportada em cada dispositivo. Evita-se, também, a atualização dos dispositivos gerenciados para o suporte de gerenciamento *web*, uma vez que é o servidor *proxy* quem faz a adaptação do gerenciamento do dispositivo para gerenciamento *web*.

2.2.3 Padrões de gerenciamento via *Web*

De maneira a garantir a interoperabilidade de soluções de gerenciamento *web*, foram formados grupos para o desenvolvimento de padrões nesta área. Dois padrões encontram-se em processo de desenvolvimento **[14][46]**:

- WBEM: *Web-Based Enterprise Management;*
- JMX: *Java Management Extensions.*

2.2.3.1 WBEM – *Web-Based Enterprise Management*

O WBEM (*Web-Based Enterprise Management*) é uma arquitetura de gerenciamento baseada em padrões *web*. Sua padronização encontra-se sob responsabilidade do DMTF (*Distributed*

Management Task Force), uma entidade formada por empresas com o objetivo de desenvolver sistemas para gerenciamento de elementos de rede **[59]**.

O WBEM possui três objetivos principais:

- Acesso via *web* a dados de gerenciamento (em aplicativos ou servidores de gerenciamento);
- Acesso direto via *web* a elementos gerenciáveis;
- Integração de dados entre aplicativos de gerenciamento.

Para atingir estes objetivos, a arquitetura baseia-se em três elementos principais:

- Um modelo para a representação abstrata dos dados de gerenciamento, denominado CIM (*Common Information Model*) **[26][27]**;
- Um protocolo para o transporte de dados e mensagens, o HTTP **[13]**;
- Regras para a codificação dos dados e das operações sobre o protocolo HTTP. Tais regras de codificação utilizam o padrão XML (*eXtensible Markup Language*), sendo denominadas xmlCIM **[22][23]**.

No WBEM, os recursos dos dispositivos gerenciados são representados como instâncias de classes definidas pelo modelo CIM, ou seja, objetos do modelo. O CIM é um modelo orientado a objetos, onde é definida uma hierarquia de classes e subclasses, sendo possível a definição de propriedades (atributos) e métodos para as classes, bem como herança simples e relacionamento entre classes. O modelo é dividido em duas partes **[25]**:

- Especificação CIM: denominado também de CIM *Meta Schema*, é a definição formal do modelo. Contém os termos utilizados para a definição do modelo, a forma de utilizá-los, bem como a sua semântica, especificando-se os conceitos de classes, propriedades, métodos, entre outros **[27]**;

- CIM *Schema*: contém o modelo de gerenciamento, ou seja, as classes utilizadas para a representação de recursos gerenciados **[26]**. Encontra-se dividido em 3 camadas **[25][26]**:
  - *Core Schema*: classes cujos elementos contém informações genéricas para quaisquer áreas de gerenciamento;
  - *Common Schemas*: classes que representam recursos específicos de uma determinada área de gerenciamento, independentes de uma determinada tecnologia ou implementação. Como exemplos de *Common Schemas, pode*m-se citar classes para representação de sistemas, dispositivos, redes, aplicações, bases de dados, usuários, entre outros;
  - *Extension Schema*: Classes que representam extensões às classes do *Common Schema*, visando representar particularidades de uma determinada implementação ou tecnologia;

As classes são descritas utilizando-se uma linguagem formal denominada MOF *(Managed Object Format).* Quando uma aplicação de gerenciamento não conhece uma determinada classe, ela pode importar a sua definição, utilizando sua descrição formal (expressa em MOF). Desta maneira, uma aplicação de gerenciamento pode manipular objetos derivados de novas classes (uma extensão, por exemplo).

Como os recursos dos dispositivos gerenciados são representados através de objetos CIM, a manipulação destes recursos é realizada através de operações abstratas sobre estes objetos. Para tanto, os agentes e os gerentes possuem uma arquitetura específica, que pode ser vista na Figura 2-10 **[59]**.

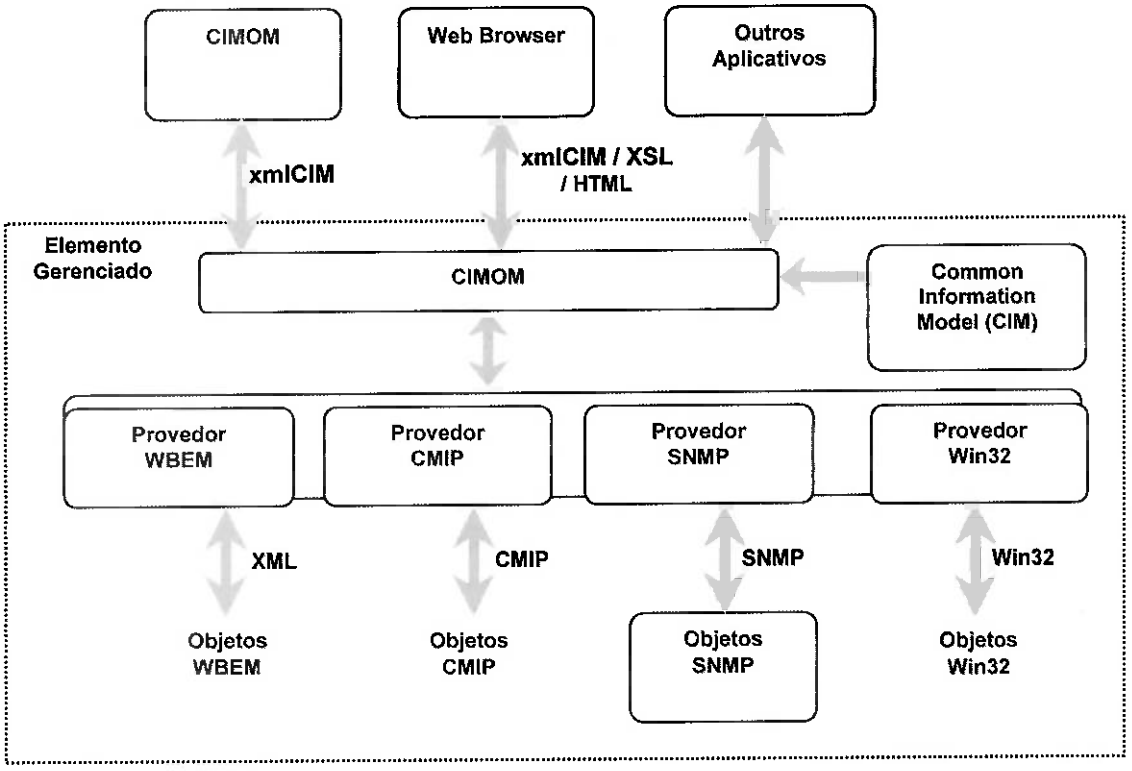


Figura 2-10: Arquitetura WBEM

Nesta arquitetura destacam-se dois subsistemas:

- *CIM Object Manager* (CIMOM): responsável pelo gerenciamento dos objetos no dispositivo gerenciado. Recebe as requisições de operações abstratas, determina quais são os provedores responsáveis pelos objeto e encaminha as operações para execução pelos provedores apropriados;
- Provedores (*Providers*): Responsáveis por executar as operações requisitadas sobre os objetos da CIM. Após receber as requisições de operações abstratas encaminhadas pelo CIMOM, o provedor determina como elas devem ser executadas, mapeando as operações abstratas do modelo em operações reais sobre os recursos gerenciados.

Os objetos de uma CIM não representam, necessariamente, recursos gerenciados locais, podendo representar recursos remotos. Para estes objetos, os provedores atuam como *proxies*: eles recebem as requisições e se comunicam com os elementos gerenciados para a

execução das operações, utilizando as arquiteturas de gerenciamento compatíveis com os elementos gerenciados. Desta maneira, a arquitetura permite **[59]**:

- Integração entre diferentes arquiteturas de gerenciamento: através de provedores específicos para cada arquitetura de gerenciamento. Como exemplo, pode-se determinar um provedor para elementos gerenciáveis via SNMP e outro para elementos gerenciáveis via CMIP;
- Integração entre diferentes aplicativos de gerenciamento: através do WBEM, há um modelo comum para os dados (CIM), bem como para operações sobre os mesmos (xmlCIM/HTTP) e para a codificação e transporte destes dados entre gerentes e agentes. Desta maneira, cada aplicativo deve incluir um CIMOM, de maneira a determinar quais objetos ele pode tratar diretamente e quais objetos devem ser tratados pelo outro aplicativo.

Estas situações são exemplificadas na Figura 2-11, que apresenta uma situação com dois aplicativos de gerenciamento distintos. Um dos aplicativos (denominado *Aplicativo de Gerenciamento WBEM* na figura) atua sobre recursos locais (através de um provedor para objetos locais) e sobre recursos remotos (através de provedores atuando como *proxies* para dispositivos gerenciados via SNMP e CMIP). O outro aplicativo (*Aplicativo de Gerenciamento de Sistemas*) atua sobre recursos diversos, através de um protocolo proprietário; os dois aplicativos são integrados pela troca de informações baseadas no modelo CIM e nas operações xmlCIM/HTTP.

uma página HTML ou página texto simples) ou para outros formatos (por exemplo, uma planilha ou uma base de dados).

A utilização do XML atende às seguintes funções dentro da arquitetura WBEM**[24]** :

- Formato abstrato para a transmissão de objetos CIM **[23]**;
- Declaração de objetos CIM que podem ser facilmente convertidos para representações diferentes (incluindo MOF) **[23]**;
- Encapsulamento de Mensagens CIM para transmissão sobre o HTTP **[22]**.

A arquitetura WBEM define um DTD para o transporte de objetos e mensagens CIM, denominado xmlCIM. Documentos XML carregando objetos CIM devem identificar o DTD xmlCIM, não necessitando incluí-lo junto ao documento (os elementos compatíveis com esta arquitetura já conhecem o DTD) **[24]**.

2.2.3.2 JMX - Java Management Extensions

O padrão JMX (*Java Management eXtensions*) define uma arquitetura para o desenvolvimento de serviços e aplicações de gerenciamento utilizando a linguagem Java. Ele tem por objetivos **[54]**:

- Criação de agentes e gerentes em Java;
- Gerenciamento remoto de aplicativos desenvolvidos em Java;
- Integração de soluções de gerenciamento.

Para atingir tais objetivos são definidos, além de uma arquitetura de gerenciamento **[54]**:

- Diversas APIs (*Application Programming Interfaces*) para o desenvolvimento dos elementos da arquitetura, bem como APIs de integração com outras arquiteturas de gerenciamento;
- Implementações de referência dos elementos da arquitetura JMX, de maneira a facilitar o desenvolvimento de novos elementos;

- Testes padronizados de compatibilidade, possibilitando verificar se novos elementos desenvolvidos estejam atendendo, de forma correta, todos os requisitos da arquitetura.

A arquitetura divide-se em 3 níveis, conforme pode ser visto na Figura 2-12 **[54]**:

- Nível de Instrumentação;
- Nível do Agente;
- Nível de Serviços Distribuídos.

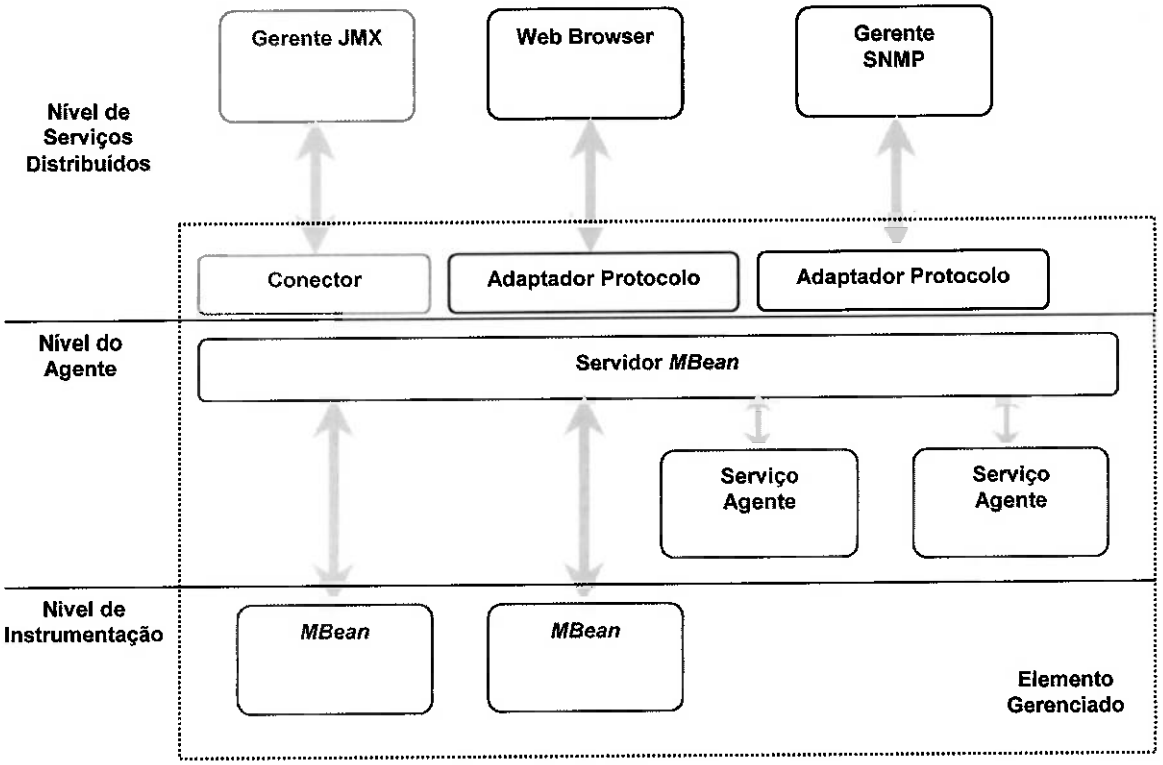


Figura 2-12: Arquitetura JMX

O nível de instrumentação é responsável pela implementação de recursos JMX gerenciáveis **[53]**. Os recursos a serem gerenciados são representados como um ou mais objetos Java denominados *Managed Beans* (ou *Mbeans*), que são desenvolvidos de acordo com um conjunto de regras definidas pela arquitetura. Para um recurso ser gerenciado deve-se, portanto, implementar um conjunto de *MBeans* que atuam como a interface entre o recurso e a

arquitetura JMX, fornecendo ao agente JMX um acesso controlado aos atributos, operações e notificações relativas ao objeto.

O nível do agente é responsável pelo controle dos recursos, permitindo que os aplicativos de gerenciamento remotos possam manipulá-los **[53]**. O agente JMX consiste de um servidor *MBeans,* um conjunto de serviços para manipulação de *MBeans* (serviços do agente), além de pelo menos um módulo de comunicação com os gerentes. Os *MBeans* do nível de instrumentação registram-se no servidor *Mbean* do agente, tornando-se, então, visível para as aplicações de gerenciamento. Estes *MBeans* são manipulados pelos serviços do agente, que podem ser dinamicamente carregados ou removidos do agente conforme a necessidade. Entre as funções do agente definidas pelos serviços, incluem-se a monitoração periódica ou sob demanda de valores de atributos de um *MBean,* a verificação de limites de valores de atributos, entre outros.

Tanto o nível de instrumentação como o do agente exigem que uma máquina virtual Java esteja disponível no dispositivo contendo os recursos gerenciados. Caso não seja possível isto, ou caso o elemento possa ser gerenciado apenas através de uma determinada arquitetura, pode-se definir um mecanismo de *proxy* na arquitetura. Define-se, então, um conjunto de *MBeans* que fornece ao agente as interfaces exigidas pela arquitetura JMX, mapeando-as para os elementos específicos da arquitetura compatível com o dispositivo.

Por fim, a comunicação entre gerentes e agentes é feita pelo nível de serviços distribuídos **[54] [53]**. São definidos dois mecanismos de comunicação:

- Conectores: fornecem ao gerente uma API para comunicação entre gerente e agente JMX independente de protocolo, denominada interface de comunicação. Para se utilizar um novo protocolo entre o gerente e o agente (por exemplo, para utilizar SSL, ao invés de TCP), bastaria trocar os respectivos conectores, não exigindo outras alterações no gerente e agente. Sendo assim, conectores diferentes provêm a mesma interface de comunicação através de protocolos diferentes, permitindo o acesso transparente ao agente sem preocupação com o protocolo utilizado;

- Adaptadores de Protocolo: mapeiam os elementos de um agente JMX e seus *MBeans* para um outro protocolo ou arquitetura de gerenciamento (por exemplo, para SNMP ou HTML).

2.3. Arquitetura do ATM - Forum

O ATM Forum especifica um modelo de referência para o gerenciamento fim-a-fim de redes ATM **[6][45]**. Neste modelo, são descritos os diversos requisitos para o gerenciamento de dispositivos ATM, bem como de redes privativas e públicas baseadas nesta tecnologia. Em particular, são definidos à luz do modelo de referência os requisitos para um serviço de gerenciamento entre rede pública e rede privativa, denominado gerenciamento CNM (*Customer Network Management* **[6]**).

2.3.1 Modelo de Referência para gerenciamento de redes ATM

O modelo de referência para o gerenciamento de redes ATM pode ser visto na Figura 2-13 **[6][45]**:

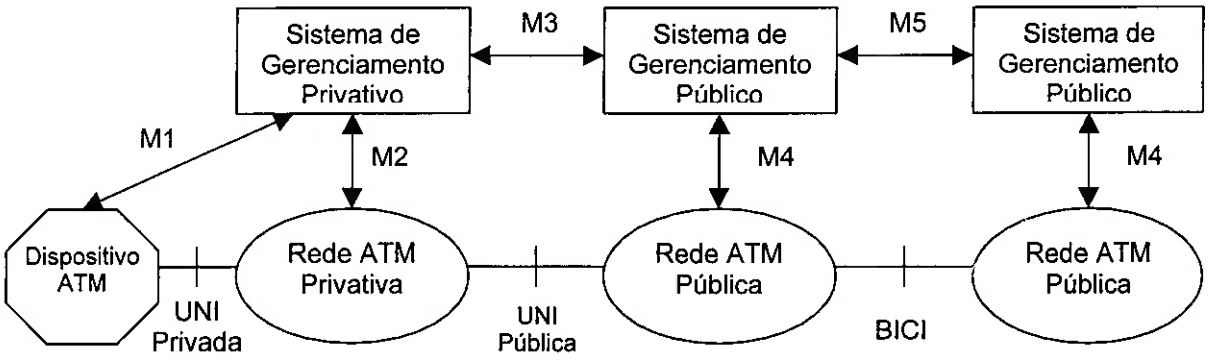


Figura 2-13: Modelo de Gerenciamento do ATM Forum

De acordo com o modelo, são necessárias 5 interfaces diferentes para o gerenciamento de redes ATM, denominadas interfaces M1 a M5. Cada interface tem as funcionalidades definidas de acordo com a parte da rede ATM a ser gerenciada.

2.3.1.1 Interfaces M1 e M2

As interfaces M1 e M2 são utilizadas para o gerenciamento de uma rede ATM privativa: enquanto a interface M1 é responsável pelo gerenciamento dos dispositivos ATM terminais (estações de trabalho e servidores, por exemplo), a interface M2 encarrega-se do gerenciamento da infra-estrutura de rede (por exemplo, *switches*).

Tais interfaces são definidas baseadas na arquitetura de gerenciamento Internet, uma vez que esta arquitetura é a mais difundida para o gerenciamento de redes. Sendo assim, o protocolo utilizado para a comunicação entre o sistema de gerenciamento e os elementos da rede privativa deve ser o SNMP. Além disto, foram definidas MIBs relevantes para o gerenciamento dos enlaces físicos (SONET/SDH, DS-1, DS-3), da camada ATM e das camadas superiores.

2.3.1.2 Interfaces M3

A interface M3 é denominada também de interface CNM (*Customer Network Management*). Ela descreve a interface entre o sistema de gerenciamento privativo de uma instituição usuária de uma rede ATM pública (*customer*) e o sistema de gerenciamento da própria rede pública, permitindo ao usuário um gerenciamento limitado da rede pública, correspondente a parte por ele utilizada. Tal gerenciamento pode ser tanto passivo (o usuário supervisiona a utilização do *backbone*) como ativo (permitindo ao usuário um certo controle sobre o *backbone*).

2.3.1.3 Interfaces M4

A interface M4 é responsável pelo gerenciamento dos elementos de uma rede ATM pública. Diferentemente das redes privativas (onde normalmente se utiliza a arquitetura SNMP), nas redes públicas, a arquitetura de gerenciamento normalmente utilizada é o TMN (*Telecommunications Management Network*), que adota como protocolo de gerenciamento o CMIP (*Common Management Information Protocol*). No entanto, na interface M4 é utilizada uma abordagem que permite diferentes arquiteturas de gerenciamento.

Na interface M4, são especificadas MIBs lógicas, que definem os requisitos de gerenciamento para as redes públicas independentemente da arquitetura e protocolos de gerenciamento a serem utilizados. Estas MIBs lógicas devem ser mapeadas para as MIBs especificas de cada arquitetura de gerenciamento, facilitando assim a utilização de arquiteturas diferentes dentro de uma mesma rede, bem como permitindo a extensão da interface para novas arquiteturas de gerenciamento.

2.3.1.4 Interfaces M5

A interface M5 é considerada a interface mais complexa do modelo, uma vez que envolve sistemas de gerenciamento de diferentes redes públicas. Ainda não há padrões definidos para esta interface.

2.3.2 Customer Network Management (CNM)

Os provedores de serviços de conectividade em geral estabelecem com os seus clientes acordos relacionados à confiabilidade e disponibilidade dos serviços prestados. Tais acordos, denominados SLAs (*Service Level Agreements),* formalizam, através de um conjunto de indicadores e métricas, as características do serviço a ser oferecido aos clientes. No entanto, os clientes devem ser capazes de monitorar estes indicadores, exigindo que o provedor ofereça mecanismos que permitam sua monitoração.

O serviço CNM é uma resposta à necessidade dos provedores de oferecer mecanismos para a verificação dos SLAs estabelecidos. O conceito principal do gerenciamento CNM (*Customer Network Management*) é de que um provedor de serviços de conectividade possa compartilhar informações e funções de gerenciamento relativas a estes serviços com as instituições que deles fazem uso (mais especificamente com os administradores das redes destas instituições). Sendo assim, o gerenciamento CNM é um serviço adicional a ser oferecido pelos provedores a seus clientes, capacitando-os a manterem uma visão mais completa das interligações entre seus *sites* através dos *backbones* dos provedores.

Uma arquitetura típica para um serviço de gerenciamento CNM pode ser vista na Figura 2-14. No provedor há um sistema de gerenciamento, responsável pela monitoração e controle de sua rede. Dada a diversidade de serviços que uma dada instituição (um dado cliente do provedor) pode utilizar e a variedade de sistemas desta instituição, uma interface padronizada é necessária para a troca de informações de gerenciamento entre o provedor e a instituição através do serviço de gerenciamento CNM. Para tanto, um elemento, denominado um **agente CNM**, é responsável pela comunicação padronizada entre os sistemas de gerenciamento dos clientes e o sistema de gerenciamento do provedor.

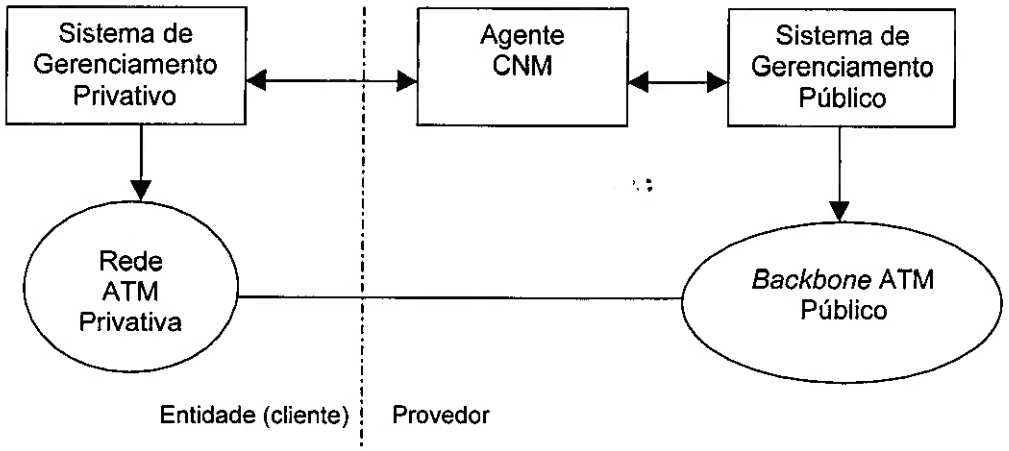


Figura 2-14: Arquitetura Típica de uma solução de gerenciamento CNM.

Há algumas diferenças entre um gerenciamento normal e um gerenciamento CNM, derivadas do compartilhamento da rede do provedor entre diversos clientes. Como a rede do provedor é compartilhada por diversos clientes, estes não podem ter acesso a todas as informações e funções de um gerenciamento tradicional. Enquanto que o provedor é responsável pelo gerenciamento da rede como um todo, cada cliente "possui" uma parte da rede, sendo que o serviço de CNM deve ser relativo a esta parte.

No gerenciamento da rede, o provedor tem uma visão física da rede, verificando enlaces e elementos de interconexão (por exemplo, *switches* e roteadores) ao longo de seu *backbone*. Os clientes, por sua vez, tem acesso a uma visão lógica desta mesma rede, composta basicamente por seus pontos de acesso ao *backbone* e as conexões lógicas entre estes

pontos. É responsabilidade do agente CNM prover cada cliente com a sua respectiva visão lógica da rede (Figura 2-15).

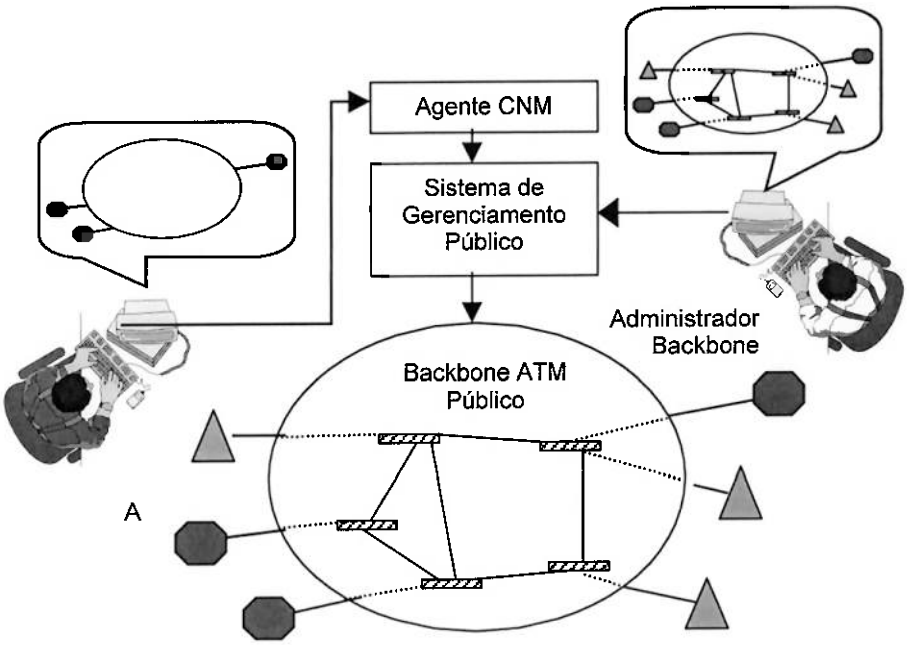


Figura 2-15: Diferença entre visão física da rede e visão lógica fornecida pelo agente CNM

Há uma diferença, também, em relação ao modo de execução das operações. Em um gerenciamento tradicional, as operações de gerenciamento são executadas pelo sistema de gerenciamento quando requisitadas pelo administrador da rede. Em um gerenciamento CNM, as operações ficam pendentes, uma vez que elas são executadas sobre esta visão lógica da rede. Tais operações devem passar pela aprovação do administrador da rede física, que determina se esta operação é possível de ser executada e se ela não implica em prejuízos para os outros clientes da rede (dependendo da operação, o próprio agente CNM pode autorizar a operação, notificando o administrador da rede física de sua execução).

Além destas diferenças, o serviço CNM não necessita apresentar as informações em tempo real, uma vez que, dependendo do número de clientes acessando o serviço, isto pode acarretar uma sobrecarga de operações de gerenciamento sobre a rede do *backbone* e prejuízos do seu

desempenho global. O serviço CNM deve garantir que estas informações estejam dentro de um intervalo de tempo fixo, de maneira que os usuários do serviço saibam o limite máximo de tempo em que as informações podem estar desatualizadas.

2.3.3 Requisitos do ATM Forum para o serviço CNM

O ATM Forum, através da especificação da interface M3, define duas classes de funções para um serviço CNM:

- Classe I: o serviço CNM deve prover informações de monitoração da parte específica da rede pública utilizada por uma dada instituição. Devem ser fornecidas informações relativas ao gerenciamento de configuração, falhas e desempenho (as áreas de contabilização e segurança não são cobertas pela especificação da interface M3);
- Classe II: o serviço CNM deve prover, além das funções da Classe I, mecanismos para a adição, modificação e remoção de circuitos virtuais.

Os requisitos definidos pela especificação da interface M3 dividem-se em requisitos gerais, a serem atendidos independentemente da classe de serviço oferecida, e requisitos específicos para cada uma das classes de serviço.

2.3.3.1 Requisitos Gerais

A especificação da interface M3 define como requisitos obrigatórios para um serviço CNM em uma rede ATM:

- O suporte a todas as funções especificadas para a Classe I de serviço;
- O protocolo a ser utilizado para a comunicação com o agente CNM deve ser o SNMP;
- As MIBs a serem utilizadas devem ser baseadas nos padrões do IETF;
- O agente deve suportar, obrigatoriamente, a MIB específica para o mecanismo de transmissão utilizado (por exemplo, MIB SONET/SDH ou DS3 caso sejam utilizadas interfaces SONET/SDH ou DS3);

- As informações a serem fornecidas ao usuário do serviço devem ser relativas somente à parte da rede pública utilizada pela instituição do usuário do serviço;
- O serviço M3 deve suportar o gerenciamento de PVCs (*Permanent Virtual Circuits*) ponto-a-ponto e interfaces UNI (*User-Network Interface*);
- Caso a rede suporte serviços de PVCs multiponto, o serviço M3 deve suportar o seu gerenciamento;
- O agente CNM deve manter registros de todas as requisições SNMP recebidas;
- O agente deve suportar as informações relativas à extensão do grupo de interfaces da MIB-II.

Além destes, um requisito importante está relacionado ao tempo de resposta das operações. Para atender às requisições de gerenciamento em um serviço CNM, o agente deve executar uma série de operações, tais como operações de segurança e de interação com a plataforma de gerenciamento, que impõem um certo atraso na resposta. O serviço de gerenciamento CNM deve garantir um limite superior para este atraso, de maneira que os usuários do serviço não recebam informações demasiadamente desatualizadas. São definidos dois parâmetros de funcionamento do agente:

- T1: limite máximo para o intervalo entre uma requisição de monitoração de uma dada variável e a sua respectiva resposta;
- T2: limite máximo para o intervalo entre a ocorrência de um evento e o envio de um alarme, notificando o usuário da rede.

Tais parâmetros T1 e T2 são pertinentes ao funcionamento do serviço de gerenciamento CNM e o requisito a ser atendido pelo agente é que T1 e T2 sejam suficientemente pequenos para que o serviço CNM seja de utilidade para os seus usuários. De acordo com o padrão M3, o provedor do serviço deve especificar os valores de T1 e T2 do serviço CNM por ele oferecido.

Além dos requisitos obrigatórios, a especificação define, também, alguns requisitos opcionais para um serviço de gerenciamento CNM:

- Suporte às funções da Classe II;
- A versão do protocolo de comunicação pode ser tanto o SNMPv1 como SNMPv2;
- O agente pode manter um *log* de todos os *SetRequests* recebidos;
- O agente pode fornecer um mecanismo específico para a visualização dos *logs* mantidos pelo agente.

2.3.3.2 Requisitos Classe I

De acordo com a especificação M3, o serviço CNM Classe I deve fornecer, obrigatoriamente, informações relativas a:

- Desempenho e configuração da camada ATM e da camada física;
- *Status* e configuração dos VPLs (*Virtual Path Links*), VCLs (*Virtual Channel Links*), VPCs (*Virtual Path Circuits*) e VCCs (*Virtual Channel Circuits*), quando suportados pela rede pública;
- Descritores de tráfego;
- Alarmes de *linkUp* e *linkDown* associados ao *status* da UNI da instituição.

Além disso, o agente CNM Classe I deve permitir somente acessos via SNMP do tipo *read-only* (*Get-Requests* e *GetNext-Requests*).

2.3.3.3 Requisitos Classe II

Os requisitos de um serviço M3 Classe II são, em sua maioria, requisitos opcionais. A especificação determina que, para prover um serviço Classe II, o agente deve suportar, obrigatoriamente, os serviços Classe I e pelo menos as funcionalidades de um dos seguintes grupos:

- *Class II ATM Level Subgroup*: o serviço deve permitir a alteração de configurações no nível ATM;
- *Class II VPC/VCC Subgroup*: o serviço deve permitir a modificação de configurações e *status* de VPLs, VCLs, VPCs e VCCs;
- *Class II Traffic Subgroup*: o serviço deve permitir a modificação de descritores de tráfego ATM em VPCs e VCCs.

2.4. Considerações Finais

O principal modelo para o gerenciamento de redes ATM, especificado pelo ATM Forum, define a utilização da arquitetura SNMP em suas interfaces de gerenciamento M1, M2 e M3. Como há um grande esforço em andamento para a definição de uma arquitetura de gerenciamento *web*, capaz de oferecer benefícios adicionais à arquitetura SNMP, é interessante verificar as suas principais características, quando consideradas para o gerenciamento de redes ATM. A Tabela 2-14 apresenta uma rápida visão comparativa entre as arquiteturas SNMP e de gerenciamento via *web*.

	SNMP	**Gerenciamento via web**
Protocolo de Transporte	UDP, não-orientado à conexão.	HTTP/TCP, orientado à conexão.
Autenticação	Básica, por nome de comunidade.	Pode-se definir diversos mecanismos de autenticação, inclusive por certificados.
Criptografia do tráfego de gerenciamento	Somente com SNMPv3.	Utilizando-se SSL, obtém-se um canal criptografado HTTPS.
Controle de Acesso	Dependente da implementação do agente SNMP.	Servidores apresentam listas de controle de acesso.
Hierarquia entre gerentes	Centralizado (SNMPv1) ou hierárquico (SNMPv2).	Distribuído.
Suporte em equipamentos de rede	Alto (SNMPv1), baixo (SNMPv2).	Baixo.
Acesso a serviços de gerenciamento	Plataforma de gerenciamento ou SNMP MIB *browser*.	*Web browser*.
Demanda por recursos computacionais pelo agente	Baixa.	Alta.

Legenda: HTTP *HyperText Transfer Protocol*
IP *Internet Protocol*
MIB *Management Information Base*
SNMP *Simple Network Management Protocol*
SSL *Secure Sockets Layer*
TCP *Transmission Control Protocol*
UDP *User Datagram Protocol*

Tabela 2-14: Aspectos comparativos das arquiteturas SNMP e de gerenciamento via *web*

As principais vantagens do gerenciamento SNMP sobre o gerenciamento via *web* são o seu grande suporte pelos dispositivos a serem gerenciados e a pouca demanda por recursos computacionais requerida pelos agentes. O gerenciamento via *web*, por sua vez, leva vantagem em relação à segurança que pode ser agregada às soluções de gerenciamento, bem como na facilidade de acesso, através de um *web browser*.

No próximo capítulo, são apresentados alguns padrões definidos para a modelagem de informações de gerenciamento em redes ATM.

3. Padrões para informações de gerenciamento ATM

Para o gerenciamento de redes ATM (*Asynchronous Transfer Mode*), é necessário a modelagem de suas informações em MIBs (*Management Information Bases*), de maneira que os gerentes possam se comunicar de maneira inequívoca com os elementos da rede. A maioria das MIBs definidas para o gerenciamento de redes ATM são especificadas de acordo com a arquitetura SNMP (*Simple Network Management Protocol*), em particular com a SMIv2 (*Structure of Management Information – version 2*). Além destas, algumas MIBs definidas pelo ATM Forum são especificadas para a arquitetura de gerenciamento de redes do modelo OSI (*Open System Interconnection*), baseada no protocolo CMIP (*Common Management Information Protocol*). Não há, ainda, nenhuma padronização para a modelagem de informações de gerenciamento de redes ATM baseadas em arquiteturas de gerenciamento *web*.

Os padrões para informações de gerenciamento SNMP de redes ATM são definidos por duas entidades distintas: o ATM Forum e o IETF (*Internet Engineering Task Force*). Os grupos de trabalho do IETF são responsáveis pela definição da arquitetura SNMP e de diversas MIBs para esta arquitetura. Já o ATM Forum define as MIBs para especificações por eles produzidas, como, por, exemplo, o padrão de *LAN Emulation* (LANE) e o ILMI (*Integrated Local Management Protocol*).

As MIBs definidas pelo IETF e pelo ATM Forum são complementares: as MIBs SNMP definidas pelo ATM Forum utilizam definições especificadas em MIBs do IETF.

3.1. Padrões IETF

Há, no IETF, um grupo de trabalho responsável pela especificação de padrões para gerenciamento de redes ATM, denominado *AToM Working Group* (o "o" da sigla vem de SONET, uma vez que este grupo é, também, responsável pela especificação de elementos para o gerenciamento de enlaces SONET). Nenhuma arquitetura adicional de gerenciamento foi especificada para o gerenciamento de redes ATM; os padrões foram definidos de acordo

com a arquitetura SNMP e consistem na especificação de diversas MIBs e elementos para a definição de novas MIBs para ATM (tais como definições de tipos de dados e macros em ASN.1 – *Abstract Syntax Notation One*).

A principal MIB é conhecida como AToM MIB. Ela possui as definições de objetos para o gerenciamento básico da infra-estrutura ATM. Além desta, são definidas, também, MIBs para o gerenciamento da camada física (tais como enlaces SDH/SONET e DS3/E3) e MIBs para serviços ATM, tais como contabilização (*accounting*), *multicast* e IP *over* ATM. Para auxiliar na especificação de novas MIBs ATM, foram especificadas também convenções textuais para tipos de dados e macros relacionadas à tecnologia ATM; tais especificações estão contidas na RFC 2514 [43].

Os padrões especificados pelo IETF, assim como o *status* de padronização de cada um, são mostrados na Tabela 3-1 [34]:

MIBs	**RFC**	**Data Publicação**	***Status***	**Nome da subárvore na hierarquia de nomeação**
AToM MIB v.1 [2]	1695	Agosto/1994	Obsoleto	*atmMIB*
AToM MIB v.2 [56]	2515	Fevereiro/1999	Proposto	*atmMIB*
IPoA MIB [30]	2320	Abril/1998	Proposto	*ipoaMIB*
DS3/E3 MIB [28]	2496	Janeiro/1999	Proposto	*ds3*
SDH/SONET MIB [55]	2558	Março/1999	Proposto	*sonetMIB*
Multicast over UNI 3.0/3.1 [21]	2417	Setembro/1998	Proposto	*MarsMIB*
Accounting MIB [38]	2512	Fevereiro/1999	Proposto	*AtmAccountingInformationMIB* e *accountingControlMIB*
Textual Conventions for ATM MIBs [43]	2514	Fevereiro/1999	Proposto	——

Legenda:
ATM *Asynchronous Transfer Mode*
IPoA *Classical IP over ATM*
MIB *Management Information Base*
SDH *Synchronous Digital Hierarchy*
SONET *Synchronous Optical Network*
UDP *User Datagram Protocol*
UNI *User-Network Interface*

Tabela 3-1: *Status* de Padronização das MIBs ATM do IETF

Há uma relação de dependência entre estas MIBs. Um agente responsável pelo gerenciamento de um dispositivo de rede ou um serviço ATM deve, obrigatoriamente, suportar as seguintes MIBs:

- MIB-II (RFC 1213): objetos básicos para gerenciamento SNMP;
- AToM MIB v.2 (RFC 2514 e 2514): objetos básicos para gerenciamento de redes ATM. Pode ser necessário que o agente suporte a AToM MIB v.1 (RFC 1695), ao invés da v.2;
- IF-MIB (RFC 1573): extensão do grupo de interfaces da MIB-II, contém objetos necessários para o gerenciamento das interfaces ATM. Pode ser necessário que o agente suporte objetos da RFC 2233, que contém a atualização da RFC 1573.

Dentro da hierarquia de nomeação da arquitetura SNMP, as MIBs ATM do IETF encontram-se normalmente sob *iso.org.dod.internet.mgmt.mib-2* (1.3.6.1.2.1), podendo encontrar-se também sob *iso.org.dod.internet.mgmt.mib-2.transmission* (1.3.6.1.2.1.10). A Figura 3-1 apresenta a posição em que se encontram as MIBs exibidas na Tabela 3-2.

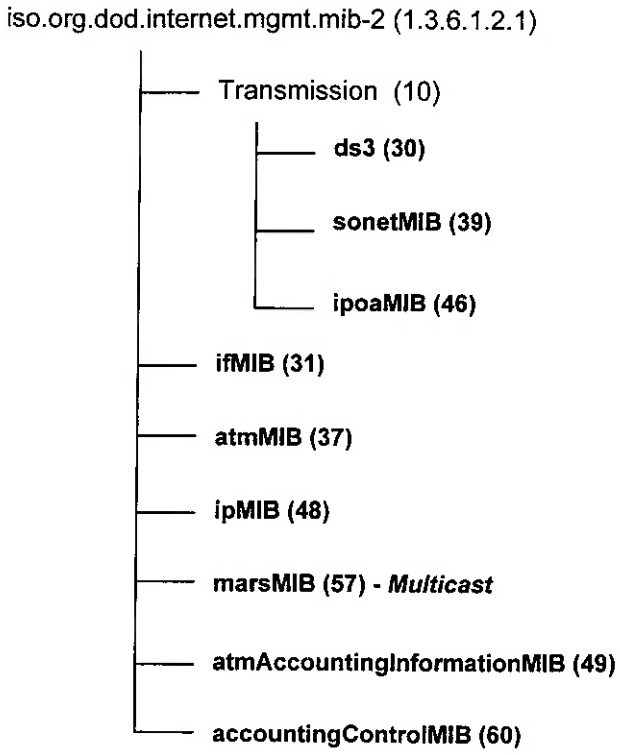


Figura 3-1: Hierarquia de Nomeação para as MIBs ATM do IETF

3.2. Padrões ATM Forum

Dentro do ATM Forum há diversos grupos de trabalho (*Technical Working Group*) responsáveis por padrões ATM. O grupo de trabalho de gerenciamento (*Network Management Working Group*) é responsável pela especificação da arquitetura de gerenciamento de redes ATM e dos requisitos das interfaces M1-M5 da arquitetura. As outras MIBs são padronizadas por grupos de trabalhos responsáveis pelos próprios serviços ATM a serem gerenciados (Tabela 3-2).

MIB	Documento	Data Publicação	Grupo de Trabalho
LANE client MIB [10]	af-lane-0038.000	Setembro/1995	*Lan Emulation* / MPOA
LANE client MIB Addendum	af-lane-0050.000	Dezembro/1995	*Lan Emulation* / MPOA
LANE server MIB [11]	af-lane-0057.000	Março/1996	*Lan Emulation* / MPOA
LANE client MIB v.2	af-lane-0093.000	Outubro/1998	*Lan Emulation* / MPOA
PNNI MIB [8]	af-pnni-0055.000	Março/1996	P-NNI
Soft PVC MIB [9]	af-pnni-0066.000	Setembro/1996	P-NNI
PNNI MIB (revisão)	af-pnni-0081.000	Maio/1997	P-NNI
ILMI MIB [7]	af-ilmi-0065.000	Setembro/1996	ILMI
ATM RMON [12]	af-nm-test-0080.000	Julho/1997	*Network Management*

Legenda:
ATM *Asynchronous Transfer Mode*
ILMI *Integrated Local Management Interface*
LANE *LAN Emulation*
MIB *Management Information Base*
MPOA *MultiProtocol over ATM*
PNNI *Private Network-Network Interface*
PVC *Permanent Virtual Circuit*
RMON *Remonte MONitoring*

Tabela 3-2: MIBs definidas pelo ATM Forum

As MIBs apresentadas na Tabela 3-2 são definidas dentro da arquitetura SNMP e utilizam a hierarquia de nomeação padronizada pelo IETF. Estas MIBs utilizam definições das MIBs ATM padronizadas pelo IETF; sendo assim um agente SNMP que suporte uma das MIBs padronizadas pelo ATM Forum deve obrigatoriamente suportar a AToM MIB [2], MIB-II [49] e IF-MIB [39] .

Dentro da hierarquia de nomeação do IETF, as MIBs padronizadas pelo ATM Forum encontram-se sob *iso.org.dod.internet.private.enterprises.atmforum* (1.3.6.1.4.1.353).

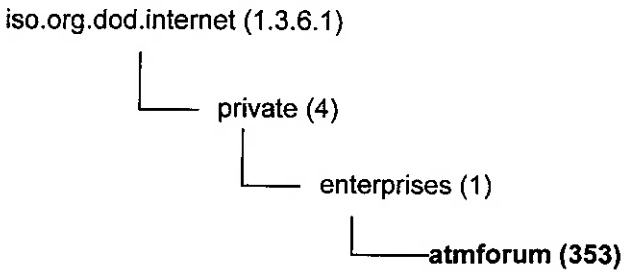


Figura 3-2: Hierarquia de Nomeação para as MIBs do ATM Forum

3.3. Principais MIBs

As diversas MIBs definidas pelo IETF e pelo ATM Forum podem ser agrupadas da seguinte maneira:

- Nível Físico: relacionadas à camada física das redes ATM, tais como: as MIBs para gerenciamento de enlaces DS-3 e SONET/SDH (respectivamente DS3 MIB e SONET MIB);
- Nível ATM: relacionadas ao próprio funcionamento da rede ATM, tais como a AToM MIB e, indiretamente, a MIB-II e IF-MIB;
- Nível de Serviços: relacionadas aos serviços que fazem uso da rede ATM, como por exemplo LAN *Emulation* ou *Classical* IP *over* ATM.

São vistas a seguir a MIB AToM, base para todas as MIBs de gerenciamento ATM, bem como as MIBs para LAN *Emulation* ou *Classical* IP *over* ATM (respectivamente definidas pelo ATM Forum e pelo IETF). As MIBs relacionadas a serviços ATM foram selecionadas por apresentarem serviços que podem ser utilizados tanto em redes privativas (tais como uma rede de um campus universitário) como em redes públicas, onde são oferecidos serviços de conectividade Ethernet ou IP, utilizando-se uma infra-estrutura ATM.

3.3.1 AToM MIB

A principal MIB para o gerenciamento de redes ATM é a AToM MIB. Definida inicialmente na RFC 1695 [2], inclui objetos para o gerenciamento básico do funcionamento de elementos de redes ATM, permitindo a monitoração de elementos de comutação, circuitos virtuais, parâmetros de QoS e camada de adaptação AAL5.

Desde a publicação da RFC 1695, houve algumas mudanças nos padrões ATM, forçando a atualização da especificação da AToM MIB (em particular, com a especificação ATM Forum *Traffic Management Specification* 4.0 [4], as classes de serviço dos circuitos virtuais foram alteradas). As macros e os tipos de dados definidos na RFC 1695 foram especificados em um

documento separado (RFC 2514 [43]), de maneira a facilitar o seu uso por outras MIBs ATM, e a nova versão da AToM MIB foi proposta através da RFC 2515 (tal versão é denominada de AToM MIB v.2) [56].

Os objetos da AToM MIB podem ser classificados em 4 categorias distintas, de acordo com a sua funcionalidade:

- Gerenciamento da camada física: os objetos desta categoria são utilizados como complemento às MIBs da camada física, quando são utilizados enlaces SONET ou DS3;
- Gerenciamento de Interfaces: os objetos desta categoria são utilizados para o gerenciamento de configuração e desempenho das interfaces físicas dos dispositivos ATM gerenciados como um todo, ao invés de se considerar cada circuito virtual;
- Gerenciamento de Circuitos Virtuais: os objetos desta categoria são utilizados para o gerenciamento de configuração de VCs (*Virtual Channels*) ou VPs (*Virtual Paths*), incluindo parâmetros de tráfego associados aos circuitos virtuais, tais como parâmetros de QoS (*Quality of Service*);
- Gerenciamento da AAL5: os objetos deste grupo permitem o gerenciamento da entidade AAL5 (*ATM Adaptation Layer 5*) presente em alguns dispositivos ATM (normalmente dispositivos terminais).

Além destes objetos, a especificação da AToM MIB define como devem ser tratados os objetos da MIB-II e da extensão IF MIB dentro do contexto de redes ATM.

3.3.1.1 Gerenciamento da Camada Física

O gerenciamento da camada física em redes ATM deve ser feito através das MIBs correlatas, tais como a SONET MIB ou a DS3 MIB. No entanto, a AToM MIB oferece objetos que auxiliam no gerenciamento da camada física de enlaces SONET e DS3. São definidos dois grupos, o *TC Sublayer Group* (para o gerenciamento da funcionalidade de mapeamento direto de células em

quadros de camadas físicas do tipo PDH ou SDH) e o *DS3 PLCP Group* (para o gerenciamento da funcionalidade de mapeamento de células em enlaces DS3, quando é utilizado o protocolo PLCP – *Physical Layer Convergence Procedure*).

3.3.1.2 Gerenciamento das Interfaces

No gerenciamento da camada de células ATM, o gerenciamento é feito por interface, agregando-se as estatísticas dos diversos circuitos que por ela passam. Para tanto, é definida na AToM MIB uma tabela, denominada *ATM Interface Configuration Table*, que deve ser usada em conjunto com os objetos da IF MIB.

Dentro da MIB-II e IF MIB, as interfaces ATM necessárias para o gerenciamento da camada de células são representadas através do tipo *atm* (valor 37). Os objetos da IF MIB e do grupo de interface da MIB-II contêm as estatísticas de desempenho e erro da camada ATM. Alguns objetos destas MIBs recebem uma interpretação especial dentro do contexto de redes ATM, tais como:

- *ifInOctets/ifOutOctets*: número de bytes recebidos e enviados na interface respectivamente; dividindo o valor destes objetos por 53 obtém-se o número de células recebidas e enviadas. Para interfaces com altas taxas de transmissão, estes valores podem ser armazenados nos objetos *ifHCInOctets/ifHCOutOctets* (contadores de 64 bits definidos na IF MIB), uma vez que com os contadores de 32 bits definidos na MIB-II o valor máximo poderia ser atingido rapidamente;
- *ifInErrors*: número de células descartadas por erros, que não puderam ser corrigidas através do valor do campo HEC (*Header Error Control*) das células;
- *ifInUnknownProtos*: número de células inválidas recebidas e descartadas, tais como células com valores inválidos de VPI/VCI (*Virtual Path Identifier/Virtual Channel Identifier*) ou outros valores inválidos nos campos do cabeçalho.

Na tabela *ATM Interface Configuration,* podem ser monitorados os parâmetros de configuração da camada ATM, tais como:

- Informações sobre os elementos vizinhos ao dispositivo, tais como: identificação da interface e endereço IP do elemento diretamente conectado à interface (normalmente o endereço IP utilizado para o gerenciamento SNMP);
- Informações sobre a configuração dos VPCs (*Virtual Path Circuits*) e VCCs (*Virtual Channel Circuits*) associados à interface, tais como: número máximo de VPCs e VCCs, número de VPCs e VCCs em uso, número máximo de *bits* de VPIs/VCIs suportados pelo *hardware* da interface, número de *bits* de VPI/VCI negociados com o equipamento vizinho, entre outros;
- Identificadores (VPI/VCI) do circuito virtual utilizado pelo ILMI (*Integrated Local Management Interface*) na interface.

3.3.1.3 Gerenciamento de Circuitos Virtuais

Há dois níveis de circuitos ou conexões virtuais em redes ATM: vias virtuais (VPs) e canais virtuais (VCs), que são tratados separadamente pelos *switches*. São definidos 5 grupos para o gerenciamento dos circuitos virtuais, sendo que para o gerenciamento de um dado VC ou VP são utilizados três grupos:

- *Virtual Path Link Group e Virtual Channel Link Group*: este grupos são utilizados para a definição, respectivamente, das características de um VPL (*Virtual Path Link*) ou de um VCL (*Virtual Channel Link*). Entre as informações contidas nestes grupos têm-se o identificador do enlace virtual (VPI, para VPLs ou VPI/VCI para VCLs), o *status* do enlace virtual, tipo do enlace virtual (se é ponto-a-ponto ou ponto-multiponto, por exemplo) e ponteiros para os descritores de tráfego do enlace virtual, que são definidos em um grupo à parte dentro da MIB. Como o tráfego nos VCs e VPs é bi-direcional, sendo possível definir características de tráfego distintas em cada sentido, são definidos dois ponteiros para os descritores de tráfego em cada VC ou VP;

- *Virtual Path Cross-Connect Group e Virtual Channel Cross-Connect Group*: estes grupos contêm a tabela de comutação de VPs e VCs (respectivamente). Estas tabelas indicam o mapeamento entre interface e VPI de entrada para interface e VPI de saída (para comutação de VPs) e o mapeamento entre interface e VPI/VCI de entrada para interface e VPI/VCI de saída;
- *Traffic Descriptor Parameter Group*: este grupo contém uma tabela para a descrição dos padrões de tráfego a serem gerenciados no dispositivo. Os padrões definidos nesta tabela não estão associados a uma interface, podendo ser utilizados por VPLs e VCLs diferentes.

Os descritores de tráfego são especificados de acordo com as definições de conformidade das especificações UNI 3.0/3.1 [5] e *Traffic Management* 4.0 [4]. Como os diferentes tipos de tráfego são caracterizados por parâmetros diferentes, a MIB define 5 objetos genéricos (*atmTrafficDescrParam1* a *atmTrafficDescrParam5*), sendo o seu significado dependente do tipo de tráfego. Por exemplo, um tráfego do tipo CBR (*Constant Bit Rate*) pode ser caracterizado pelos parâmetros PCR (*Peak Cell Rate*) e CDVT (*Cell Delay Variation Tolerance*); a AToM MIB define que estes parâmetros sejam armazenados, respectivamente, em *atmTrafficDescrParam1* e *atmTrafficDescrParam2* quando se descreve um trafégo CBR para o gerenciamento.

3.3.1.4 Gerenciamento da AAL5

O gerenciamento da camada de adaptação AAL5 é similar ao gerenciamento da camada de células. Ele é feito por um grupo de objetos da AToM MIB, denominado *AAL5 CPCS (Common Part Convergence Sublayer) Performance Group* e pelo grupo *interfaces* da MIB-II.

O grupo *AAL5 CPCS Performance* contém objetos para a monitoração de informações de desempenho específicas da AAL5, tais como PDUs (*Protocol Data Units*) descartadas por erro de CRC (*Cyclic Redundancy Check*) ou PDUs parcialmente remontadas que foram descartadas por *time-out* na remontagem, entre outros.

A AToM MIB especifica que a entidade AAL5 seja representada, para fins de gerenciamento, através de uma interface interna, de tipo *aal5* (valor 49). Sendo assim, os parâmetros de configuração e estatísticas de desempenho podem ser obtidos através do grupo interfaces da MIB-II; seus objetos recebem uma interpretação específica:

- *IfMtu*: indica o tamanho máximo da PDU que pode ser processada pela entidade AAL5;
- *IfInOctets/ifOutOctets*: indicam, respectivamente, o número de bytes das PDUs AAL5 recebidas e enviadas;
- *IfInUcastPkts/ifOutUcastPkts*: indicam, respectivamente, o número de PDUs AAL5 enviadas para uma entidade de camada superior para processamento e recebidas de uma entidade de camada superior para transmissão na rede. O objeto *ifOutUcastPkts* não indica o número de PDUs enviadas na rede, pois devem ser descontadas as PDUs com erros;
- *IfInErrors/ifOutErrors*: indicam, respectivamente, o número de PDUs com erros recebidas da rede e recebidas de uma entidade de camada superior para transmissão. Entre causas de erros, pode-se citar erros de CRC-32, *time-out* de remontagem e SDUs de tamanho excessivo;
- *IfInDiscards/ifOutDiscards*: indicam, respectivamente, número de PDUs recebidas da rede e de uma entidade de camada superior, respectivamente, e descartadas por problemas como, por exemplo, *buffer* de entrada ou saída cheios;
- *IfInMulticastPkts, ifInBroadcastPkts, ifOutMulticastPkts, ifOutBroadcastPkts, IfSpeed, ifInUnknownProtos*: devem conter o valor 0, uma vez que estes objetos não são usados na interface interna da AAL5;
- *IfPhysAddress*: deve conter uma *string* de comprimento 0, uma vez que este objeto não é usado na interface interna da AAL5.

3.3.1.5 ATM Supplemental MIB

A utilização da MIB AToM para o gerenciamento de conexões virtuais pode ser visto em detalhes em [57]. No entanto, esta MIB não atende todos requisitos necessários para o gerenciamento de circuitos virtuais comutados (SVCs), como demonstra o estudo em [50]. Tais deficiências são endereçadas por uma MIB em desenvolvimento [42], denominada *ATM Supplemental MIB*, que define objetos adicionais para gerenciamento tanto de SVCs (*Switched Virtual Circuits*) como de PVCs (*Permanent Virtual Circuits*). As principais funcionalidades adicionais fornecidas por esta MIB em sua última especificação são:

- Tabelas para a monitoração de SVCs: através das tabelas relativas a circuitos virtuais na MIB AToM, o administrador da rede pode criar, configurar e remover, manualmente, PVCs. Ao contrário dos PVCs, os SVCs são criados e removidos automaticamente pelas aplicações que os utilizam, não sendo possível ao gerente alterar a configuração dos SVCs. Sendo assim, os SVCs só podem ser monitorados remotamente pelo gerente, devendo ficar em tabelas separadas fornecidas por esta MIB que permitem apenas operações do tipo *get;*

- Monitoração e configuração de protocolos de sinalização: esta MIB fornece objetos que permitem a monitoração das estatísticas das entidades de sinalização. Além disso, são definidos objetos para a configuração da sinalização nas interfaces do equipamento (por exemplo, se a interface estiver configurada para utilização da sinalização UNI 3.1 (*User-Network Interface*), deve-se configurar o papel da interface na sinalização – usuário ou rede);

- Configuração do serviço ILMI: o serviço ILMI é utilizado pelo *switch* para informar a dispositivos ATM sobre os prefixos de rede a serem utilizados no endereçamento, bem como sobre a localização de serviços da rede ATM (por exemplo, servidores de *LAN Emulation*, servidor ATMARP, entre outros). Esta MIB permite que estas informações sejam configurada nos *switches* via SNMP;

- *Traps* para indicar a queda de um PVC: estas *traps* visam indicar, rapidamente, ao gerente problemas de conectividade, evitando que esses problemas sejam detectados apenas através de *polling* nos objetos de *status* de cada PVC definido no *switch* ATM. Como pode haver uma longa lista de PVCs definidos, estas *traps* visam diminuir a sobrecarga de processamento nos elementos da rede e o consumo desnecessário de banda da rede pelo protocolo de gerenciamento.

3.3.2 IPoA

A MIB IPoA, definida pela RFC 2320 [30] de acordo com as normas da SMIv2, é utilizada para o gerenciamento do serviço de *Classical IP over ATM* (CLIP). Faz parte do grupo *transmission* (Figura 3-1).

Definido pela RFC 2225 [31][37], o serviço CLIP permite que estações e servidores comuniquem-se entre si através de endereçamento IP apenas, não necessitando utilizar endereços ATM. Em uma rede ATM com CLIP, as sub-redes IPs são lógicas e são denominadas LIS (*Logical IP Subnetwork*). A resolução de endereços em uma LIS é feita através de um servidor ATMARP, que faz o mapeamento de endereços IPs em endereços ATM e vice-versa. Para pertencer e se comunicar dentro de uma LIS, um elemento deve ser um cliente CLIP para interagir com o servidor ATMARP.

A MIB IPoA é composta por três partes:

- Básica: contém objetos que devem ser implementados tanto pelos clientes como pelos servidores de uma LIS;
- Cliente: contém objetos que devem ser implementados somente em clientes de uma LIS;
- Servidor: contém objetos que devem ser implementados somente em servidores ATMARP.

Como o serviço de CLIP pode utilizar tanto SVCs como PVCs, a MIB IPoA inclui objetos específicos para o gerenciamento de conexões dos dois tipos de circuitos, comutados e permanentes.

De acordo com o padrão, esta MIB deve ser usada em conjunto com outras MIBs. Sendo assim, um agente que a implemente deve suportar também as seguintes MIBs (Figura 3-1):

- RFC 2011: Extensões para o grupo *IP* da MIB-II (ipMIB). Contém os parâmetros relativos à camada IP em um elemento de rede que utiliza o serviço CLIP;
- RFC 2233: Extensões do grupo *Interfaces* da MIB-II (IF MIB). Diversos objetos da MIB IPoA fazem referência a uma interface de rede ATM. O objeto *ifType* do grupo *Interfaces* em clientes CLIP recebe o valor *ipOverAtm* ou 114;
- RFC 1695: AToM MIB v.1 (atmMIB). Contém os parâmetros dos circuitos virtuais utilizados no serviço CLIP.

As principais funções da MIB IPoA são:

- Gerenciamento de Configuração:
  - Criação e Remoção de uma LIS (sub-rede lógica IP);
  - Criação e Remoção de servidores ATMARP e clientes de uma LIS;
  - Manutenção da tabela de resolução de endereços ATM em IP e vice-versa;
  - Identificação dos VCCs utilizados no serviço IPoA;
- Gerenciamento de Desempenho:
  - Estatísticas do servidor ATMARP e dos clientes de uma LIS;
- Gerenciamento de Falhas:
  - Estatísticas das operações do serviço com falhas;
  - Notificações (alarmes) de eventos significativos.

A MIB não cobre as áreas funcionais relativas ao gerenciamento de segurança e contabilização.

3.3.2.1 Definições Básicas

Todos os servidores ATMARP e clientes CLIP devem implementar as seguintes tabelas:

- *ATM Logical IP Subnet (LIS) Table*: contém todas as LIS das quais o dispositivo gerenciado é membro. Indexada pelo endereço IP da sub-rede associada a cada LIS, contém os parâmetros de utilização desta LIS, tais como a MTU (*Maximum Transfer Unit*), o tipo de encapsulamento (se o quadro AAL5 deve ou não incluir um cabeçalho do padrão 802.2 LLC – *Logical Link Control*), temporizadores, entre outros;
- *ATM Logical IP Subnet Interface Mapping Table*: esta tabela mapeia as LIS nas interfaces ATM que devem suportá-las;
- *ATMARP Remote Server Table*: esta tabela contém uma lista dos servidores ATMARP a serem utilizados pelo dispositivo gerenciado. É indexada conjuntamente pelo endereço IP da sub-rede associada a uma LIS, o endereço ATM do servidor ATMARP desta LIS e interface de rede do dispositivo utilizada na comunicação com o servidor. Além de indicar o *status* operacional do servidor ATMARP, pode-se definir se este servidor deve ou não ser utilizado pelo dispositivo gerenciado;
- *ATM VC Table*: esta tabela contém uma lista de todos os VCs utilizados pelo dispositivo para a comunicação entre os outros membros das LIS;
- *ATM Config PVC Table*: esta tabela visa a criação e remoção dos PVCs utilizados na comunicação entre membros de uma LIS.

Além destas tabelas, todos os clientes e servidores devem suportar a emissão de uma notificação denominada *ipoaMtuExceeded*, que indica quando um quadro recebido possui uma MTU maior do que a negociada.

3.3.2.2 Clientes CLIP

A parte específica de clientes CLIP da MIB IPoA é composta por uma tabela, denominada *ATMARP Client Table*. Cada linha da tabela corresponde a um cliente CLIP (um dado equipamento pode estar conectado a diversas LIS), sendo ela indexada pelo endereço IP do cliente. Esta tabela contém parâmetros de configuração, desempenho e falhas:

- Configuração: mantém os valores do endereço ATM do cliente e do servidor ATMARP utilizado pelo cliente;
- Desempenho: mantém as estatísticas das requisições e respostas ATMARP e InATMARP recebidas e enviadas pelo cliente. São mantidas também as estatísticas de respostas ATMARP negativas (ATMARP_NAK) recebidas pelo cliente;
- Falhas: mantém as estatísticas das requisições e respostas ATMARP e InATMARP recebidas com problemas, tais como: pacotes inválidos, respostas não recebidas e operações inválidas. Além disso, um cliente pode erroneamente receber requisições ATM ARP (quando outro cliente está configurado com o seu endereço como sendo o do servidor ATMARP) e, neste caso, ele pode enviar ATMARP_NAK como resposta, sendo que o agente deve manter as estatísticas dos ATMARP_NAK enviados.

3.3.2.3 Servidor CLIP

A parte específica de servidor ATMARP da MIB IPoA é composta por uma tabela, denominada *ATMARP Server Table* e pelas notificações a serem implementadas pelo servidor.

A *ATMARP Server Table* é composta por informações relacionadas aos servidores ATMARP associados ao agente, onde cada linha da tabela corresponde a um servidor ATMARP. Ela é indexada por dois objetos, o endereço ATM e o endereço IP do servidor ATMARP e contém os seguintes parâmetros:

- Configuração: mantém os valores do endereço ATM do servidor e o endereço IP da LIS a qual o servidor pertence;

- Desempenho: mantém as estatísticas das requisições e respostas ATMARP e InATMARP recebidas e enviadas pelo servidor. São mantidas, também, as estatísticas de respostas ATMARP negativas (ATMARP_NAK) enviadas pelo servidor;
- Falhas: mantém as estatísticas das requisições e respostas ATMARP e InATMARP recebidas com problemas, tais como: pacotes inválidos e operações inválidas. Além disso, mantém estatísticas de clientes com endereços IP duplicados em cada LIS.

Além da configuração e monitoração, a tabela permite a criação e remoção de servidores ATMARP.

São especificadas 3 notificações que o agente deve suportar. As duas primeiras, *ipoaLisCreate* e *ipoaLisDelete*, são utilizadas para avisar o gerente que uma LIS foi criada ou removida (respectivamente). A terceira notificação, *ipoaDuplicateIpAddress*, indica ao gerente que elementos diferentes tentaram associar diferentes endereços ATM ao mesmo endereço IP.

3.3.3 LANE MIB

O serviço de *LAN Emulation*, ou LANE, visa fornecer aos protocolos da camada de rede serviços da camada de enlace de uma rede local (LAN) do tipo Ethernet ou Token Ring.

De acordo com o ATM Forum, são necessárias quatro MIBs para o gerenciamento de uma rede ATM com LANE 1.0 [10][11]:

- LEClient MIB: define os objetos para o gerenciamento dos clientes de LANE (LEC - *Lan Emulation Client)*;
- ELAN MIB: define os objetos para o gerenciamento de um servidor LECS (*LAN Emulation Configuration Server*) e configuração de ELANs (*Emulated LANs)*;
- LES MIB: define os objetos para o gerenciamento de um servidor LES (*LAN Emulation Server*);
- BUS MIB: define os objetos para o gerenciamento de um servidor BUS *(Broadcast and Unknown Server)*.

Para o gerenciamento do serviço de LAN Emulation deve-se utilizar as MIBs relativas aos servidores de LANE: ELAN MIB, LES MIB e BUS MIB. Elas são definidas em [11] como três MIBs independentes, uma vez que nem sempre os três servidores são implementados no mesmo dispositivo. Como exemplo, o servidor LECS pode estar em um dispositivo separado dos servidores LES e BUS ou mesmo não existir dentro da rede (pode-se utilizar a configuração estática ao invés de se utilizar o LECS). Além disso, as MIBs LANE foram especificadas de maneira que um agente seja capaz de gerenciar diversos servidores LANE, dentre os quais mais de um servidor LES, BUS e/ou LECS.

A Figura 3-3 identifica a posição das MIBs LANE na hierarquia de nomeação.

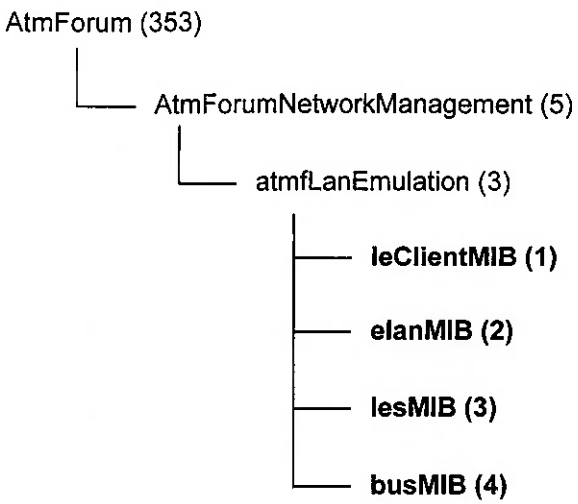


Figura 3-3: Hierarquia de Nomeação para as MIBs de LANE

As principais funções das MIBs dos servidores de LANE:

- Gerenciamento de Configuração:
  - Criação e Remoção de ELANs;
  - Atribuição de clientes a uma ELAN;
  - Monitoração da topologia de uma ELAN (quais LECs estão associados a cada LES);
  - Identificação dos VCCs utilizados pelo LES, BUS e LECS;
- Gerenciamento de Desempenho:
  - Estatísticas de cada servidor da ELAN;
  - Estatísticas de cada par servidor de ELAN e LEC (BUS-LEC, LES-LEC);
- Gerenciamento de Falhas:
  - *Status* operacional de cada componente de LANE;
  - *Logs* de falhas para cada componente de LANE.

A MIB não cobre as áreas funcionais relativas ao gerenciamento de segurança e contabilização.

O agente para o gerenciamento dos servidores de LANE deve, também, suportar a AToM MIB. Em particular, os VCCs utilizados são apenas identificados nas MIBs LANE; os seus parâmetros devem ser obtidos através da AToM MIB. Além disso, o agente deve suportar a MIB-II, pois apesar dos servidores não possuírem uma interface ATM, eles podem receber e enviar tráfego através de uma ou mais interfaces ATM (por exemplo, em um *switch*). As MIBs dos servidores definem quais são estas interfaces, sendo os seus parâmetros obtidos através da MIB-II.

3.3.3.1 LANE Client MIB

Um dispositivo ATM pode pertencer a mais de uma LANE; para tanto ele deve possuir um LEC para cada rede emulada à qual ele pertence. Cada LEC é, normalmente, implementado no dispositivo através de uma interface de rede virtual e, sendo uma interface, há uma linha na tabela de interfaces da MIB-II correspondendo a cada LEC (cada LEC corresponde, também, a uma entrada na tabela de extensões do grupo de extensão definida pela RFC 1573). O tipo de interface LEC é definido pelo tipo de rede emulada, podendo o objeto ifType da MIB-II receber um dos seguintes valores:

- aflane8023 (59): o LEC emula um interface Ethernet/802.3;
- aflane8025 (60): o LEC emula um interface Token Ring/802.5.

A *LANE client* MIB possui diversas tabelas, a maioria de implementação obrigatória. As principais tabelas são:

- *LEC Configuration Table*: responsável pela criação, remoção e configuração dos LECs em um dispositivo ATM, contendo parâmetros do tipo *read-write*. Nesta tabela, encontram-se os objetos para a configuração inicial do cliente LANE, tais como: o nome e o tipo da ELAN à qual o cliente pertence, indicação se ele deve usar um

servidor LECS, o endereço ATM do LES (caso ele não use nenhum LECS), entre outros;

- *LEC Status Table*: Contém objetos (com permissão de leitura apenas) que definem o *status* operacional de cada LEC. Além do *status* da interface LEC e de códigos de erro, contém os endereços dos servidores LECS e LES sendo utilizados;
- *LEC Statistics Table*: As colunas desta tabela indicam as estatísticas de funcionamento do LEC, tais como: quantos LE_ARP *requests* e *replies* foram recebidos e transmitidos por cliente, o número de quadros de controle transmitidos e recebidos e as estatísticas de falha no estabelecimento de SVCs;
- *LEC Server Connections Table*: identifica os VPIs, VCIs e interfaces de rede utilizadas nos VCCs estabelecidos entre o LEC e os servidores de LANE, a saber: *Configuration direct VCC, Control direct VCC, Control distribute VCC, Multicast Send VCC* e *Multicast Forward VCC*;
- *LE ARP Cache*: Contém a tabela ARP para resolução entre endereços MAC e endereços ATM. Contém, também, objetos que definem como a entrada na tabela ARP foi criada (manual ou automaticamente), permitindo a adição e remoção de novas entradas manualmente.

Além destas tabelas, a LANE client MIB possui algumas tabelas de implementação condicional. Tais tabelas referem-se a particularidades do cliente LANE 802.5 onde, em adição à informação do endereço ATM, há informações sobre rotas entre anéis Token Ring que devem ser mantidas e utilizadas.

3.3.3.2 ELAN MIB

A ELAN MIB é dividida em duas partes, sendo uma delas relativa à configuração do servidor LECS e outra referente à configuração das ELANs de responsabilidade do servidor LECS.

Na porção relativa à ELAN, há dois grupos principais:

- *ELAN Administration Group*: define quais são as políticas implementadas pelo LECS que são utilizadas para decidir a qual ELAN um dado LEC deve se associar. Como exemplo, o LECS pode definir a ELAN de um dado LEC baseado no endereço ATM, MAC, tipo da LAN emulada, nome da ELAN definida pelo cliente, entre outros;
- *ELAN Configuration Group*: contém os parâmetros de configuração para cada ELAN.

A porção da MIB referente à configuração do servidor LECS contém diversos grupos. Como a especificação da MIB foi definida de maneira que um dado agente possa gerenciar diversos servidores LECS, a MIB define os diversos parâmetros de cada LECS gerenciado pelo agente. As principais tabelas são:

- *LECS Configuration Table*: responsável pela criação, remoção e configuração dos servidores LECSs em um dispositivo ATM, contendo parâmetros do tipo *read-write*. Nesta tabela, encontram-se os objetos referentes ao *status* do servidor LECS e outros relativos à sua configuração inicial, tais como: as políticas utilizadas pelo LECS para a atribuição da ELAN a um dado cliente, as interfaces (e endereços ATM) em que o LECS estará recebendo pedidos de informações dos clientes, entre outros;
- *LECS to ELAN Mapping Table*: indica quais são as ELANs que estão sob responsabilidade de cada servidor LECS;
- *LECS Error Log Control Table*: indica os parâmetros de operação do mecanismo de *log*, tais como: o número máximo de entradas na tabela de *log*, qual a última entrada do *log*, entre outros. Através desta tabela, pode-se, também, verificar o *status* do mecanismo de *log*, habilitá-lo ou desabilitá-lo, bem como remover as entradas da tabela de *log*, reinicializando-a;
- *LECS Log Table*: contém o *log* de falhas do servidor LECS, indicando, para cada falha, data/hora da falha, a sua natureza e o endereço ATM do cliente que causou a falha;
- *LECS Statistics Table*: registra as estatísticas de funcionamento do servidor LECS, tais como número de requisições de configuração recebidas corretamente dos clientes bem

como as que foram descartadas, número de requisições com erros de endereçamento ATM, entre outros.

3.3.3.3 LES MIB

A LES MIB contém os principais objetos relativos à operação do serviço LANE: através da LES MIB pode-se saber quais são os clientes ativos e os servidores (LES e BUS) em cada uma das ELANs, bem como obter a tabela de mapeamento entre endereços MAC e endereços ATM. É composta pelos seguintes grupos:

- *LES Configuration Table*: contém a configuração de todos os LES gerenciados pelo agente, indicando o endereço ATM de cada LES, qual ELAN cada um deles atende, além de objetos referentes à monitoração do *status* e ao controle da operação de cada LES;
- *LES VCC Table*: contém uma lista de todos os *Control Distribute VCCs*, mantidos entre o LES e os respectivos clientes para a resolução de endereços MAC para ATM;
- *BUS Table*: contém uma lista com o endereço ATM dos servidores BUS responsáveis pelas ELANs; tal informação é passada pelo LES ao cliente na fase de inicialização;
- *LES MAC ARP Table*: contém o mapeamento de endereços MAC (*unicast* ou *broadcast*) para endereços ATM efetuado pelo LES; no caso de endereços MAC de *broadcast,* o mapeamento é feito para o endereço do BUS. Os endereços da tabela podem ter natureza estática (quando configurados diretamente pelo gerente nesta tabela) ou dinâmica (quando a informação é proveniente do registro de um LEC em uma ELAN, durante a sua inicialização);
- *LES-LEC Table*: contém as informações relativas aos clientes associados a cada ELAN, mapeando um LES e seus respectivos LECs. Para cada cliente, contém informações relativas ao seu *status* bem como à sua configuração, tais como: o *Control Direct* VCC (VCC utilizado para a resolução ATM-MAC), o seu endereço ATM, e a

indicação se o cliente está operando em modo normal (uma estação conectada diretamente à rede ATM, com um endereço MAC único) ou *proxy* (quando diversos endereços MAC são mapeados para um único endereço ATM, como, por exemplo em um equipamento de rede com uma porta ATM e várias portas Ethernet que executa *bridging* entre as duas redes);

- *LES Statistics Table*: mantém as estatísticas de desempenho e falhas do LES. Entre as estatísticas de falhas, encontram-se contadores para identificar erros em endereços ATM e MAC, LEC IDs, pacotes com campos inválidos e outras falhas no registro de um cliente em uma ELAN. Em relação ao desempenho, há objetos que indicam as estatísticas das operações de resolução de endereços (operações LE_ARP) bem como de registro de novos clientes junto ao LES;
- *LES-LEC Statistics Table*: contém, para cada par LES-LEC, as estatísticas de desempenho das operações LE_ARP. Mantém estatísticas mais detalhadas em relação à tabela anterior (que mantinha apenas estatísticas globais da operação) como, por exemplo, estatísticas da resolução de endereços *unicast* ou *broadcast*;
- *LES Log Control Table e LES Log Table*: funcionam de maneira análoga às tabelas *LECS Error Log Control Table* e *LECS Log Table*, sendo o controle do funcionamento do mecanismo de *log* executado pela primeira tabela e os registros de *log* mantidos na segunda tabela.

3.3.3.4 BUS MIB

A BUS MIB contém os seguintes grupos:

- *BUS Configuration Table*: contém a configuração de todos os BUS gerenciados pelo agente, indicando o endereço ATM de cada BUS, qual ELAN cada um deles atende, além de objetos responsáveis pela monitoração do *status* e pelo controle da operação de cada BUS;

- *BUS VCC Table*: contém uma lista de todos os *Multicast Forward VCCs*, que são utilizados pelo BUS para enviar o tráfego de *broadcast* e *multicast* para os participantes da ELAN;
- *BUS-LEC Table*: de maneira análoga à LES-LEC Table da LES MIB, contém as informações de mapeamento entre cada BUS e seus respectivos LECs. Para cada cliente, contém informações relativas à sua configuração, tais como o *Multicast Send* VCC e o seu endereço ATM;
- *BUS Statistics Table*: mantém as estatísticas gerais de desempenho e falhas do BUS. Entre as estatísticas de falhas, encontram-se contadores para identificar falhas no estabelecimento dos VCCs *multicast send* e *multicast forward* bem como quadros descartados por falta de recursos ou *timeout*. Em relação ao desempenho, há objetos que indicam as estatísticas das operações do BUS, tais como: quadros *unicast*, *multicast* ou de controle recebidos;
- *BUS-LEC Statistics Table*: contém, para cada par BUS-LEC, estatísticas de desempenho das operações do BUS. Entre elas, encontram-se estatísticas sobre os quadros recebidos e encaminhados para cada cliente, tanto para quadros *unicast* e *multicast* como para quadros onde o cliente não possui o endereço ATM do destinatário;
- *BUS Log Control Table e BUS Log Table*: funcionam de maneira análoga às tabelas de *log* na LES MIB e na LECS MIB.

3.4. Considerações Finais

Conforme visto neste capítulo, os padrões de MIBs para o gerenciamento de redes ATM utilizam a arquitetura SNMP. Além disso, diversas informações necessárias ao gerenciamento estão contidas em MIBs utilizadas dentro do contexto de redes ATM, porém que não são definidas, exclusivamente, para esta tecnologia, tais como a IF MIB ou a ipMIB.

Tais características exigem que uma solução para gerenciamento de redes ATM seja capaz de interagir com equipamentos de rede e dispositivos ATM gerenciáveis via SNMP, uma vez que dificilmente eles devem ser compatíveis com outras arquiteturas de gerenciamento. Além disso, estas soluções devem buscar as informações em diversas MIBs e consolidá-las para se obter uma visão da situação da rede, adicionando uma certa complexidade para a solução de gerenciamento e/ou para os administradores responsáveis pelo gerenciamento.

O próximo capítulo apresenta os requisitos que devem ser atendidos por uma solução de gerenciamento CNM para redes ATM, baseado nos estudos das arquiteturas e das MIBs relacionadas a esta tecnologia.

4. Requisitos da arquitetura de uma solução de Gerenciamento CNM

Conforme visto anteriormente, o serviço de gerenciamento CNM (*Customer Network Management*) é provido por uma ferramenta de gerenciamento, denominada agente CNM, que atua entre o sistema de gerenciamento do *backbone* ATM (*Asynchronous Transfer Mode*) e os sistemas de gerenciamento das instituições que fazem uso deste *backbone*. A arquitetura da solução de gerenciamento CNM deve incluir, além do próprio agente CNM, a especificação da interface de comunicação entre o gerente e o agente CNM, conforme pode ser visto na Figura 4-1.

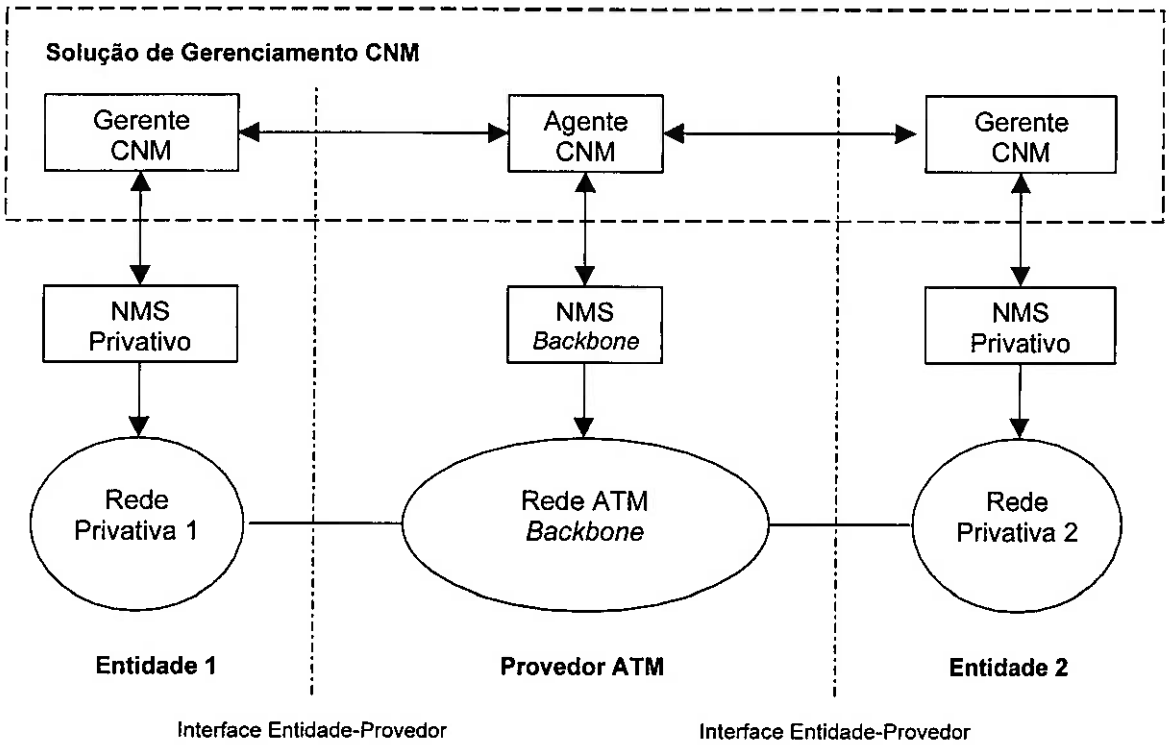


Figura 4-1: Contexto de uma solução de gerenciamento CNM

Para redes ATM, o serviço CNM é definido de acordo com a especificação da interface de gerenciamento M3 do modelo do ATM Forum. Esta especificação é bastante completa e, além dos requisitos funcionais, são definidos:

- Métodos de transporte para tráfego M3 (tais como: a utilização da própria rede ATM, linhas privativas ou mesmo conexões discadas), discussão de suas características e fatores necessários para a seleção de um ou mais destes métodos de transporte;
- Objetos das MIBs AToM e IF MIB necessários para atender cada requisito. Através da especificação M3 e das MIBs citadas, o gerente sabe quais objetos deve utilizar para atender um determinado requisito e como estes objetos devem ser manipulados;
- Classes de serviços CNM, de maneira que os serviços CNM possam ser implementados e/ou oferecidos de forma gradual, de acordo com a necessidade do provedor de conectividade;
- Arquitetura *proxy* padronizada, especificando que todo o serviço deve ser fornecido por um agente CNM.

A especificação da interface M3 é também mais completa que a especificação equivalente para redes *Frame Relay*, que apresenta apenas os requisitos funcionais do serviço, deixando em aberto as especificações da arquitetura. Sendo mais completa, reduz-se a probabilidade de existir problemas de interoperabilidade entre implementações desenvolvidas a partir dessa especificação para redes ATM.

No entanto, a especificação M3 apresenta alguns inconvenientes:

- A interface M3 foi especificada a partir da AToM MIB (*Management Information Base*) v.1 (RFC 1695), não tendo sido atualizada desde 1994. Não leva em consideração, portanto, atualizações nos padrões ATM (por exemplo, as classes de serviço ATM especificadas pela norma *Traffic Management 4.0*) ocorridos desde então;
- A interface M3 deve ser utilizada somente para o gerenciamento de PVCs (*Permanent Virtual Connections*), uma vez que, para o gerenciamento de circuitos virtuais do tipo SVC (*Switched Virtual Connections*) e SPVC (*Semi-Permanent Virtual Connections*), devem ser utilizados objetos de outras MIBs em adição aos objetos da AToM MIB;

- A especificação do protocolo SNMP (*Simple Network Management Protocol*) para a comunicação com o agente CNM exige que as instituições usuárias do *backbone* ATM possuam uma ferramenta para gerenciamento de redes ATM, dada a complexidade de se gerenciar esta tecnologia diretamente através dos valores dos objetos das MIBs ATM. Isto requer dos administradores destas instituições conhecimento maior sobre a tecnologia, além de recursos computacionais maiores para a plataforma de gerenciamento, restringindo o gerenciamento às instituições capazes de dispor de recursos financeiros para investir na formação destes profissionais e nas ferramentas necessárias ao seu suporte;
- O protocolo SNMP não oferece a segurança necessária a um gerenciamento CNM, em termos de autenticação, confidencialidade, integridade, controle de acesso e não-repudiação.

Um dos principais objetivos deste trabalho é especificar a arquitetura de uma solução de gerenciamento CNM para redes ATM, capaz de oferecer um serviço classe I, de acordo com a especificação M3 e evitar alguns dos inconvenientes descritos acima.

Este capítulo apresenta os requisitos da solução de gerenciamento CNM para redes ATM; a especificação destes requisitos foi baseada não só na especificação M3, como também na experiência prática do autor no gerenciamento de uma rede ATM onde foi detectada a necessidade de uma ferramenta de gerenciamento CNM.

Os requisitos da solução de gerenciamento CNM para redes ATM apresentados neste capítulo encontram-se divididos em três partes:

- Requisitos Básicos do Serviço CNM: descreve os requisitos que o serviço CNM deve atender, em função do uso previsto;
- Requisitos Funcionais do serviço CNM: descreve as funcionalidades que o serviço CNM deve prover a seus usuários;

- Requisitos da Arquitetura da solução de gerenciamento CNM: descreve como deve ser composta, internamente, a arquitetura genérica de uma solução de gerenciamento CNM, visando atender os requisitos básicos e os requisitos funcionais do serviço.

4.1. Requisitos Básicos para um serviço CNM

Como o agente CNM é acessado por diferentes usuários, possivelmente com níveis distintos de conhecimento em relação à tecnologia ATM, é necessário que o agente seja capaz de prover serviços a estes diferentes usuários de maneira distinta. Para isto é importante que o agente CNM atenda aos seguintes requisitos:

- Particionamento das Informações: As informações relativas à rede devem ser particionadas, de maneira que os usuários somente podem visualizar uma parte da rede, relacionada aos serviços por ele utilizados;
- Definição de Níveis Funcionais: o agente deve ser capaz de oferecer diferentes níveis funcionais de serviço, atendendo usuários com diferentes requisitos (por exemplo, usuários básicos e avançados) e/ou privilégios (por exemplo, usuários técnicos e não-técnicos);
- Interface Customizada: baseado no perfil do usuário, a interface deve fornecer acesso somente a informações sobre equipamentos e a funções permitidas, de acordo com a sua instituição e o seu nível de conhecimento (por exemplo, interface distinta para usuários básicos e avançados);
- Controle de Operações: o agente deve controlar as operações requisitadas pelos usuários do serviço, de maneira a evitar que um usuário de uma dada instituição execute uma operação de gerenciamento que possa impactar os serviços utilizados por outras instituições;

- Facilidade de Uso: O agente deve possuir uma interface gráfica que permita a usuários sem grandes conhecimentos de ATM obter algumas estatísticas básicas da utilização da rede.

4.2. Requisitos Funcionais

O modelo OSI define 5 áreas funcionais de gerenciamento, normalmente referenciadas pela sigla FCAPS (*Fault, Configuration, Accounting, Performance and Security,* respectivamente Falhas, Configuração, Contabilidade, Desempenho e Segurança). Através destas áreas funcionais, é possível classificar as funcionalidades suportadas por uma solução de gerenciamento.

O agente CNM definido neste trabalho deve cobrir as áreas funcionais relativas a gerenciamento de falhas, desempenho e configuração. As áreas funcionais de segurança e contabilização não são atendidas pelo agente CNM especificado neste trabalho, uma vez que os padrões para redes ATM relativos a estas áreas são relativamente recentes e nem sempre há equipamentos de rede que os implemente.

4.2.1 Requisitos Gerais

O agente CNM deve suportar algumas funcionalidades básicas de gerenciamento.

- *Log* de Transações: todas as operações executadas no agente devem ser armazenadas em um *log*, de maneira a permitir o *troubleshooting* do sistema;
- Definição de alarmes: é interessante que os usuários do serviço CNM possam definir limites para algumas variáveis a serem monitoradas. Com isto o agente pode, no caso destes limites serem atingidos, emitir uma notificação a estes usuários. Preferencialmente, o tipo da notificação a ser emitida deve ser configurável, atendendo às necessidades dos usuários do serviço CNM (por exemplo, o responsável por um dado *switch* pode ser notificado através de um *e-mail* ou uma *trap* SNMP). O agente CNM deve suportar os seguintes tipos básicos de alarmes:

- Alarme de Comunicação: disparado quando a variável ultrapassou um dos limites estabelecidos que indica um problema de indisponibilidade e/ou conectividade, impactando a utilização da rede ATM;
- Alarme de Qualidade de Serviço: disparado quando a variável monitorada ultrapassou um dos limites estabelecidos, indicando uma degradação da qualidade de serviço associada à variável monitorada;
- Alarme de Processamento: disparado quando há um problema interno de processamento no sistema de gerenciamento CNM (por exemplo, uma falha de software do agente), impedindo a verificação do alarme configurado;

- Relatórios e gráficos de dados históricos: o agente deve manter armazenados alguns dados coletados ao longo de sua execução, de maneira a fornecer relatórios e gráficos de acompanhamento aos usuários do serviço e, com isso, permitir a monitoração do funcionamento da rede ao longo do tempo;
- Estatísticas em tempo real: os usuários do serviço devem ser capazes de receber informações coletadas em tempo real. O agente deve ser capaz de coletar estatísticas e, ao mesmo tempo, controlar esta coleta, de maneira que ela não impacte os serviços de gerenciamento e a própria operação da rede;
- Informações de configuração: através do agente deve ser possível visualizar informações relacionadas à configuração de equipamentos ATM, de acordo com as restrições de segurança definidas pelo nível funcional de serviço associado ao usuário.

4.2.2 Gerenciamento de Falhas

No gerenciamento CNM, é importante que o usuário possa determinar a ocorrência de problemas com o serviço contratado. Sendo assim, o agente deve fornecer informações capazes de auxiliar o administrador na detecção e correção de problemas de rede. Para tanto ele deve suportar as seguintes funcionalidades:

- Monitoração de taxas de erro: o agente deve monitorar as MIBs relativas às camadas física e ATM, observando objetos da MIB relativos a erros de transmissão e recepção. É importante observar as taxas de erro para procurar detectar situações de erro, tais como: problemas no meio físico, tráfego excedendo a capacidade de comutação dos *switches*, entre outros;
- Alarmes para taxas de erros: o agente deve permitir a definição de limites para as taxas de erro, emitindo um alarme caso estes limites sejam atingidos;
- Histórico de *Downtime*: através do agente deve ser possível observar quando um determinado elemento crítico da rede ficou indisponível, bem como determinar por quanto tempo este serviço ficou indisponível. Como exemplos de elementos críticos, podem ser considerados tanto equipamentos de rede (roteadores e *switches*) como serviços (por exemplo, serviços de *LAN Emulation*, *Classical IP over ATM* e *MultiProtocol over ATM*);
- Testes de Conectividade: através do agente deve ser possível executar de testes de conectividade, verificando se um dado elemento pode ser acessado pelo agente e obtendo algumas estatísticas deste acesso, tais como: tempo médio e taxa de erro.

4.2.3 Gerenciamento de Configuração

O gerenciamento CNM deve permitir a visualização de detalhes das configurações utilizadas pela infra-estrutura ATM, tais como:

- Circuitos Virtuais: o agente deve ser capaz de exibir uma tabela com os circuitos virtuais (PVCs, SVCs e SPVCs) e seus parâmetros. Em particular, devem ser exibidos os parâmetros de QoS (*Quality of Service*) negociados e/ou reais;
- Configuração dos *Switches*: o agente deve exibir os dados referentes ao *switch* ATM, tais como: informações gerais e informações sobre as suas interfaces. Para as interfaces, deve ser exibidas, pelo menos, a taxa de transmissão (objetos *ifSpeed* da

MIB-II em interfaces do tipo *atm*). Opcionalmente, o agente pode fornecer informações relacionadas a limitações do *hardware* ATM utilizado, tais como: o número máximo de VPCs/VCCs (*Virtual Path Connection/Virtual Channel Connection*) na interface (objetos *atmInterfaceMaxVpcs* e *atmInterfaceMaxVccs* da AToM MIB) e o número máximo de *bits* de VPI/VCI (*Virtual Path Identifier/Virtual Channel Identifier*) que podem ser utilizados na interface (objetos *atmInterfaceMaxActiveVpiBits* e *atmInterfaceMaxActiveVciBits* da AToM MIB); tais informações podem ser úteis para se determinar problemas de configuração do enlace com o provedor, por exemplo);

- Configuração dos serviços ATM: o agente deve ser capaz de exibir informações relativas à configuração dos serviços ATM, como por exemplo:
  - Configuração de ELANs *(Emulated LAN)*: em uma rede ATM onde é empregado o serviço de LAN Emulation, o agente deve ser capaz de exibir as configurações das ELANs existentes, assim como os servidores e clientes ativos de cada ELAN;
  - ARP (*Address Resolution Protocol*) cache: em uma rede ATM com suporte ao serviço CLIP (*Classical IP over ATM*), o agente deve ser capaz de exibir o conteúdo do cache de resolução de endereços IP (*Internet Protocol*) em endereços ATM mantido no servidor ATM-ARP;
- Topologia da rede: o agente deve ser capaz de desenhar grafos topológicos das várias visões da rede. Por exemplo, pode haver a visão da topologia física (nível ATM) ou a visão de sub-redes emuladas (ELANs e/ou LIS – *Logical IP Subnetwork*) através de serviços ATM.

4.2.4 Gerenciamento de Desempenho

Para o gerenciamento CNM é importante a monitoração do funcionamento da rede, de maneira que o usuário possa verificar se ela se comporta de acordo com o previsto, atendendo a níveis

de serviços (SLAs - *Service Level Agreements*) fixados em contrato, relacionados ao desempenho da rede.

Dentro do gerenciamento de desempenho, o agente deve atender aos seguintes requisitos:

- Relatórios e Gráficos de Funcionamento: o agente deve ser capaz de obter estatísticas em tempo real, emitir relatórios e desenhar gráficos de parâmetros de funcionamento, tais como:
  - Estatísticas de enlaces: o agente deve, pelo menos, fornecer o número total de bytes sendo enviados e recebidos por um dado enlace (objetos *ifInOctets* e *ifOutOctets* da MIB-II em interfaces do tipo *atm*);
  - Estatísticas de serviços ATM: Para o serviço CLIP, o agente deve fornecer as estatísticas das requisições de resolução de endereços e respostas recebidas e enviadas pelo servidor ATMARP. Para o serviço LANE (*LAN Emulation*), o agente deve fornecer, ao menos, as estatísticas relativas ao registro de clientes na rede emulada (ELAN) e das requisições de resolução de endereço recebidas e respondidas pelo LES (*Lan Emulation Server*). Deve, também, fornecer estatísticas dos *frames* de *unicast* e *multicast* enviados através do BUS (*Broadcast and Unknown Server*);
- Definição de Limites de Desempenho: o agente deve permitir a definição de limites para os parâmetros de funcionamento da rede (ou seja, para as variáveis que podem ser observáveis através do agente), estabelecendo intervalos aceitáveis de desempenho da rede e emitindo alarmes no caso dos limites serem atingidos.

4.3. Requisitos da Arquitetura de uma solução de gerenciamento CNM

Dada a exigência de grandes recursos computacionais para o gerenciamento de redes ATM e a necessidade de escalabilidade do serviço CNM para atender os gerentes remotos em número

crescente, é natural que uma solução de gerenciamento CNM esteja estruturada em uma arquitetura cliente/servidor, onde:

- Gerente CNM: corresponde ao cliente da arquitetura, através do qual os usuários da rede ATM podem ter acesso aos serviços CNM. É responsabilidade do gerente a interface homem-máquina do sistema e a comunicação segura com o agente CNM;
- Agente CNM: corresponde ao servidor que fornece os serviços de gerenciamento CNM. O agente é responsável pela implementação da maioria das funcionalidades previstas em 4.2. Além disso, o agente CNM deve interagir com o sistema de gerenciamento interno da rede ATM, executando a conversão entre as operações requisitadas pelo usuário (através do gerente CNM) em operações apropriadas junto ao sistema de gerenciamento interno.

Uma arquitetura de serviço CNM para redes ATM deve, na medida do possível, utilizar padrões especificados por entidades de padronização relacionadas à área, tais como o IETF (*Internet Engineering Task Force*) ou o ATM Forum. Desta maneira, a solução não fica restrita a implementações privativas de plataformas de gerenciamento, equipamentos ATM ou mesmo plataformas computacionais. No entanto, como ele deve interagir com elementos externos (por exemplo, a plataforma de gerenciamento), nem sempre é possível utilizar padrões abertos; nestes casos, particularidades de uma dada implementação devem ficar confinadas dentro da arquitetura.

Sendo assim, tanto o gerente como o agente devem possuir uma estrutura modular, facilitando não só a adição e alteração de funcionalidades, como também isolando particularidades de implementação em módulos específicos.

Para uma melhor análise, os requisitos do gerente e do agente devem ser vistos em separado.

4.3.1 Requisitos gerais da arquitetura CNM

O agente CNM é responsável por interagir com os usuários, oferecendo os serviços de gerenciamento CNM. Para tanto, deve atender aos seguintes requisitos:

- Portabilidade: o agente deve suportar diversas plataformas computacionais para a sua execução. Em particular, deve ser capaz de ser executado em plataformas normalmente utilizadas em soluções ATM, tais como: plataforma MS-Windows NT e plataformas Unix: SUN Solaris, IBM AIX, Linux, entre outras;
- Disponibilidade: a arquitetura deve permitir que o serviço esteja sempre disponível aos usuários, ou seja, a queda do enlace ATM ligando uma determinada instituição ao provedor da rede não deve impedir que o serviço seja acessado. Sendo assim, a arquitetura deve permitir que ele seja acessado por um caminho alternativo, além do acesso direto através do *backbone* ATM como, por exemplo, linhas privativas ou até mesmo conexões discadas;
- Interação com Plataformas de Gerenciamento (NMS-*Network Management System*): para fornecer serviços de gerenciamento aos usuários da rede, o agente deve ser capaz de interagir com as plataformas de gerenciamento utilizadas na rede ATM.

4.3.2 Requisitos de Segurança

O administrador de uma instituição ligada a um *backbone* ATM não deve ser capaz de visualizar informações de outras instituições ligadas a esta mesma rede, nem deve ser capaz de executar operações de gerenciamento que interfiram no funcionamento global da rede ou da parte acessada por outra instituição. A questão da segurança é, portanto, fundamental para o oferecimento de um serviço de gerenciamento CNM.

Há dois aspectos de segurança a serem considerados: o controle de acesso aos serviços CNM e o tráfego das informações entre o agente e o gerente CNM.

O acesso às funcionalidades do agente deve ser controlado e, portanto, a solução deve atender aos seguintes requisitos:

- Autenticação: todo acesso de um usuário ao serviço CNM deve ser corretamente identificado; o agente deve garantir que a identificação do usuário não é falsa;
- Controle de Acesso: o acesso do usuário de uma dada instituição deve ser restrito aos serviços e componentes da rede relacionados à instituição à qual ele pertence. O agente deve controlar o acesso do usuário, restringindo seu acesso, baseado no perfil associado a cada usuário autenticado.

As informações relativas aos usuários que podem acessar o sistema e seu perfil devem ser armazenadas em uma base de dados em separado. Esta base deve ser protegida contra acessos indevidos e só deve ser acessada através do sistema de gerenciamento CNM.

Para o tráfego das informações, um canal seguro deve ser estabelecido entre o agente e os gerentes CNM. Para tanto, tanto o agente como os gerentes devem atender aos seguintes requisitos:

- Confidencialidade: o tráfego dos dados entre o agente e os gerentes CNM deve ser criptografado, garantindo o sigilo das operações e dados de gerenciamento;
- Integridade: deve haver mecanismos que garantam que as informações de gerenciamento não sejam modificadas no trajeto entre um gerente e um agente.

4.3.3 Modularidade do Sistema

Tanto o agente como o gerente devem possuir uma estrutura modular, que facilite a adição e alteração de suas funcionalidades.

Como o agente deve implementar as visões lógicas da rede, deve haver os seguintes módulos, relacionados com o particionamento das informações da rede:

- Autenticação;

- Controle de Acesso;
- Comunicação com Bases de Dados.

Deve haver, também, módulos específicos para a comunicação do agente com os outros elementos externos, tais como: os gerentes CNM e a plataforma de gerenciamento da rede:

- Comunicação com Plataforma de Gerenciamento: deve haver um módulo específico para a comunicação com a plataforma de gerenciamento, a partir do qual operações genéricas são mapeadas em operações particulares de uma dada plataforma;
- Comunicação com os Gerentes: o módulo responsável pela comunicação com os gerentes deve atender aos requisitos de segurança de confidencialidade e integridade.

Além destes módulos, deve haver módulos para os serviços de gerenciamento CNM.

Por fim, o gerente deve, também, possuir uma estrutura modular, baseada, pelo menos, em dois módulos:

- Comunicação com o Agente: de maneira análoga ao agente este módulo deve atender aos requisitos de segurança de confidencialidade e integridade;
- Interface Homem-Máquina: o gerente é responsável pela implementação da interface entre os usuários e o sistema de gerenciamento.

Os módulos do sistema devem obedecer a uma hierarquia rígida. Todos os acessos do gerente para o agente passam obrigatoriamente pelos módulos de autenticação e controle de acesso, de maneira a identificar os usuários e restringir o que eles podem fazer no sistema. Caso o usuário tenha permissão para executar uma dada operação, ela é repassada para a camada de serviços CNM que, por sua vez, deve interagir com a plataforma de gerenciamento. Tanto o acesso à base de dados como à plataforma de gerenciamento devem ser feitos pelos módulos correspondentes, isolando assim as particularidades destes sistemas externos.

A Figura 4-2 mostra a relação entre os módulos do sistema.

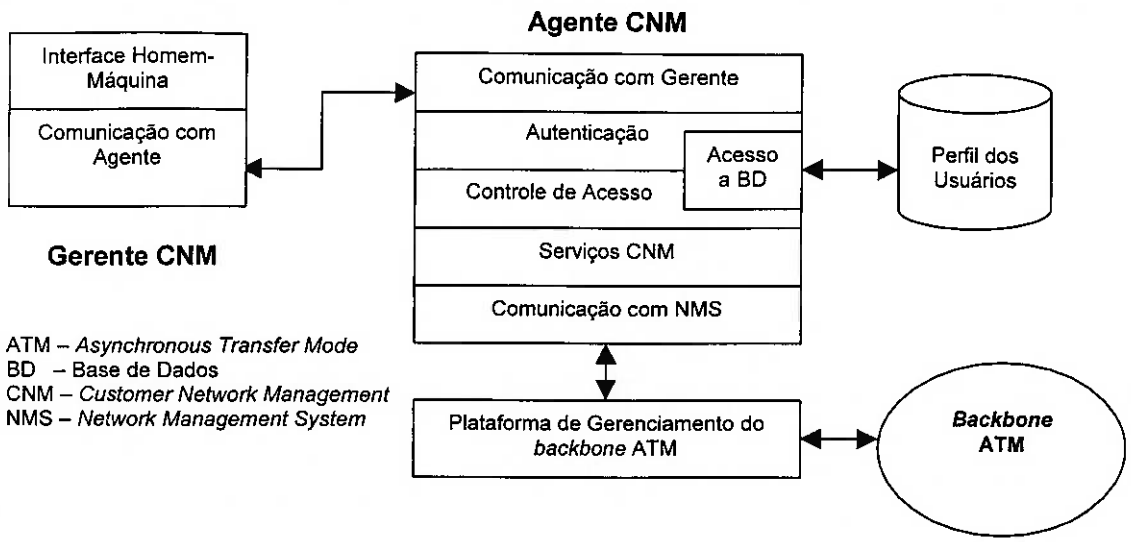


Figura 4-2: Relação entre os módulos da arquitetura de uma solução de gerenciamento CNM

4.3.3.1 Autenticação

Através do módulo de autenticação, o usuário é identificado junto ao sistema. Todos os acessos ao sistema devem ser autenticados, de maneira a satisfazer os requisitos de controle de acesso.

O módulo de autenticação deve suportar diferentes mecanismos de autenticação, de maneira a suportar não somente uma implementação gradual, como também a definição de níveis de segurança diferenciados, de acordo com os requisitos dos usuários do sistema.

As informações necessárias à autenticação do usuário devem ficar armazenadas na base de dados com o perfil dos usuários. As informações críticas de autenticação (como, por exemplo, as senhas dos usuários) devem ser armazenadas de forma criptografada na base de dados; além disso, é necessário que a base de dados esteja protegida, para que a autenticação não seja comprometida.

4.3.3.2 Controle de Acesso

Uma vez identificado o usuário através do módulo de autenticação, o módulo de controle de acesso ao sistema define para este usuário quais recursos ele pode acessar junto ao sistema.

Devem ser definidas diferentes visões lógicas da rede, com diferentes tipos e níveis de permissões de acesso a operações e elementos da rede, de acordo com o perfil de cada usuário. Sendo assim, no módulo de controle de acesso deve ser verificado se :

- a operação requisitada pelo usuário é permitida, de acordo com o seu nível de serviços;
- os elementos da rede a serem manipulados pela operação do usuário encontram-se todos dentro da visão lógica da rede a que ele tem acesso.

Para esta filtragem das operações, este módulo deve recuperar as informações do perfil do usuário junto à base de dados correspondente, utilizando o módulo de comunicação apropriado.

Todas as requisições recebidas dos gerentes devem passar pelo módulo de controle de acesso, de maneira que somente as requisições válidas sejam executadas.

4.3.3.3 Perfil dos Usuários do Sistema

A base de dados com o perfil dos usuários do sistema deve ser acessada através de um módulo específico, que é responsável por mapear as operações genéricas em operações específicas da base de dados implementada.

O perfil dos usuários do sistema deve ser definido através de duas bases de dados relacionadas, contendo informações relativas às instituições que utilizam a rede ATM e dos usuários destas instituições com acesso ao sistema de gerenciamento CNM.

Para definir o perfil de cada usuário, é necessário entender a representação da rede ATM dentro do sistema. Matematicamente a rede pode ser representada através de um grafo não dirigido R=(V, A) onde:

$V = \{ v_k \mid k = 1, \ldots, n \}$ = conjunto dos n vértices (ou nós) de R

$A = \{ a_{ij} \mid i = 1, ..., n-1, \; j = i + 1, ..., n \;$ e a_{ij} representa o enlace entre v_i e $v_j \; \} =$
$=$ conjunto dos enlaces de R

conforme pode ser visto na Figura 4-3.

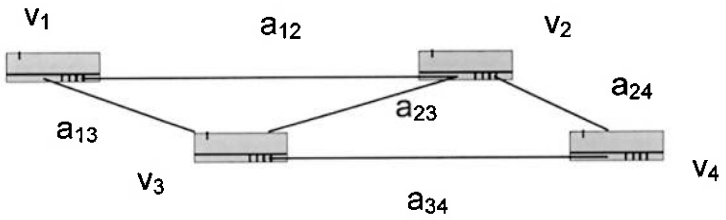


Figura 4-3: Grafo representando uma rede ATM

Como cada instituição e_i utiliza uma parte da rede ATM, o particionamento lógico da rede pode ser representado através de subgrafos $R_i'=(V_i', A_i')$, onde $V_i' \subset V$, $A_i' \subset A$ e $\bigcup R_i' = R$. Sendo assim, R_i' corresponde à visão lógica da rede associada a uma instituição e_i. Os usuários do sistema estão sempre associados a uma instituição e, portanto, associados a uma visão lógica.

Utilizando duas bases de dados adicionais, para a representação dos vértices e arestas, é possível representar as redes lógicas R_i' bem como o perfil dos usuários através de um conjunto de bases de dados relacionais. A Figura 4-4 apresenta o modelo Entidade-Relacionamento destas bases.

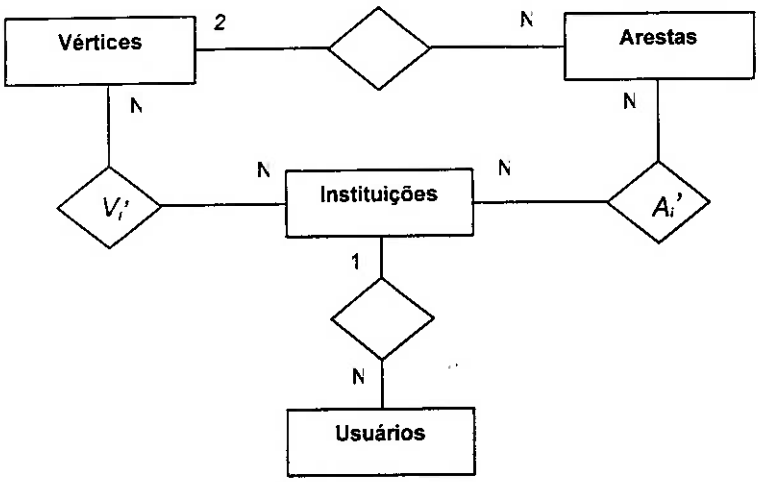


Figura 4-4: Modelo E-R representando redes lógicas $R_i' = (V_i', A_i')$

Portanto, o subgrafo R_i , correspondente à visão lógica da rede particular de uma dada instituição e_i , é representada pelos relacionamentos entre as bases de dados das instituições e as bases de dados de vértices e arestas. Através do relacionamento entre a base de dados do perfil do usuário e a base de dados da instituição, tem-se a relação entre os usuários do sistema e as visões lógicas da rede.

Associado ao perfil do usuário tem-se, também, a definição do nível funcional. Conforme visto anteriormente, o sistema deve apresentar níveis funcionais diferenciados, de maneira a atender diferentes requisitos funcionais dos usuários. O sistema deve, então, definir conjuntos de funcionalidades associados a cada nível funcional, sendo que cada usuário deve ser associado a um nível de serviço.

Na base de dados de perfil dos usuários, deve haver ao menos os seguintes atributos:

- Instituição à qual o usuário pertence;
- Nível funcional associado ao usuário;
- Informações relativas aos mecanismos de autenticação, tais como: senhas ou certificados. Tais informações devem ser armazenadas de forma criptografada.

Além das informações relacionadas à rede ATM, há outras informações que podem ser mapeadas às instituições, relacionadas aos serviços ATM que são oferecidos pelo provedor.

Por exemplo, informações relacionadas a ELANs ou LIS devem ser mapeadas às instituições que as utilizam. Tais informações devem constar, também, na base de dados da instituição.

4.3.3.4 Serviços CNM

O módulo de serviços de CNM é responsável por implementar as funcionalidades previstas no agente CNM. Neste módulo, devem ser mantidos um *log* das transações requisitadas, estatísticas da rede, bem como devem ser tratados e emitidos alarmes referentes a eventos significativos da rede. É responsabilidade deste módulo, também, a criação de relatórios periódicos ou sob demanda.

De acordo com o tipo de serviço, as funções podem ser classificadas em:

- Funções periódicas: estas funções devem ser executadas independentemente de uma requisição do usuário do serviço. Tais funções incluem, por exemplo, o levantamento periódico de estatísticas e do *status* da rede bem como a emissão de alarmes;
- Funções sob demanda: estas funções devem ser executadas quando requisitadas pelo usuário do serviço. Tais funções incluem, por exemplo, o levantamento em tempo real de estatísticas e outras informações da rede, bem como a alteração dos parâmetros das funções periódicas.

4.3.3.5 Comunicação com Plataforma de Gerenciamento

É importante que, na comunicação entre o agente CNM e as plataformas de gerenciamento utilizadas na rede ATM. sejam utilizados padrões abertos de gerenciamento. Isto evita que o agente seja dependente de uma única plataforma de gerenciamento e permite que ele possa ser utilizado junto a várias plataformas sem alterações significativas. No entanto, isto depende da plataforma permitir o acesso através de padrões abertos, o que nem sempre é possível.

O acesso do agente à plataforma de gerenciamento deve ser feito através de funções genéricas, que devem ser mapeadas em funções específicas de cada plataforma. Preferencialmente, as funções genéricas devem ser mapeadas em primitivas SNMP entre o

agente e a plataforma de gerenciamento; caso não seja possível pode-se utilizar APIs (*Application Programming Interface*) de desenvolvimento ou ferramentas de linha de comando.

Estas funções genéricas devem ficar concentradas em uma biblioteca de funções para comunicação com a plataforma de gerenciamento. Desta maneira, a migração do sistema para uma outra plataforma implica apenas na conversão desta biblioteca, ou seja, no mapeamento das funções genéricas desta biblioteca em funções específicas da plataforma.

De maneira análoga, o agente CNM não deve utilizar valores de objetos pertencentes a extensões proprietárias de MIBs, de maneira a evitar dependência da solução em relação a fabricantes de equipamentos ATM e a não restringir a solução a implementações específicas. Sendo assim, a solução deve utilizar única e exclusivamente valores das MIBs padronizadas para ATM.

4.3.3.6 Gerente CNM e Interface Homem-Máquina

Como o agente CNM deve ser acessado por usuários distintos, pertencentes a instituições independentes, o gerente deve ser o mais genérico possível. Para tanto, o gerente deve atender aos seguintes requisitos:

- Universalidade de Acesso: nem todas as instituições ligadas ao *backbone* ATM possuem plataformas de gerenciamento específicas para tecnologia ATM; sendo assim, o agente deve ser acessado a partir de um gerente genérico;
- Portabilidade do Gerente: o gerente para acesso ao agente CNM deve ser portável a diversas plataformas, de maneira a atender ao maior número possível de usuários;
- Facilidade de Uso: o gerente deve apresentar uma interface amigável, de maneira a facilitar a sua utilização pelos usuários do sistema.

A interface do sistema com o usuário deve ser, também, facilmente modificável, possibilitando a sua alteração sem exigir mudanças significativas no agente.

4.3.3.7 Comunicação entre o Agente e o Gerente

O protocolo de comunicação entre o agente e o gerente deve atender aos requisitos de segurança definidos anteriormente. Para tanto este protocolo deve possuir mecanismos de criptografia, garantindo a confidencialidade e integridade da comunicação. Deve, também, incluir um mecanismo para o suporte à autenticação do usuário junto ao sistema.

Preferencialmente, o protocolo deve ser orientado à conexão, garantindo a entrega de mensagens aos usuários quando requisitado. Para protocolos não-orientados à conexão, o agente e o gerente devem implementar primitivas para a confirmação de recebimento de mensagens.

4.4. Considerações Finais

Um serviço CNM para redes ATM apresenta uma série de requisitos, relacionados às funcionalidades a serem oferecidas. Uma solução de gerenciamento CNM deve atender estes requisitos, seguindo os requisitos da arquitetura genérica definidos neste capítulo.

Baseado nos requisitos apresentados neste capítulo, é proposta, no próximo capítulo, uma arquitetura de uma solução via *web* para gerenciamento CNM para redes ATM.

5. Descrição do Agente CNM via *Web*

A arquitetura proposta segue uma arquitetura de aplicações via *web*. O agente CNM (*Customer Network Management*) é composto por um servidor *web* e diversos *scripts*, denominados componentes funcionais, que executam as operações de gerenciamento CNM através de interação com a plataforma de gerenciamento ATM (*Asynchronous Transfer Mode*). A comunicação entre os servidor *web* e os componentes funcionais é feita através do padrão CGI (*Common Gateway Interface* ***[41]***). Associado aos componentes funcionais há um conjunto de bases de dados, contendo o perfil dos usuários do sistema e as redes lógicas associadas ao perfil dos usuários. O gerente da solução é composto por páginas HTML (*HyperText Markup Language*) tradicionais, visualizadas através de um *browser* padrão.

A arquitetura geral pode ser vista na Figura 5-1.

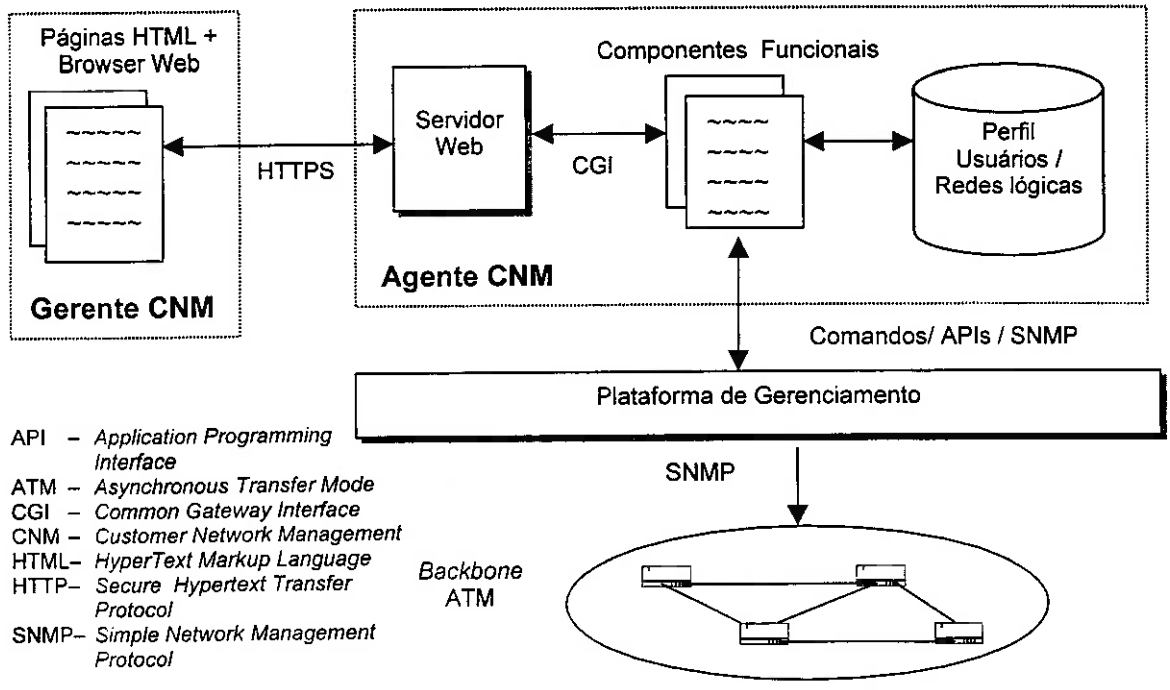


Figura 5-1: Arquitetura Geral da Solução de Gerenciamento CNM

A utilização de uma arquitetura de gerenciamento via *web* atende aos requisitos gerais da arquitetura (descritos em **4.3** e **4.3.1**):

- Arquitetura Cliente/Servidor: uma aplicação *web-based* é baseada em uma arquitetura cliente/servidor, onde os detalhes de comunicação são tratados entre o *browser* e o servidor, não havendo necessidade da se preocupar com estes detalhes no desenvolvimento da aplicação principal;
- Portabilidade do *browser* e do servidor: praticamente todas as plataformas computacionais apresentam *browsers* e servidores *web*, sendo que em diversas delas os *browsers* e servidores já são instalados junto com o próprio sistema operacional;
- Integração com NMS (*Network Management System*): na arquitetura *web* há diversos mecanismos que permitem que páginas HTML sejam construídas dinamicamente, utilizando dados obtidos através da interação com aplicações externas. Com isto, podem ser criadas criar páginas HTML com dados obtidos através da interação com a plataforma de gerenciamento da rede ATM, integrando-se parte das funcionalidades da plataforma à solução CNM.

A utilização de CGI na solução proposta em detrimento de outras alternativas para o desenvolvimento de aplicações *web-based* visa atender aos requisitos de portabilidade da solução. Apesar de mais sofisticadas, alternativas como páginas ASP (*ActiveX Server Pages*) ou PHP (*PHP Hypertext Preprocessor*), servlets Java e extensões ISAPI (*Information Server Application Programming Interface*) ou NSAPI (*Netscape Server Application Programming Interface*) são específicas de algumas implementações de servidores *web*, ao contrário da especificação CGI, que é básica nas diversas implementações existentes.

A arquitetura proposta não utiliza, também, padrões de arquitetura de gerenciamento via *web* em desenvolvimento, tais como o JMX (*Java Management eXtensions*) e o WBEM (*Web-Based Enterprise Management*), pelos seguintes motivos:

- O JMX encontra-se, ainda. em processo de definição da sua arquitetura. Apesar das especificações para a criação de um agente JMX estarem completas. os gerentes JMX e os mecanismos para a comunicação com agentes JMX ainda não foram

completamente especificados, não sendo possível o desenvolvimento da solução sobre este padrão;

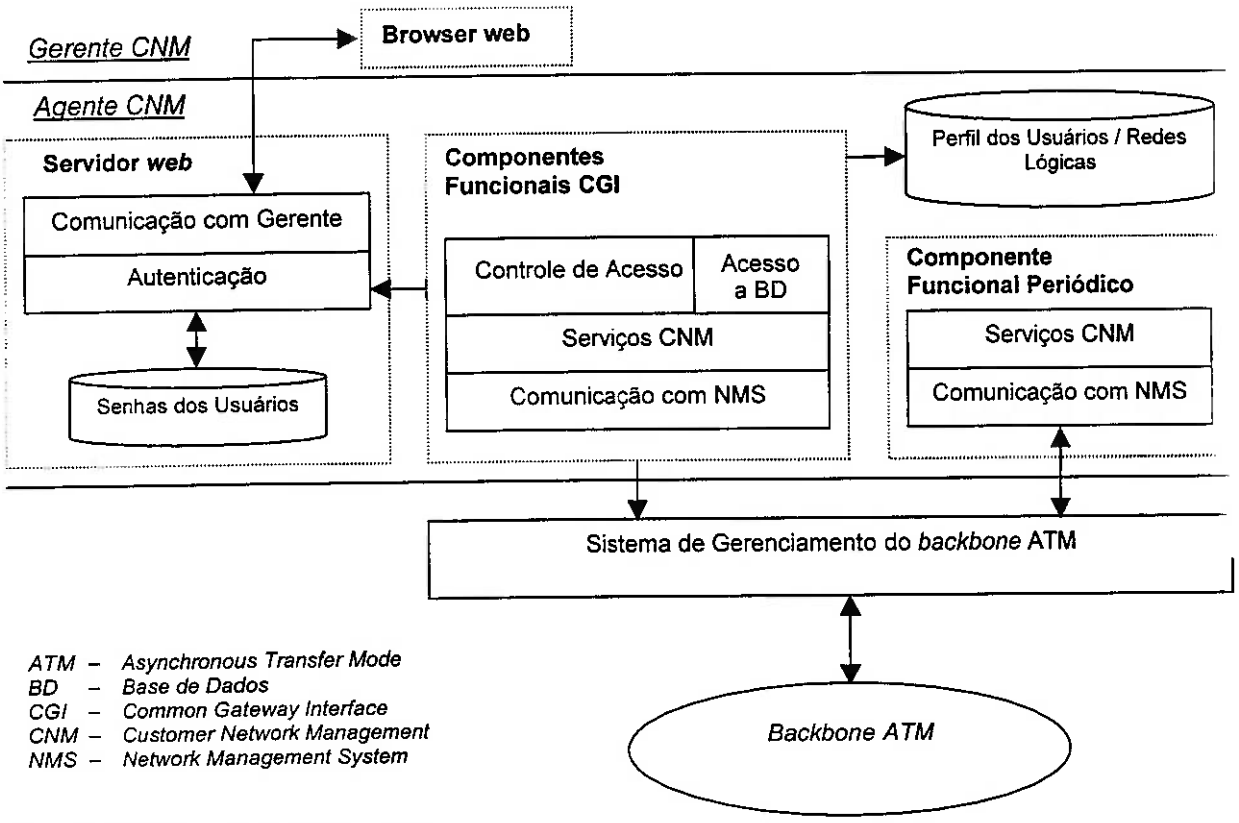
- Apesar da especificação básica do WBEM estar mais definida, ela exige a utilização de um modelo de informações (CIM - *Common Information Model*) diferente dos modelos tradicionais de informações de gerenciamento. Como não há ainda classes CIM definidas para as informações de gerenciamento ATM, optou-se por utilizar o modelo de informações da arquitetura SNMP (*Simple Network Management Protocol*), dado o maior suporte da arquitetura por parte dos diversos elementos de rede ATM e plataformas de gerenciamento, viabilizando, assim, a implementação de um protótipo da arquitetura proposta.

5.1. Módulos da arquitetura

Conforme visto anteriormente, a arquitetura do agente é composta pelos seguintes módulos:

- Um servidor *web;*
- Um conjunto de *scripts*, denominados componentes funcionais, composto por diversas bibliotecas de funções;
- *Browsers web* para acesso ao sistema.

A Figura 5-2 apresenta o papel destes módulos na arquitetura proposta, relacionando-os com os módulos especificados em **4.3.3** (**Figura 4-2**). As funções dos módulos da arquitetura



proposta são apresentadas a seguir.

Figura 5-2: Módulos da arquitetura de gerenciamento da solução CNM proposta

5.1.1 Servidor *Web*

De acordo com a especificação dos requisitos da arquitetura de uma solução de gerenciamento CNM, o servidor *web* encarrega-se das funções de dois módulos definidos anteriormente:

- Módulo de autenticação;
- Módulo de comunicação com o gerente.

As funcionalidades destes módulos já se encontram implementadas no servidor *web*, sendo somente necessária a configuração apropriada para a sua utilização dentro da arquitetura de gerenciamento CNM.

5.1.1.1 Protocolos de comunicação

Para a comunicação entre o gerente e o agente, o protocolo de comunicação utilizado é o chamado HTTPS, que consiste da combinação do protocolo HTTP (*Hypertext Transfer Protocol*) e do protocolo de segurança SSL (*Secure Sockets Layer* **[29]**), garantindo uma conexão criptografada entre o agente e o gerente. Com isto, são atendidos não só os requisitos de segurança de confidencialidade e integridade (definidos em **4.3.2**), como também o requisito do protocolo entre os gerentes e o agente ser orientado à conexão, conforme definido em **4.3.3.7**.

5.1.1.2 Módulo de Autenticação

Foram especificados para o protocolo HTTP **[13]** funcionalidades básicas de autenticação, que se encontram implementadas nos diversos *browsers* e servidores *web* existentes. Sendo assim, o módulo de autenticação do agente CNM é implementado diretamente pelo servidor *web*: basta configurá-lo para que todo acesso ao conjunto de páginas e *scripts* CGI do agente seja autenticado. O servidor encarrega-se da comunicação com o browser que, por sua vez, solicita as informações de autenticação do usuário do sistema.

Com este tipo de configuração, o servidor *web* somente executa os *scripts* CGI se o usuário estiver corretamente autenticado, independente do mecanismo de autenticação utilizado (é possível definir outros mecanismos para autenticação dentro do protocolo HTTP como, por exemplo, certificados digitais). Uma vez o usuário autenticado, o servidor *web* envia automaticamente aos *scripts* CGI a identificação deste usuário dentro do sistema.

As informações necessárias para a autenticação dos usuários junto ao servidor *web* ficam em uma base de dados própria do servidor e não na base de dados do perfil do usuário.

5.1.2 Gerente CNM

O gerente do sistema é composto por um *browser* HTML padrão, atendendo aos requisitos funcionais dos módulos relativos à interface homem-máquina (composta por páginas HTML simples e imagens) e de comunicação com o agente CNM, conforme especificado em **4.3.3.6** (Figura 5-3).

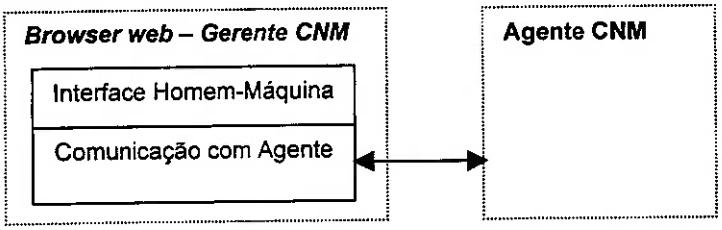


Figura 5-3: Módulos do Agente CNM da arquitetura proposta

Não foi utilizado nenhum tipo de função que exija o processamento do lado do gerente, tais como linguagens de *scripts* (*Javascript, Vbscript*), concentrando as funcionalidades no agente. Desta maneira, evita-se a restrição da solução a implementações particulares de *browsers* e a ocorrência, também, de problemas com versões de linguagens de *scripts* implementadas nos *browsers.*

A utilização do *browser* atende, também, aos requisitos de portabilidade e universalidade de acesso para o gerente (**4.3.3.6**). Apesar do trabalho não incluir um estudo de usabilidade para a definição da interface homem-máquina, a utilização de um *browser* visa, também, explorar a familiaridade dos usuários com a navegação em páginas HTML na Internet, obtendo-se uma interface de fácil uso.

5.1.3 Componentes Funcionais

Os componentes funcionais são responsáveis pelas funcionalidades de gerenciamento CNM do agente. A arquitetura do agente é composta por dois tipos de componentes funcionais:

- Componentes CGI: funções que requerem interação com os usuários do sistema. Os componentes CGI são executados somente quando é feita uma requisição, através de um *browser*, utilizando parâmetros fornecidos pelos usuários;
- Componente Periódico: funções que devem ser executadas de forma periódica, sem interação com o usuário. Como não há interação com usuários (o componente periódico é executado pelo sistema operacional), seus parâmetros de execução devem ser definidos em arquivos de configuração.

A interação entre estes componentes pode ser vista na Figura 5-4. Conforme visto, todo o acesso dos usuários ao sistema é feito através de um *browser*, sendo a requisição recebida pelo servidor *web* (item *1a* da Figura 5-4). O servidor *web* executa a autenticação do usuário (*2a*); caso a autenticação seja bem-sucedida, o servidor *web*, através do padrão CGI, passa a requisição para um componente CGI (*3a*), que deve executar as funções de controle de acesso aos serviços CNM (*4a*). Se a requisição demandar uma interação imediata do sistema com a rede, o componente CGI encarrega-se de interagir com a plataforma de gerenciamento (*5a*), devolvendo os resultados desta interação ao *browser*. Caso a requisição envolva a definição de coletas periódicas de dados ou a verificação de alarmes, o componente CGI encarrega-se de alterar os arquivos de configuração do componente periódico (*5b*) de maneira que, na próxima execução periódica, sejam utilizadas as novas definições de coleta de dados e alarmes. Uma requisição ao componente CGI pode envolver, também, a geração de gráficos ou relatórios baseados em dados coletados periodicamente; neste caso, o componente CGI deve acessar os arquivos com estes dados (*5b*).

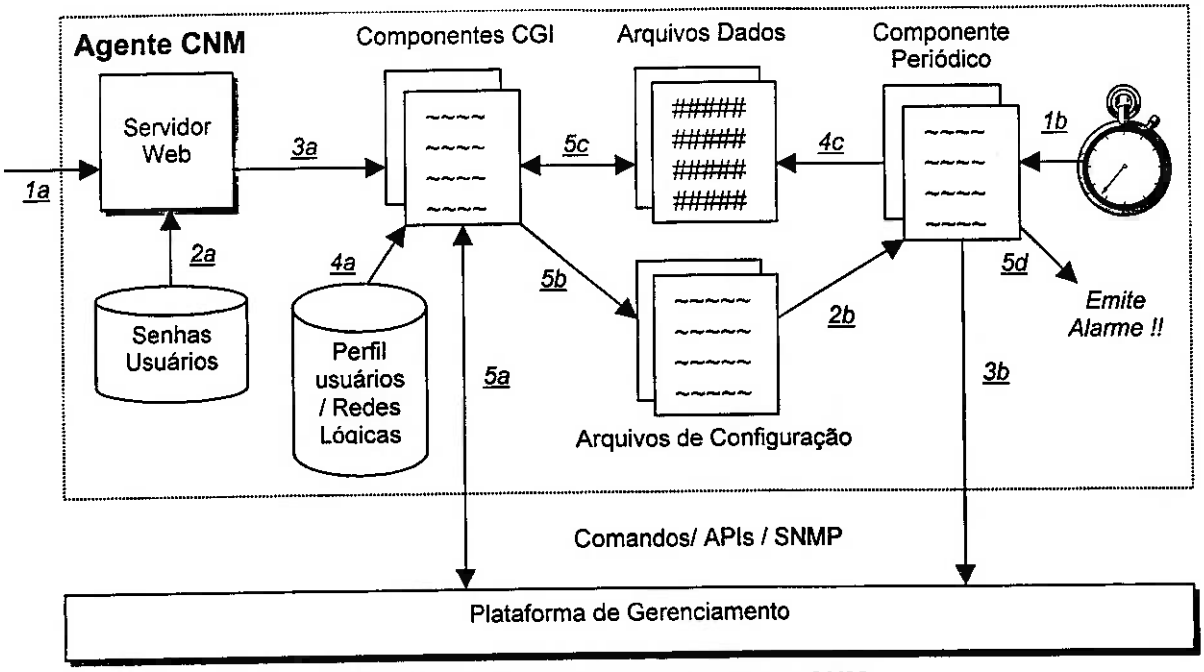


Figura 5-4: Fluxo Interno do Agente CNM

O componente periódico, por sua vez, é executado pelo sistema operacional (*1b*) . Através dos arquivos de configuração, o componente periódico verifica quais as variáveis devem ser coletadas (*2b*), interagindo com a plataforma de gerenciamento para obter os valores destas variáveis (3*b*). Os valores coletados são, então, armazenados em arquivos de dados para uso posterior (*4c*) do componente CGI. Se houver algum alarme definido para uma dada variável, o componente periódico verifica os limites definidos e, caso seja necessário, emite um alarme (*5d*).

5.1.3.1 Componente CGI

Os componentes CGI são compostos por um conjunto de *scripts* CGI, onde as funcionalidades do sistema devem ser implementadas através de bibliotecas de funções. De maneira geral, os *scripts* devem executar os passos descritos na Tabela 5-1, de acordo com a ordem apresentada, conforme a divisão de módulos funcionais definida no capítulo anterior:

Camada de Funções	**Passo executado**
Comunicação com gerente	Valida os parâmetros enviados pelo *browser* do gerente.
Controle de Acesso	Verifica se o usuário tem permissão para executar a função.
	Se o acesso não for permitido, então registra no *log* a operação requisitada pelo gerente e o porquê do erro e retorna mensagem de erro ao gerente.
	No caso de acesso permitido, continua execução.
Serviços CNM	Registra no *log* a função requisitada.
	Executa a função.
Comunicação com gerente	Retorna o resultado para o gerente.

Legenda: CNM *Customer Network Management*

Tabela 5-1: Passos da execução de um *script* CGI genérico do sistema no agente

Através do padrão CGI, os componentes CGI recebem do servidor *web* os parâmetros enviados pelo gerente CNM (via *browser*), bem como a identificação do usuário dentro do sistema, de acordo com a autenticação executada.

5.1.3.2 Componente Periódico

O componente periódico é composto por um *script* que é executado pelo sistema operacional. Assim como no componente CGI, as suas funcionalidades devem ser implementadas através de bibliotecas de funções, uma vez que compartilha funções com o componente CGI (em particular as bibliotecas de funções de comunicação com a plataforma de gerenciamento). Dentro do componente periódico não há necessidade de controle de acesso ou autenticação, uma vez que os arquivos de configuração do sistema só podem ser alterados através do componente CGI, que se encarrega da autenticação e controle de acesso. Todas as funções executadas pelo componente periódico encontram-se dentro do contexto de serviços CNM.

Como cada elemento da rede pode ser compartilhado por diversas instituições, elas podem definir alarmes e relatórios para uma mesma variável em um mesmo elemento da rede (por exemplo, um *switch* ATM). Para cada execução do componente periódico, as variáveis de cada elemento da rede devem ser coletadas uma única vez, minimizando interações com a plataforma de gerenciamento. É responsabilidade do componente periódico ordenar a coleta de

informações, de maneira que a freqüência da coleta seja minimizada para não interferir no funcionamento da rede.

A Tabela 5-2 resume as operações a serem executadas pelo componente periódico.

Camada de Funções	**Passo executado**
Coleta Periódica	Coleta os valores das variáveis monitoradas nos diversos elementos da rede, conforme arquivos de configuração.
Alarmes	Para cada variável monitorada, verifica se há um alarme definido.
	Se houver um alarme definido e se os limites deste alarme foram atingidos, então notifica a instituição correspondente e gera uma entrada nos *logs* do sistema.
Armazenamento dos dados	Armazena os dados coletados, que podem ser utilizados para a geração de relatórios e gráficos quando necessário.

Tabela 5-2: Passos da execução do componente periódico

5.1.4 Bibliotecas de funções do sistema

As bibliotecas de funções são responsáveis por implementar as funcionalidades do sistema, sendo utilizadas pelos componentes funcionais. A divisão em bibliotecas visa facilitar a adição e alteração de funcionalidades.

Não há, nas bibliotecas, funções separadas para um enlace da rede, um serviço ou um *switch* ATM, sendo todas as funções definidas para elementos da rede ATM (por exemplo, há somente uma função para a coleta de dados, e não funções separadas para a coleta de dados de *switches* e de enlaces). Desta maneira, pode-se definir um código único para os componentes funcionais, sendo a distinção feita dentro das funções. Com isto, minimizam-se alterações no código do agente, caso se deseje incorporar novos elementos de rede a serem gerenciados (por exemplo, um novo serviço na rede).

5.1.4.1 Funções de controle de acesso

A biblioteca de controle de acesso deve apresentar apenas uma função pública (visível às outras funções do sistema), que indica se o usuário pode ou não executar a operação requisitada, de acordo com a Tabela 5-3.

Acesso_permitido	Verifica se o usuário tem permissão para executar a operação requisitada nos elementos de rede desejados, devolvendo um valor booleano (verdadeiro ou falso).
	Entrada: ▪ Nome do usuário no sistema; ▪ Código da operação requisitada; ▪ Lista de elementos de rede a serem manipulados.
	Saída: ▪ Valor booleano (verdadeiro ou falso), indicando se o usuário pode ou não executar a operação requisitada sobre os elementos desejados .

Tabela 5-3: Funções da biblioteca de controle de acesso do sistema

O nome do usuário, utilizado como parâmetro para a função *acesso_permitido*, é a identificação do usuário dentro do sistema, após ter sido corretamente autenticado pelo servidor *web*. Para determinar se o usuário pode ou não executar a operação requisitada (se o acesso do usuário é permitido), a função de controle de acesso deve:

- Recuperar da base de dados a visão lógica R da rede da instituição e o nível funcional do usuário, baseada na informação de autenticação enviada pelo servidor *web*;
- Verificar se a função requisitada é permitida pelo nível funcional do usuário;
- Verificar se os parâmetros enviados pelo gerente contêm elementos de rede que não pertencem a R.

Estas verificações devem ser feitas através de duas funções auxiliares, descritas na Tabela 5-4. Estas funções são internas ao módulo de controle de acesso, sendo utilizadas exclusivamente pela função *acesso_permitido* (Tabela 5-3).

Operacao_permitida	Verifica se o usuário tem permissão para executar a operação requisitada, de acordo com o nível funcional associado ao usuário, devolvendo um valor booleano (verdadeiro ou falso).
	Entrada: ▪ Nome do usuário no sistema; ▪ Código da operação requisitada.
	Saída: ▪ Valor booleano (verdadeiro ou falso), indicando se o usuário pode ou não executar a operação requisitada.
Elemento_permitido	Verifica se o usuário tem permissão para manipular o elemento da rede (nó ou enlace) desejado, de acordo com a visão lógica da rede associada à instituição do usuário. Devolve um valor booleano (verdadeiro ou falso).
	Entrada: ▪ Instituição; ▪ Lista de elementos de rede a serem manipulados.
	Saída: ▪ Valor booleano (verdadeiro ou falso), indicando se o usuário pode ou não manipular o elemento de rede desejado.

Tabela 5-4: Funções Internas da biblioteca de controle de acesso do sistema

Para a função *operacao_permitida*, definida na Tabela 5-4, deve haver uma tabela com a lista das operações definidas no sistema e o nível funcional exigido para a execução de cada operação.

5.1.4.2 Funções para acesso ao perfil do usuário

O acesso às informações relativas ao perfil do usuário deve ser feito através da função Perfil_usuario, descrita na Tabela 5-5. Esta função é responsável por esconder do resto do sistema as particularidades de implementação da base de dados do perfil dos usuários do sistema.

Perfil_usuario	A partir dos dados de um usuário, retorna todos os dados relativos ao seu perfil.
	Entrada: ▪ Usuário.
	Saída: ▪ Instituição à qual o usuário pertence; ▪ Nível funcional associado ao usuário; ▪ Visão lógica da rede (Grafo R).

Tabela 5-5: Função pública da biblioteca de acesso ao perfil do usuário

5.1.4.3 Funções CGI

Para facilitar o desenvolvimento de *scripts* CGI para o sistema, foi definida uma biblioteca para a criação destes componentes funcionais CGI, contendo as funções descritas na Tabela 5-6.

Cnm_cgi	Executa as funções de comunicação com o gerente e controle de acesso, conforme definidos em 5.1.3.1, executando os registros apropriados no *log*
	Entrada: ▪ Informações enviadas pelos formulários HTML; ▪ Código da operação requisitada pelo *script* CGI, para controle de acesso e registro no *log*.
	Saída: ▪ Permite ou não a continuação da execução do *scripts* CGI
Erro_html	Mostra uma mensagem de erro e encerra a execução do *script* CGI.
	Entrada: ▪ Mensagem de erro.
	Saída: ▪ Página HTML com mensagem de erro do sistema

Legenda: HTML *Hypertext Markup Language*
CGI *Common Gateway Interface*

Tabela 5-6: Funções públicas da biblioteca para componentes funcionais CGI

Para a criação de um novo *script* CGI para o sistema, basta iniciar o *script* com uma chamada para a função *cnm_cgi* : se houver algum problema relativo à comunicação ou ao controle de acesso, o *script* CGI é automaticamente encerrado, exibindo mensagens de erro apropriadas ao gerente. Desta maneira, caso a chamada à função *cnm_cgi* seja bem-sucedida, é assegurado ao sistema que o usuário foi corretamente autenticado e possui permissão para a operação requisitada, bem como os parâmetros do formulário foram corretamente validados.

5.1.4.4 Funções CNM

Conforme visto em **4.3.3.4** (Requisitos da arquitetura de um agente CNM), as funções do sistema podem ser divididas em duas classes distintas: as que são executadas periodicamente e as que são executadas sob demanda de um usuário do sistema. As bibliotecas de funções CNM são compartilhadas pelos dois tipos de componentes funcionais do sistema.

5.1.4.4.1 Funções de *Log*

O módulo de funções de *log* do sistema deve fornecer ao sistema duas funções, de acordo com a Tabela 5-7.

Registra_log	Registra a ocorrência de evento no arquivo de *log* .
	Entrada: ▪ Hora e data da ocorrência; ▪ Usuário e instituição responsável pelo evento; ▪ Mensagem de explicação do evento ocorrido; ▪ Código de identificação do evento; ▪ Categoria do evento.
	Saída: ▪ Arquivo de *log* atualizado, contendo um novo registro com os parâmetros de entrada.
Mostra_logs	Mostra os registros de *log* de acordo com parâmetros de busca fornecidos como entrada da função. Podem ser utilizadas diversos parâmetros de entrada para a busca; os parâmetros não utilizados devem vir em branco.
	Entrada: ▪ Hora e data de início do intervalo de busca do evento; ▪ Hora e data do fim do intervalo de busca do evento; ▪ Usuário e instituição responsável pelo evento; ▪ Código de identificação do evento procurado; ▪ Categoria do evento procurado; ▪ Número de mensagens do *log* a serem exibidas.
	Saída: ▪ Lista com os registros desejados, contendo as seguintes informações em cada registro: ➢ Hora e data da ocorrência; ➢ Usuário e instituição responsável pelo evento registrado; ➢ Mensagem de explicação do evento registrado; ➢ Código de identificação do evento registrado; ➢ Categoria do evento.

Tabela 5-7: Funções da biblioteca de *logs* do sistema

Cada registro no arquivo de *logs* é identificado por um código, através do qual é possível correlacionar registros diferentes correspondentes a uma mesma operação ou conjunto de operações (por exemplo, uma requisição e o resultado da requisição). Além disso, os eventos devem ser classificados em algumas categorias, de maneira que se possa procurar e visualizar registros de uma dada categoria (por exemplo, podem ser definidas uma categoria para operações de coleta de dados, uma para operações envolvendo alarmes e uma para operações de configuração do sistema).

Os registros de *log* de uma dada instituição só devem ser vistos por seus usuários; sendo assim, devem haver *logs* separados para cada instituição.

5.1.4.4.2 Funções de Alarme

O sistema deve permitir que sejam definidos um limite superior e um inferior para os valores das variáveis coletadas periodicamente. Caso algum destes limites seja ultrapassado, o sistema deve emitir alarmes.

Estes alarmes devem ser configurados de acordo com os tipos definidos em **4.2.1**. Além disso, o sistema deve permitir a definição de diferentes métodos de notificação para os usuários como, por exemplo, notificação via *e-mail* ou via *trap* SNMP.

As funções de alarme são definidas na Tabela 5-8:

Definir_alarme	Define um alarme para ser verificado pelo módulo periódico
	Entrada: ▪ Elemento da rede e objeto da MIB do elemento a ser monitorado por situações de alarmes; ▪ Limite superior e inferior para disparo de alarmes; ▪ Tipo de alarme; ▪ Método de notificação; ▪ Parâmetro específico da notificação.
	Saída: ▪ Arquivo de configuração atualizado, contendo uma nova definição de alarme.
Exibir_alarmes	Mostra quais alarmes foram definidos por usuários de uma dada instituição. Pode exibir todos os alarmes ou apenas alarmes definidos em relação a um determinado elemento da rede.
	Entrada: ▪ Elemento de rede monitorado; ▪ Usuário e instituição.
	Saída: ▪ Lista dos alarmes definidos no sistema contendo, para cada alarme, os seguintes dados: ➢ Elemento da rede e objeto da MIB do elemento a ser monitorado por situações de alarmes; ➢ Limite superior e inferior para disparo do alarme; ➢ Tipo de alarme; ➢ Método de notificação; ➢ Parâmetro específico da notificação.
Verificar_alarmes	Verifica se, para um dado elemento de rede e variável, os limites de disparo de um alarme foram atingidos, indicando se o alarme deve ou não ser disparado.
	Entrada: ▪ Elemento de rede e valor da variável monitorada; ▪ Limites superior e inferior para disparo do alarmes.
	Saída: ▪ Valor booleano que indica se o alarme deve ser disparado (verdadeiro) ou não (falso).
Remover_alarme	Remove um alarme definido previamente
	Entrada: ▪ Elemento de rede e variável a ser monitorado por situações de alarmes; ▪ Limite superior e inferior para disparo de alarmes; ▪ Tipo do alarme; ▪ Método de notificação; • Parâmetro específico da notificação.
	Saída: ▪ Arquivo de configuração atualizado, sem a definição do alarme.
Notificação	Emite uma notificação, conforme configuração do alarme.
	Entrada: ▪ Elemento de rede e variável monitorada; ▪ Limites definidos para disparo do alarme e valor observado; ▪ Tipo do alarme; ▪ Método de notificação; ▪ Parâmetros da notificação.

	Saída: ▪ Alarme disparado, conforme parâmetros de entrada.

Tabela 5-8: Funções de alarme do sistema

As notificações dos alarmes são especificadas através de funções específicas, facilitando a adição de novos tipos de alarmes. Pelo menos duas funções devem ser definidas para a notificação dos alarmes, a *notificação_email* e *trap_snmp* (Tabela 5-9).

Notificacao_email	Envia um e-mail para o responsável pela definição do alarme, conforme endereço contido no parâmetro do alarme
	Entrada: ▪ Elemento de rede e variável monitorada; ▪ Limite atingido para disparo do alarme e valor observado; ▪ Endereço de e-mail do responsável pelo alarme.
	Saída: ▪ Notificação via e-mail enviada, conforme parâmetros de entrada.
Trap_snmp	Emite uma *trap* SNMP para uma plataforma de gerenciamento, determinada na configuração do alarme.
	Entrada: ▪ Endereço IP do gerente SNMP a ser notificado; ▪ Elemento de rede relacionado ao alarme; ▪ Variável e valor da variável monitorado.
	Saída: ▪ *Trap* SNMP enviada, conforme parâmetros de entrada.

Legenda: SNMP *Simple Network Management Protocol*

Tabela 5-9: Funções para notificação de alarmes emitidos

Dada a dependência da plataforma de gerenciamento, a função *trap_snmp* deve estar contida na biblioteca de comunicação com a plataforma de gerenciamento (Tabela 5-13).

5.1.4.4.3 Funções de Coleta Periódica

A funcionalidade da coleta periódica é utilizada como base para as funcionalidades de alarmes e para a geração de gráficos e relatórios. As funções de coleta periódica são definidas na Tabela 5-10.

Definir_coleta	Define a coleta periódica do valor de uma variável de um dado elemento da rede.
	Entrada: ▪ Elemento de rede a ser monitorado; ▪ Variável cujos valores serão coletados; ▪ Periodicidade da coleta.
	Saída: ▪ Arquivo de configuração atualizado, contendo o registro de uma nova variável a ser monitorada.
Exibir_coleta	Mostra quais variáveis e elementos de rede estão sendo monitorados pelo agente, de acordo com o usuário e instituição.
	Entrada: ▪ Usuário e instituição responsável pelo elemento de rede monitorado.
	Saída: ▪ Lista das monitorações configuradas no sistema, contendo os seguintes dados das monitorações definidas: ➢ Elementos da rede e objetos da MIB dos elementos a serem monitorado; ➢ Periodicidade da coleta.
Remover_coleta	Remove uma coleta periódica definida previamente.
	Entrada: ▪ Elemento de rede e variável monitorados; ▪ Periodicidada da coleta.
	Saída: ▪ Arquivo de configuração atualizado, sem o registro da coleta periódica .
Coletar_dados	Interage com a plataforma de gerenciamento, coletando os dados requisitados.
	Entrada: ▪ Elemento de rede e variável monitorados.
	Saída: ▪ Arquivos de dados atualizados com os dados requisitados.

Tabela 5-10: Funções de coleta periódica do sistema

Conforme visto em 5.1.3.2, os dados coletados são armazenados, para serem utilizados na geração de relatórios e gráficos, quando necessário.

5.1.4.4.4 Funções de Gerenciamento de Configuração

Como os requisitos do sistema proposto neste trabalho são voltados para monitoração de uma rede ATM, há apenas duas funções definidas para o gerenciamento de configuração, indicada na Tabela 5-11.

Ver_config_equip	Exibe os parâmetros de configuração de um elemento da rede
	Entrada: ▪ Elemento de rede cujas configurações serão exibidas.
	Saída: ▪ Página HTML contendo parâmetros da configuração do elemento da rede.
Ver_VC	Exibe os parâmetros de configuração relativos aos trechos do circuito virtual (um dado VPLs ou VCLs) que passa por um elemento da rede.
	Entrada: ▪ Elemento de rede cujo VPL/VCL deseja-se exibir; ▪ VPI/VCI do VPL/VCL.
	Saída: ▪ Parâmetros do descritor de tráfego utilizado; ▪ Entrada na tabela de comutação do elemento da rede

Legenda: VCI *Virtual Channel Identifier*
VCL *Virtual Channel Link*
VPI *Virtual Path Identifier*
VPL *Virtual Path Link*

Tabela 5-11: Funções de gerenciamento de configuração

5.1.4.4.5 Funções de Gráfico e Relatório

As funções de gráfico e relatório, apresentadas na Tabela 5-12, utilizam-se dos arquivos de dados contendo o valor das variáveis monitoradas pelo componente periódico. Estas funções geram uma saída em HTML, devendo esta saída ser enviada diretamente para o gerente CNM.

Relatorio	Gera relatório, em formato HTML, sobre o comportamento da variável de um dado elemento da rede, ao longo das coletas feitas pelo componente periódico.
	Entrada: ▪ Elemento da rede e variável monitorada.
	Saída: ▪ Página HTML com um relatório do comportamento da variável monitorada.
Gráfico	Gera um gráfico sobre o comportamento da variável de um dado elemento da rede, ao longo das coletas feitas pelo componente periódico.
	Entrada: ▪ Elemento da rede e variável monitorada.
	Saída: ▪ Página HTML contendo um gráfico do comportamento da variável monitorada.

Legenda: HTML *Hypertext Markup Language*

Tabela 5-12: Funções de relatório

5.1.4.5 Funções para acesso à plataforma de gerenciamento

O agente deve utilizar, exclusivamente, de informações presentes em MIBs SNMP padronizadas. Para tanto, as seguintes funções são necessárias ao agente:

Ler_dados	Lê os valores de um conjunto de variáveis da MIB SNMP de um elemento da rede, através da interação com a plataforma de gerenciamento
	Entrada: ▪ Elemento de rede e conjunto de variáveis a serem monitorados.
	Saída: ▪ Valores das variáveis desejadas.
Trap_snmp	Emite uma *trap* SNMP para o gerente SNMP especificado, para a notificação de alarmes ocorridos na rede.
	Entrada: ▪ Endereço IP do gerente SNMP a ser notificado; ▪ Elemento de rede relacionado ao alarme; ▪ Variável e valor da variável monitorado.
	Saída: ▪ *Trap* SNMP enviada, conforme parâmetros de entrada.

Legenda: IP *Internet Protocol*
MIB *Management Information Base*
SNMP *Simple Network Management Protocol*

Tabela 5-13: Funções de acesso à plataforma de gerenciamento

Tanto a função *ler_dados* como a *trap_snmp* são dependentes do tipo de plataforma de gerenciamento da rede ATM; sendo assim dentro desta biblioteca devem haver funções específicas para a utilização de diferentes plataformas.

5.1.5 Relação entre componentes funcionais e funções das bibliotecas

As funções das bibliotecas podem ser utilizadas tanto por outras funções como diretamente pelos componentes funcionais. Na Figura 5-5 podem ser vistas as funções que o componente funcional necessita para o seu funcionamento, tanto diretamente (*coletar_dados* e *verificar_alarmes*) como indiretamente (*ler_dados, notificação, notificação_email* e *trap_snmp).*

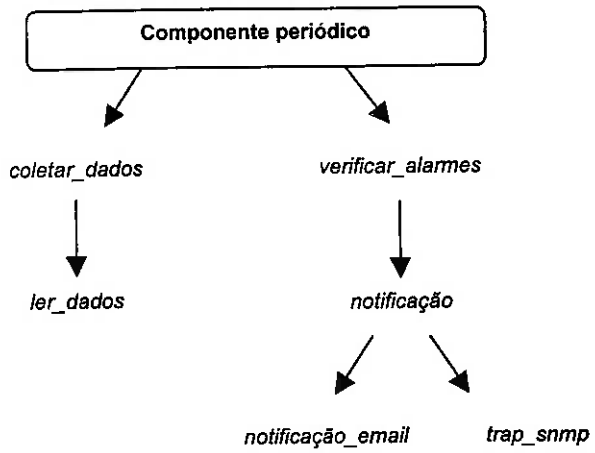


Figura 5-5: Relação entre o componente periódico e as funções das bibliotecas

A Figura 5-6 apresenta a relação entre os componentes funcionais CGI e as bibliotecas de funções. Como diferentes componentes CGI executam diferentes funcionalidades do sistema, algumas funções das bibliotecas são utilizadas dependendo da funcionalidade do componente CGI (em destaque na Figura), enquanto outras funções são utilizadas em todos os componentes CGI.

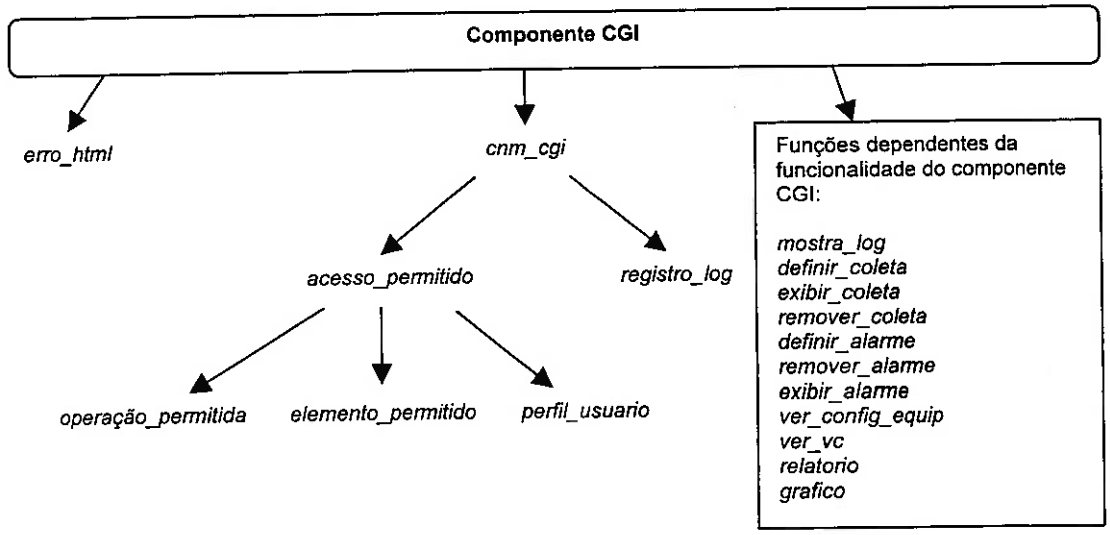


Figura 5-6: Relação entre os componentes CGI e as funções das bibliotecas

5.2. Implementação

A partir da arquitetura da solução e das bibliotecas de funções, especificada em 5.1, foi implementado um protótipo da arquitetura gerenciamento CNM proposta. Os itens a seguir tratam dos aspectos de implementação desse protótipo.

5.2.1 Linguagens Utilizadas

Os componentes funcionais foram desenvolvidos utilizando-se a linguagem Perl **[58]**, devido aos seguintes benefícios:

- Rapidez de Desenvolvimento: por ser uma linguagem interpretada, permite uma rápida prototipação;
- Portabilidade: existem versões do interpretador da linguagem para diversas plataformas (em particular para Windows NT e diversas versões de Unix), atendendo aos requisitos de portabilidade;
- Facilidade para a criação de *scripts* CGI: por ser uma linguagem bastante utilizada no desenvolvimento de *scripts* CGI, há diversas bibliotecas que facilitam o desenvolvimento, bem como uma grande variedade de material de referência sobre o assunto. Isto facilita não só o desenvolvimento deste trabalho como também futuras extensões. Para o sistema foi utilizada a biblioteca CGI.pm, parte integrante da linguagem;
- Modularidade: a linguagem é extremamente modular, permitindo a criação e utilização de bibliotecas. Com isto é possível estruturar o agente de maneira que seus módulos possam ser facilmente reutilizados posteriormente;
- Extensibilidade: além da possibilidade de desenvolver bibliotecas, a linguagem oferece diversos mecanismos para a integração com outras linguagens de desenvolvimento (tais como as linguagens C e Java), com programas externos (por exemplos,

comandos Unix) e bases de dados. Tal característica é extremamente importante para atender ao requisito de integração com a plataforma de gerenciamento ATM.

5.2.2 Servidor *web*

O servidor *web* utilizado foi o Apache. A sua escolha deveu-se aos seguintes fatores:

- Portabilidade: assim como a linguagem Perl, o Apache suporta diversas plataformas; em particular suporta as plataformas definidas como requisitos para o agente CNM (Windows NT/ Unix);
- Modularidade: o servidor Apache possui uma estrutura modular, de maneira que novas funcionalidades podem ser adicionadas de acordo com a necessidade. Tal característica é importante por permitir que as funcionalidades de segurança necessárias ao gerenciamento CNM sejam agregadas ao servidor *web*. Algumas destas funcionalidades, como por exemplo a parte de criptografia, nem sempre podem ser utilizadas em alguns servidores *web*, devido à legislação vigente em alguns países que restringe a sua exportação;
- Integração com Perl: um dos principais módulos do Apache é responsável pela interpretação de *scripts* CGI em Perl (*mod-perl*), de maneira a otimizar a sua execução e tempo de resposta.

5.2.3 Operações CNM e Níveis Funcionais

Foram implementadas diversas operações de gerenciamento CNM. Conforme visto na especificação de requisitos, nem todos os usuários tem acesso a todas as operações do sistema: o nível funcional associado aos usuários determina quais operações do sistema eles podem ter acesso.

Foram definidos no sistema dois níveis funcionais:

a) Básico: destinado a usuários com poucos conhecimentos de ATM; as funções do sistema mascaram terminologias e parâmetros específicos do ATM e permitem apenas a

monitoração dos elementos de rede. Um usuário com nível funcional básico pode visualizar apenas alguns dados e ter acesso às seguintes operações:

- Visualizar configuração de enlaces e *switches* (apenas as informações genéricas dos grupos *system* e interfaces da MIB-II);
- Gerar relatórios;
- Visualizar gráficos de dados coletados periodicamente;
- Visualizar os alarmes definidos;
- Visualizar *logs*.

b) Avançado: destinado a usuários com conhecimento de ATM. Tais usuários podem se beneficiar do acesso a informações particulares da tecnologia, tais como a tabela de comutação dos *switches*. Um usuário com nível funcional avançado pode, além das operações do usuário básico:

- Definir coletas periódicas;
- Definir alarmes;
- Coletar os valores imediatos de variáveis para uma checagem rápida;
- Visualizar tabelas de comutação dos *switches* e informações sobre os VPLs e VCLs.

Em relação aos alarmes do sistema, foram definidos dois tipos de notificações, podendo ser enviados um *e-mail* e/ou uma *trap* SNMP quando da ocorrência de um alarme definido.

5.2.4 Ambiente de Desenvolvimento

A implementação do sistema foi feita em um ambiente composto pelos seguintes softwares:

- Red Hat Linux 7.0;
- Perl 5.6.0;

- Apache 1.3.12.

Como o escopo do trabalho é o gerenciamento de redes ATM, foram utilizadas algumas ferramentas disponíveis na Internet, de maneira a concentrar o desenvolvimento nas características de gerenciamento de redes ATM e nas funções de um gerenciamento CNM:

- RRDTool 1.0.28;
- Módulos de segurança para Apache.

5.2.4.1 RRDTool

O RRDTool **[44]** é um utilitário desenvolvido em Perl e C que armazena, de forma consolidada, dados coletados periodicamente, bem como permite a geração de gráficos e relatórios a partir destes dados. Este utilitário é empregado junto ao componente periódico, para o armazenamento dos dados coletados e, junto aos componentes CGI, para a geração dos gráficos.

5.2.4.2 Módulos de Segurança para Apache

Para atender aos requisitos de segurança do agente CNM, devem ser utilizados os seguintes módulos para o servidor Apache:

- Mod-SSL: através deste módulo, o servidor Apache pode utilizar o protocolo SSL, oferecendo o serviço de HTTPS. São garantidos os requisitos de confidencialidade exigidos pelo serviço CNM;
- Autenticação: o Apache permite a utilização de diferentes mecanismos de autenticação. Dentro do escopo deste trabalho, é utilizado um módulo de autenticação simples, baseado na combinação de usuários/senhas.

A utilização de mecanismos de segurança integrados ao servidor *web* permite uma fácil expansão dos serviços de segurança de autenticação e confidencialidade (lembrando que tais serviços são de responsabilidade do servidor *web* na arquitetura proposta). Como a

implementação destes serviços de segurança é independente dos *scripts* CGI responsáveis pelas funções de gerenciamento, estes módulos de segurança podem ser trocados ou complementados sem envolver alterações significativas nos *scripts* CGI. Como exemplo, pode-se facilmente utilizar mecanismos de autenticação baseados em certificados sem envolver alterações no resto da solução, uma vez que o servidor *web* passa para o *script* CGI apenas a informação do nome do usuário no sistema, garantindo que ele foi corretamente autenticado.

5.2.5 Detalhes de implementação

A implementação seguiu um paradigma *bottom-up;* a seqüência de implementação é descrita abaixo:

a) Implementação das bibliotecas: inicialmente foram implementadas as bibliotecas especificadas em 5.1.4. Cada biblioteca foi implementada e testada separadamente, verificando-se as funcionalidades propostas;

b) Componente periódico: a partir das bibliotecas implementadas no item anterior, foi implementado o componente periódico. Para os testes do componente periódico, a biblioteca de interface com a plataforma de gerenciamento da rede foi alterada para ler os dados de um arquivo texto (contendo valores coletados manualmente de elementos da rede). Foi especificado um intervalo de 5 minutos entre as coletas;

c) Interface do Gerente CNM: a interface HTML do gerente do sistema foi a seguir desenvolvida, utilizando-se o utilitário Allaire HomeSite 4.0. Através desta ferramenta, foi feito um desenvolvimento visual desta interface para, a seguir, efetuar os *links* entre as páginas e nomear os *scripts* CGI a serem executados. O código HTML das páginas foi comentado de maneira que pudesse facilmente ser incorporado nos scripts CGI responsáveis por sua geração;

d) *Scripts* CGI: foram implementados os *scripts* que executam as funcionalidades dos componentes CGI da arquitetura. Estes *scripts* utilizam as bibliotecas implementadas em (a), incorporando as interfaces HTML desenvolvidas em (c);

e) Comunicação com a plataforma de gerenciamento: por fim, a biblioteca de comunicação com a plataforma de gerenciamento foi alterada, para ler os dados diretamente da plataforma de gerenciamento escolhida para os testes.

5.2.5.1 Configuração do Sistema

Há um arquivo que define as configurações principais do sistema. Há três tipos de configurações presentes:

- Interfaces Externas: estes parâmetros indicam qual o tipo de base de dados e do NMS utilizados pelo sistema, de maneira que se possa mapear as operações genéricas de comunicação para as funções específicas das aplicações externas;
- Localização de arquivos: indicam os diretórios onde se encontram os diversos arquivos do sistema;
- Operações: cada operação do sistema possui alguns parâmetros associados, tais como: o nível funcional exigido para a execução da operação e categoria e mensagem da operação para registro nos *logs* do sistema.

5.2.5.2 Bases de Dados

As bases de dados do sistema foram definidas de acordo com o especificado em **4.3.3.3**, sendo utilizada a mesma nomenclatura definida neste item. As bases de dados implementadas contém as seguintes informações:

- Vértices: contém uma descrição de cada elemento da rede (para visualização no gerente CNM) e o endereço IP a ser utilizado na monitoração, bem como um índice para cada vértice;

- Arestas: contém os índices dos vértices que formam os enlaces (os elementos de rede nas pontas dos enlaces, a interface a ser usada para a monitoração do enlace e o índice de cada enlace na base);
- Usuários: contém o nome do usuário no sistema, o índice funcional associado ao usuário, uma descrição do usuário (por exemplo, o nome completo do usuário para visualização no gerente CNM), bem como o índice da instituição à qual o usuário pertence. O nome do usuário no sistema é utilizado como índice de busca na base;
- Instituições: contém a sigla e a descrição das instituições usuárias da rede ATM, bem como a visão da rede lógica associada à instituição. A visão da rede lógica é implementada através da lista de vértices e arestas às quais os membros da instituição podem ter acesso através do sistema, conforme relacionamento visto na Figura 4-4. A sigla da instituição é utilizada como índice de busca na tabela.

Optou-se por uma implementação simples em relação à base de dados, utilizando-se arquivos-texto, com linhas representando os registros e caracteres separando os campos de cada registro. Contudo, a implementação da base de dados é isolada do resto do sistema, sendo estes arquivos acessados através de uma biblioteca em Perl, o que permite diferentes implementações de bases de dados sem alterações no resto do sistema.

Não foi implementada nenhuma ferramenta para cadastro das informações nestas bases de dados, devendo ele ser feito manualmente.

5.2.5.3 *Logs* do sistema

Os *logs* do sistema foram implementados através de arquivos texto simples, com linhas representando os registros, havendo um arquivo separado para cada instituição.

5.3. Implantação e Testes

De maneira a testar o agente em um ambiente real, ele foi implementado dentro do âmbito da Rede Metropolitana de Alta Velocidade de São Paulo (RMAV-SP), formada por um *backbone*

ATM interligando diversas instituições de pesquisa, tais como LARC-USP, CCE-USP, FAPESP, PUC-SP, INCOR, UNIFESP-EPM e contando com o apoio da Telefonica e Net-SP como provedores do meio físico **[15]**.

Há, na RMAV-SP, a necessidade de um gerenciamento CNM, uma vez que diversas instituições encontram-se conectadas a um *backbone* ATM, gerenciado por uma única instituição (atualmente o LARC-USP). Como este *backbone* é compartilhado por diversas instituições, é necessário fornecer aos integrantes do consórcio informações relativas ao seu gerenciamento. Não há, no entanto, nenhuma ferramenta na plataforma de gerenciamento adotada na RMAV-SP (Tivoli Netview) que possa atender aos requisitos de um sistema de gerenciamento CNM.

Para oferecer um gerenciamento CNM aos integrantes do consórcio RMAV-SP, é necessário que o agente criado seja capaz de interagir com a plataforma Netview, utilizando as ferramentas da plataforma para complementar as funcionalidades necessárias ao gerenciamento.

Na RMAV-SP, é utilizado o protocolo SNMP para a comunicação entre a estação de gerenciamento e os elementos da rede. O serviço de *Classical IP over ATM* (CLIP) é utilizado para o encapsulamento das primitivas SNMP em células ATM, sendo utilizados SVCs (*Switched Virtual Circuits*) na comunicação entre o gerente SNMP e os agentes presentes nos dispositivos gerenciados.

5.3.1 Arquitetura de Teste

A topologia da rede ATM utilizada para os testes pode ser vista na Figura 5-7.

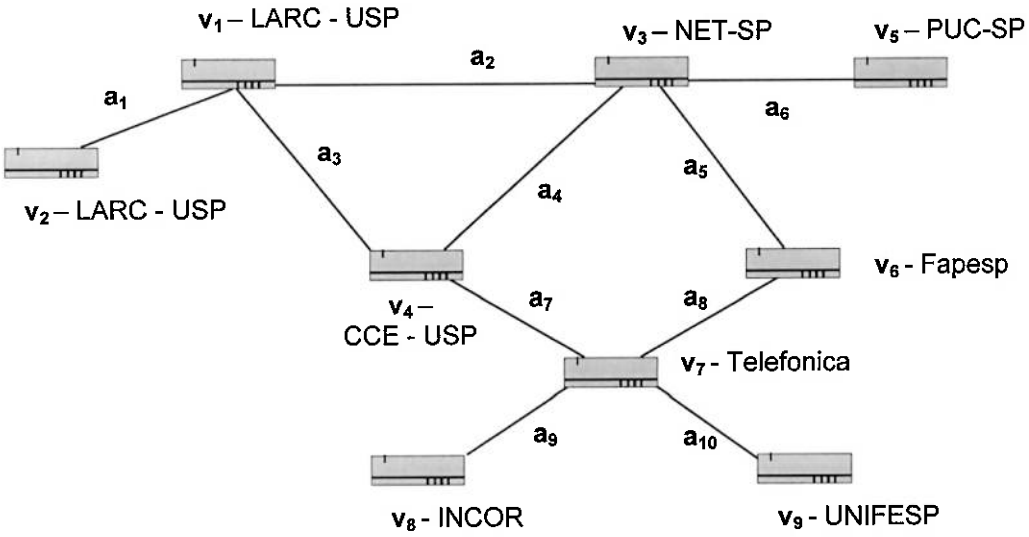


Figura 5-7: Topologia da RMAV-SP

Utilizando a nomenclatura definida em **4.3.3.3**, a Figura 5-7 representa a rede física R = (V, A) a ser particionada, onde

V = { v_k | k = 1, ... , 9 } = conjunto dos *switches* de R

A = { a_i | i = 1, ..., 10 } = conjunto dos enlaces de R

O particionamento lógico da rede foi determinado em função dos participantes do consórcio, conforme mostra a Tabela 5-14.

Instituição	**Vértices da Instituição**	**Arestas da Instituição**
LARC-USP	v_1 , v_2	a_1 , a_2 , a_3
CCE-USP	v_4	a_3 , a_4 , a_7
INCOR	v_8	A_9
UNIFESP	v_9	a_{10}
PUC-SP	v_5	A_6
FAPESP	v_6	a_5 , a_8
Telefonica	v_7	a_7 , a_8 , a_9, a_{10}
NET-SP	v_3	a_2 , a_4, a_5, a_6

Tabela 5-14: Particionamento lógico da RMAV-SP

O particionamento definido na Tabela 5-14 foi representado internamente no sistema através das base de dado do perfil dos usuários, dentro da tabela de instituições (conforme definições em **4.3.3.3** e 5.2.5.2).

Para os testes foram escolhidas duas instituições distintas e foram definidos dois usuários para cada instituição testada, sendo um com um nível funcional básico definido em seu perfil de acesso e outro com nível funcional avançado.

5.3.2 Testes Executados

O sistema foi testado segundo as seguintes funções:

- Verificação do registro no arquivo de *log* de todas as operações executadas;
- Busca de entradas no arquivo de *log*, utilizando parâmetros variados, tais como data/hora, instituição, usuário, categoria ou código de identificação do evento;
- Tentativa de acessar *logs* de outras instituições;
- Definição de alarmes para cada um dos tipos de notificação permitidos pelo sistema e para pelo menos duas variáveis distintas, em elementos de rede diferentes. Tentativa de definição de alarmes para elementos de rede não associados ao perfil do usuário;
- Exibição de alarmes definidos;
- Geração de eventos na rede que causam o disparo dos alarmes definidos no item anterior e verificação das funcionalidades de detecção e notificação dos eventos;
- Definição de coleta periódica de estatísticas da rede. Devem ser coletados valores de variáveis distintas de vários elementos da rede. Tentativa de definição de coletas para variáveis de elementos de rede não associados ao perfil do usuário;
- Exibição de coletas definidas;

- Geração de gráficos com os valores coletados periodicamente, para verificar se a coleta está sendo executada. Geração de eventos na rede que causem alterações no gráfico;
- Visualização da configuração de um enlace da rede. Tentativa de visualização da configuração de um enlace não permitido pelo perfil do usuário;
- Visualização da configuração de um *switch* da rede. Tentativa de visualização da configuração de um *switch* não permitido pelo perfil do usuário.

Com estes testes foi possível verificar se o sistema está executando corretamente o particionamento da rede de acordo com a visão lógica definida para cada usuário/instituição e se o sistema restringe corretamente o acesso dos usuários aos elementos de rede definidos pela respectiva visão lógica. Além disso, como nem todas as operações descritas podem ser executadas pelo usuário básico, os testes serviram, também, para testar o controle de acesso do sistema às operações definidas pelo nível funcional dos usuários.

5.3.3 Resultados

Com essa implementação foi possível validar a arquitetura proposta em relação aos requisitos descritos no capítulo 4. A arquitetura implementada atende aos requisitos gerais apresentados em **4.1**, fornecendo níveis funcionais distintos para os usuários, interface gráfica de fácil uso para pessoas acostumadas a navegar em páginas HTML, bem como mecanismos para o controle de acesso dos usuários às informações e às operações do sistema, de acordo com a configuração dos níveis funcionais e o particionamento das informações da rede ATM.

Para a implementação do particionamento da rede ATM foi necessário mapear a topologia da rede, descrita em forma de um grafo não-dirigido, para um formato apropriado para o armazenamento em uma base de dados relacional. O método utilizado para este mapeamento pode ser visto em 5.2.5.2 e 5.3.1.

Os requisitos da arquitetura, definidos em **4.3**, são atendidos pela arquitetura definida neste capítulo, bem como pelos módulos e bibliotecas definidos em 5.1 e 5.1.4, respectivamente. Os benefícios trazidos pela modularidade da arquitetura em relação à comunicação com a plataforma de gerenciamento foi validada no próprio decorrer do trabalho: para o desenvolvimento e teste inicial das bibliotecas do sistema foram utilizados arquivos-texto com dados coletados da plataforma de gerenciamento, que foram substituídos por funções específicas de comunicação com a plataforma Netview para o teste final com a plataforma de gerenciamento da RMAV-SP, não sendo necessário alterar as outras bibliotecas do sistema.

Para atender aos requisitos funcionais definidos em **4.2**, foi definido um conjunto de bibliotecas de funções. As funcionalidades do sistema foram implementadas utilizando exclusivamente estas bibliotecas: os *scripts* CGIs foram responsáveis apenas pelas chamadas destas funções e pela formatação dos resultados obtidos em páginas HTML (formulários, gráficos e/ou páginas simples), enquanto que o componente funcional apenas controla a ordem de chamada das funções. Com isto, foi possível verificar que as funções especificadas permitem a definição de praticamente todas as funcionalidades da arquitetura descritas em **4.2**, com exceção das seguintes funcionalidades:

- Testes de Conectividade: como as bibliotecas utilizam exclusivamente objetos de MIBs padronizadas, não foi possível executar testes de conectividade, que dependem de extensões proprietárias das MIBs dos fabricantes ou da utilização conjunta do protocolo ICMP (*Internet Control Message Protocol*) no sistema;
- Configuração e estatísticas de serviços ATM: optou-se por implementar somente as funcionalidades relativas a *switches* e enlaces ATM, devido a restrições de tempo e do ambiente utilizado para os testes;
- Estatísticas em tempo real: optou-se por implementar somente a coleta periódica de estatísticas, para não sobrecarregar a rede utilizada para testes;

- Grafos Topológicos: optou-se por não implementar mecanismos para a geração de grafos topológicos. Tal implementação requer a utilização de mecanismos para o processamento gráfico junto ao gerente CNM (através de *applets* Java) e, conforme definido em 5.1.2, optou-se por uma implementação inicial que não dependesse de particularidades de implementação de *browsers*.

Em relação à segurança, foi possível controlar o acesso às informações apresentadas pelo sistema, de acordo com o particionamento da rede definido. Além disso, utilizando o protocolo SNMP dentro da rede ATM e o protocolo HTTPS entre o gerente e o agente CNM, foi possível suplantar os problemas de segurança inerentes ao protocolo SNMP (de confidencialidade e integridade dos dados trafegados) na porção externa do *backbone* ATM (o HTTPS possui mecanismos para garantir a confidencialidade e integridade dos dados transmitidos).

A princípio, a utilização do SNMP internamente pode parecer insegura, uma vez que é possível para uma estação ligada a rede ATM fazer-se passar pela estação de gerenciamento (não está previsto no padrão do CLIP ou LANE (*LAN Emulation*) nenhum mecanismo que controle qual endereço IP uma estação pode utilizar). No entanto, pode-se fazer duas observações a este respeito:

- A porção mais crítica em termos de segurança, do ponto de vista dos usuários do serviço CNM, é entre os gerentes e o agente CNM, pois internamente assume-se que a segurança é de responsabilidade do provedor do serviço. Dentro da arquitetura proposta, a utilização de HTTPS entre o agente e o gerente soluciona este problema;
- As características da tecnologia ATM podem ser utilizadas para melhorar a segurança interna: através da configuração de PVCs (*Permanent Virtual Circuits*) e SPVCs (*Semi-Permanent Virtual Circuits*) entre os elementos da rede e o gerente, restringe-se a possibilidade de uma estação qualquer emular a estação de gerenciamento pois, para tanto, necessitaria obter o mesmo endereço ATM e reconfigurar os PVCs e/ou SPVCs nos elementos da rede, além de necessitar do mesmo endereço IP da estação de gerenciamento.

5.4. Considerações Finais

Conforme visto anteriormente, a solução proposta atende aos requisitos definidos no capítulo anterior. Para um melhor entendimento de como estes requisitos são atendidos, eles são resumidos em três tabelas e relacionados com a solução proposta: a Tabela 5-15 apresenta os requisitos básicos de uma solução para um serviço CNM, a Tabela 5-16 apresenta os requisitos funcionais e a Tabela 5-17 apresenta os requisitos da arquitetura da solução.

Particionamento das Informações	A arquitetura proposta inclui uma base de dados para o armazenamento das visões lógicas da rede, de acordo com o particionamento definido em 4.3.3.3. A biblioteca de perfil do usuário indica para o sistema qual a visão lógica associada à instituição do usuário.
Definição de Níveis Funcionais	As funções das bibliotecas de controle de acesso e perfil do usuário reconhecem e tratam o nível funcional associado ao usuário.
Interface Customizada	O componente CGI só exibe os elementos da rede que pertencem à visão lógica associada à entidade.
Controle de Operações	O usuário só pode executar as funções definidas no sistema; além disso, a função *acesso_permitido* (biblioteca de controle de acesso) leva em consideração o nível funcional do usuário para autorizar as operações.
Facilidade de Uso	A interface baseada em HTML visa facilitar o uso do sistema.

Tabela 5-15: Resumo dos requisitos básicos atendidos pela solução proposta

Requisitos Gerais	
Log de Transações	A função *cnm_cgi* registra cada operação no log.
Definição de alarmes	A função *definir_alarme* é responsável pela definição de alarmes no sistema.
Relatórios e gráficos de dados históricos	As funções *relatório* e *gráfico* utilizam dados armazenados, coletados pelo componente periódico.
Estatísticas em tempo real	A arquitetura prevê apenas a coleta periódica, com um intervalo mínimo de tempo. Para a implementação, foi utilizado um valor de 5 minutos como mínimo entre as coletas.
Informações de configuração	A função *ver_config_equip* permite a visualização de dados de configuração de *switches* ATM.
Gerenciamento de Falhas	
Monitoração e alarmes para taxas de erro	A arquitetura não possui bibliotecas com funções específicas de monitoração e de alarmes para taxas de erro; tais funcionalidades são implementadas através da coleta de variáveis da MIB SNMP e definição do alarme correspondente, usando as funções genéricas definidas na arquitetura.
Histórico de *Downtime*	A arquitetura não possui bibliotecas com funções específicas para o levantamento deste histórico; ele pode ser implementado através da definição de uma coleta periódica para a variável da MIB SNMP *sysUptime* (para *switches*) ou *ifOperStatus* (para enlaces).
Testes de Conectividade	A arquitetura não possui mecanismos para testes de conectividade.
Gerenciamento de Configuração	
Circuitos Virtuais	A função *ver_VC* permite a visualização dos circuitos virtuais em uso em um dado *switch*.
Configuração dos *Switches*	A função *ver_config_equip* permite a visualização da configuração de um dado *switch*.
Configuração dos serviços ATM	A arquitetura não possui funções para a visualização da configuração de serviços ATM.
Topologia da rede	A arquitetura não prevê mecanismos para a visualização gráfica da topologia da rede ATM.
Gerenciamento de Desempenho	
Relatórios e Gráficos de Funcionamento	A coleta de dados periódica e as funções *relatorio* e *grafico* permitem a geração de gráficos e relatórios com estatísticas de enlaces (não foi implementada a monitoração das estatísticas de serviços ATM)
Definição de Limites de Desempenho	As funções da biblioteca de alarmes permite a definição de um limite inferior e superior para os valores coletados, definindo intervalos de funcionamento para os valores das variáveis coletadas.

Tabela 5-16: Resumo dos requisitos funcionais atendidos pela solução proposta

Requisitos gerais da arquitetura CNM	
Portabilidade	A arquitetura proposta baseia-se em componentes e características independentes de plataforma, tais como: ambiente *web*, variáveis da MIB SNMP e bibliotecas de funções portáveis. A implementação utilizou *softwares* com versões para as plataformas citadas, tais como o servidor Apache e a linguagem Perl.
Disponibilidade	A arquitetura prevê indiretamente a questão da disponibilidade: em um servidor com mais de uma interface de rede, o servidor *web* atende indistintamente requisições vindas das várias interfaces. Se estas interfaces estiverem ligadas a redes distintas, o requisito de disponibilidade é atendido.
Integração com Plataforma de Gerenciamento	As funções *trap_snmp* e *ler_dados* provêm uma interface genérica para as particularidades das plataformas de gerenciamento. Na implementação estas funções foram mapeadas para funções que permitiam a comunicação com a plataforma Netview e com a leitura de dados a partir de arquivos-texto.
Requisitos de Segurança	
Autenticação	A arquitetura atende a este requisito, utilizando os mecanismos de autenticação existentes no servidor e *browser web*.
Controle de Acesso	A arquitetura define uma biblioteca para o controle de acesso ao sistema, bem como indica como deve ser a estrutura do componente CGI para se ter um acesso controlado.
Confidencialidade e Integridade	A arquitetura utiliza o protocolo HTTPS para atender a estes requisitos.
Modularidade do Sistema	
Autenticação	As funcionalidades do módulo de autenticação definido nos requisitos da arquitetura são atendidos pelo servidor *web*.
Controle de Acesso	A arquitetura define uma biblioteca específica para o controle de acesso.
Perfil dos Usuários do Sistema	A arquitetura define uma biblioteca específica para o gerenciamento do perfil do usuário.
Serviços CNM	A arquitetura não define um módulo específico para os serviços CNM. As funcionalidades deste módulo são atendidas por funções específicas, pertencentes a diferentes bibliotecas, e utilizadas pelos componentes funcionais (conforme pode ser visto em 5.1.5).
Comunicação com Plataforma de Gerenciamento	A arquitetura define uma biblioteca específica para a comunicação com a plataforma de gerenciamento.
Gerente CNM	A utilização do *browser* do lado do gerente atende aos requisitos de universalidade de acesso, portabilidade, facilidade de uso e comunicação com o agente CNM.

Tabela 5-17: Resumo dos requisitos de uma arquitetura atendidos pela solução proposta

6. Considerações Finais

6.1. Avaliação Crítica

Conforme visto no capítulo 1, o principal objetivo proposto deste trabalho foi a definição de uma arquitetura capaz de oferecer um serviço de gerenciamento CNM (*Customer Network Management*) de um *backbone* ATM (*Asynchronous Transfer Mode*) a instituições usuárias deste *backbone*, que oferecesse os seguintes benefícios:

- Possibilitar aos administradores destas instituições o acesso a interfaces e ferramentas de gerenciamento ATM, evitando que haja sistemas complexos de gerenciamento ATM em suas redes privativas;
- Diminuir os custos de gerenciamento para as instituições, evitando que eles necessitem de novas ferramentas ou plataformas para o gerenciamento ATM;
- Evitar ou diminuir a necessidade de treinamento para que os administradores das redes privativas destas instituições possam gerenciar a sua parte no *backbone*;
- Prover uma infra-estrutura escalar para o serviço, capaz de agregar novas funcionalidades quando necessário;
- Controlar o que cada administrador pode fazer em termos de gerenciamento, evitando que uma operação de um administrador interfira no comportamento global do *backbone*.

Os três primeiros objetivos foram atendidos através de uma arquitetura via *web*, uma vez que o acesso às informações de gerenciamento da rede ATM é feito sem necessitar que as instituições usuárias da rede ATM possuam estas ferramentas. O seu uso é facilitado pela interface homem-máquina baseada em HTML (*Hypertext Markup Language*), familiar a usuários pela similaridade com a navegação em páginas HTML na Internet.

Através da arquitetura e das funções definidas no capítulo 5, baseadas na especificação de requisitos apresentada no capítulo 4, os dois últimos objetivos são atendidos. A estruturação modular da arquitetura proposta e o isolamento de particularidades de implementação em bibliotecas de funções permite facilmente agregar ou modificar funcionalidades sem a necessidade de alterações profundas em todo o sistema. Através da definição do perfil dos usuários, contendo quais operações cada usuário pode executar (ou seja, o nível funcional associado ao usuário) e quais elementos cada usuário pode monitorar (a rede lógica associada à instituição do usuário), há um controle sobre as operações que podem impactar o funcionamento da rede. Além disso, concentrando a coleta das informações no componente periódico, o sistema pode ordenar e minimizar as interações com a plataforma de gerenciamento, minimizando a possibilidade de impacto no desempenho da rede devido a excesso de tráfego de gerenciamento.

Baseado na arquitetura proposta, foi implementado um protótipo do sistema, sendo possível verificar o seu comportamento em uma rede de produção. Através do protótipo, foi possível monitorar estatísticas da rede ATM e receber alarmes, a partir de eventos definidos no sistema. O protótipo foi implementado e implantado em sistemas operacionais diferentes (Linux e AIX, respectivamente), utilizando plataformas de gerenciamento distintas (dados lidos de arquivos-texto com as informações coletadas e dados lidos diretamente da rede ATM, através de plataforma Netview).

Um outro resultado obtido com este trabalho foram as representações das visões lógicas de uma rede ATM. Através da teoria de grafos, foi definido no trabalho uma maneira de se representar a topologia das visões lógicas intuitivamente, a partir da topologia da rede. Esta representação em grafos é convertida para uma representação mais apropriada para o processamento do sistema através de recursos de uma base de dados relacional.

Como resultados complementares do trabalho, há ainda o levantamento dos padrões SNMP para gerenciamento de redes ATM.

6.2. *Trabalhos Futuros*

A partir deste trabalho podem ser desenvolvidos diversos tópicos, estendendo as funcionalidades da solução proposta.

Em relação à interface homem-máquina, pode-se fazer um estudo aprofundado em relação à facilidade de uso da interface, utilizando metodologias e métricas de usabilidade para avaliar a adequação da interface ao uso proposto e, se necessário, definir uma interface homem-máquina mais apropriada. A arquitetura modular do sistema isola as particularidades da interface do funcionamento do agente, permitindo a alteração da interface sem necessitar de alterações significativas no agente. Além disso, como a interface homem-máquina do sistema foi definida em HTML, alterações na interface podem ser facilmente feitas sem exigir grandes conhecimentos de linguagens de programação.

O trabalho utilizou exclusivamente características comuns a equipamentos de rede ATM, conforme os padrões existentes. Pode-se complementar o sistema de gerenciamento, acrescentando novas funcionalidades, tais como:

- Informações das MIBs (*Management Information Bases*) privativas dos diversos dispositivos ATM (adicionando-se ao sistema "*drivers*" para acesso a estas MIBs privativas);
- Modelagem dos serviços ATM para o gerenciamento CNM;
- Modelagem do perfil dos usuários em um sistema de diretórios, tais como o X.500 ou LDAP (*Lightweight Directory Access Protocol*) permitindo, por exemplo, a priorização do tráfego de gerenciamento baseado no perfil do usuário (utilizando-se equipamentos de rede que suportem a especificação DEN – *Directory-Enabled Networking*) ou integração com outras ferramentas que suportem LDAP.
- Funcionalidades para o oferecimento de serviço CNM classe II, ou seja, que permita não só a monitoração da rede como também alterar algumas de suas funções de forma controlada.

7. Bibliografia

[1] ABUSAMRA, J. ATM Net Management: Missing Pieces. **Data Communications**, Maio, 1998. Disponível em http://www.data.com/tutorials/missing.html. Acessado em 01/08/2000.

Esta artigo apresenta uma visão geral dos desafios que envolvem o gerenciamento de redes ATM.

[2] AHMED, M., TESINK, K. **RFC 1695 - Definitions of Managed Objects for ATM Management Version 8.0 using SMIv2**. August 1994. Disponível em http://www.ietf.org/rfc/rfc1695.txt. Acessado em 22/12/2000.

Define a 1ª especificação de uma MIB para o gerenciamento de redes ATM, conhecida como AToM MIB. Apesar de não estar de acordo com as normas ATM mais recentes, a maioria da implementação dos agentes SNMP em *switches* ATM utilizam-se desta especificação.

[3] ALEXANDER, P., CARPENTER, K. ATM Net Management: A Status Report. **Data Communications**, Setembro, 1995. Disponível em http://www.data.com/tutorials/atm_net_management.html. Acessado em 01/08/2000.

Esta artigo apresenta uma visão geral dos desafios que envolvem o gerenciamento de redes ATM, detalhando a estratégia para o gerenciamento de redes ATM especificada pelo ATM Forum (arquitetura de gerenciamento e células específicas para o gerenciamento)

[4] ATM FORUM. **ATM Forum Traffic Management Specification, Version 4.0.** June 1996. Disponível em ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000. Acessado em 22/12/2000.

Norma que define os tipos de tráfego que podem ser definidos em um circuito virtual ATM. Define também os parâmetros de qualidade de serviço relacionados a estes tráfegos.

[5] ATM FORUM. **ATM User-Network Interface, Version 3.1 (UNI 3.1) Specification.** 1994. Disponível em ftp://ftp.atmforum.com/pub/approved-specs/af-uni-0010.002. Acessado em 22/12/2000.

Definição das normas para a sinalização entre um elemento de interconexão ATM e um dispositivo terminal. Define os parâmetros de definição de tipos de tráfego

[6] ATM FORUM. **Customer Network Management (CNM) for ATM public network service (M3 specification).** October, 1994. Disponível em ftp://ftp.atmforum.com/pub approved-specs/af-nm-0019.000. Acessado em 22/12/2000.

Definição da arquitetura de gerenciamento ATM definida pelo ATM Forum, bem como a definição da arquitetura para o gerenciamento CNM.

[7] ATM FORUM. **Integrated Local Management Interface (ILMI) Specification, Version 4.0.** September, 1996. Disponível em ftp: ftp.atmforum.com/pub/approved-specs/af-ilmi-0065.000. Acessado em 22/12/2000.

Definição da norma para a sinalização de gerenciamento entre elementos de rede diretamente conectados.

[8] ATM FORUM. **Private Network-Network Interface Specification, Version 1.0.** March, 1996. Disponível em ftp: ftp.atmforum.com/pub approved-specs af-pnni-0055.000. Acessado em 22/12/2000.

Definição das normas de sinalização e roteamento para a definição de rotas entre elementos de interconexão ATM. Define também a MIB necessárias para o gerenciamento de redes que se utilizam de PNNI.

[9] ATM FORUM. **Private Network-Network Interface Specification Version 1.0 Addendum (Soft PVC MIB)** September, 1996. Disponível em ftp://ftp.atmforum.com/pub/approved-specs/af-pnni-0066.000. Acessado em 22/12/2000.

Definição da MIB para o gerenciamento de conexões do tipo SPVC.

[10] ATM FORUM. **LAN Emulation – Client Management Specification, Version 1.0.** September, 1995. Disponível em ftp://ftp.atmforum.com/pub/approved-specs/af-lane-0038.000. Acessado em 22/12/2000.

[11] ATM FORUM. **LAN Emulation – Server Management Specification, Version 1.0.** March, 1996. Disponível em ftp://ftp.atmforum.com/pub/approved-specs/af-lane-0057.000. Acessado em 22/12/2000.

Normas que definem as MIBs para o gerenciamento do serviço de LAN Emulation over ATM.

[12] ATM FORUM. **Remote Monitoring MIB Extensions for ATM Network, Version 1.0.** May 1997. Disponível em ftp://ftp.atmforum.com/pub/approved-specs/af-nm-test-0080.000. Acessado em 22/12/2000.

Contém a definição de uma MIB análoga à MIB RMON (específica de redes Ethernet e Token Ring), porém definida para coleta de estatísticas em redes ATM.

[13] BERNERS-LEE, T., et. al. **RFC 2616 – Hypertext Transfer Protocol - HTTP/1.1.** June 1999. Disponível em http://www.ietf.org/rfc/rfc2616.txt. Acessado em 22/12/2000.

Este documento contém a especificação principal do protocolo HTTP, contendo formato das mensagens, códigos de erro, bem como considerações sobre segurança e *caching* relacionadas ao protocolo.

[14] CARVALHO, T.C.M.B., REDIGOLO, F.F. **Apostilas do curso de gerenciamento de redes do LARC.** Setembro, 2000.

Documentação do curso de gerenciamento de redes ministrado pelo LARC, cobrindo, entre outros tópicos. gerenciamento SNMP e gerenciamento via web.

[15] CARVALHO, T.C.M.B., RUGGIERO, W.V., REDIGOLO, F.F., et. al. RMAV-SP: Implantação e Utilização da Internet 2 de São Paulo. **II Workshop RNP2.** Belo Horizonte: Setembro, 2000.

Artigo que descreve a experiência prática na implantação do *backbone* da Internet 2 em São Paulo, composto pela rede metropolitana RMAV-SP. Cobre desde a implementação física da rede até a implementação dos serviços de Tele-educação, Tele-medicina e gerenciamento da rede.

[16] CASE, J., FEDOR, M., SCHOFFSTALL, M., DAVIN, J. **RFC 1157 - Simple Network Management Protocol.** May 1990. Disponível em http://www.ietf.org/rfc/rfc1157.txt. Acessado em 22/12/2000.

Define a arquitetura SNMP e seus principais elementos, assim como define o protocolo de gerenciamento SNMP (suas primitivas, protocolo de transporte utilizado, entre outros)

[17] CASE, J., McCLOGHRIE, K., ROSE, M., WALDBUSSER, S. **RFC 1901 - Introduction to Community-based SNMPv2.** January 1996. Disponível em http://www.ietf.org/rfc/rfc1901.txt. Acessado em 22/12/2000.

Contém a descrição geral da arquitetura SNMPv2, de acordo com a revisão publicada no início de 1996. Descreve, de forma suscinta, as alterações entre esta versão e a versão anterior da arquitetura (SNMPv1).

[18] CASE, J., McCLOGHRIE, K., ROSE, M., WALDBUSSER, S. **RFC 2578 – Structure of Management Information Version 2.** April, 1999. Disponível em http://www.ietf.org/rfc/rfc2578.txt. Acessado em 22/12/2000.

Define a 2ª versão da SMI, definindo as regras para especificação de MIBs. Contém melhorias em relação às RFCs 1157 e 1212, como a eliminação de diversas ambigüidades na especificação anterior e a definição de novos tipos de dados, sendo capaz de suportar especificações de MIBs mais elaboradas e completas. As MIBs para o gerenciamento de redes ATM são especificadas de acordo com as regras desta e das RFCs 2579 e 2580.

[19] CASE, J., McCLOGHRIE, K., ROSE, M., WALDBUSSER, S. **RFC 2579 - Textual Conventions for SMIv2.** April, 1999. Disponível em http://www.ietf.org/rfc/rfc2579.txt. Acessado em 22/12/2000.

Na especificação de uma MIB, é interessante poder-se definir novos tipos de dados, baseados nos tipos definidos na SMI. Como benefícios obtêm-se especificações mais fáceis de serem compreendidas, bem como mais concisas, uma vez que pode-se definir estes tipos uma única vez e utilizá-los em diversas MIBs. Este documento especifica como podem ser definidos estes tipos de dados, denominados convenções textuais (textual conventions), bem como especifica novos tipos de dados.

[20] CASE, J., McCLOGHRIE, K., ROSE, M., WALDBUSSER, S. **RFC 2580 - Conformance Statements for SMIv2.** April, 1999. Disponível em http://www.ietf.org/rfc/rfc1904.txt. Acessado em 22/12/2000.

Na especificação de uma MIB, é interessante poder-se definir níveis diferentes para a sua implementação, permitindo uma implementação em etapas: começando com uma implementação inicial básica (implementação mínima) e adicionando-se novos objetos até uma implementação completa. Este documento especifica como devem ser definidos os níveis de implementação para uma MIB, definindo as regras para que uma determinada implementação possa estar em conformidade com um determinado nível de implementação.

[21] CHUNG, C., GREENE, M. **RFC 2417 - Definitions of Managed Objects for Multicast over UNI 3.0/3.1 based ATM Networks.** September 1998. Disponível em http://www.ietf.org/rfc/rfc2417.txt. Acessado em 22/12/2000.

Definição da MIB para o gerenciamento de redes ATM que utilizam-se do serviço de Multicast.

[22] DISTRIBUTED MANAGEMENT TASK FORCE. **Specification for CIM Operations over HTTP, version 1.0.** July 20th, 1999. Disponível em http://www.dmtf.org/download/spec/xmls/CIM_HTTP_Mapping10.htm. Acessado em 22/12/2000.

Este documento define como as operações de gerenciamento da arquitetura WBEM (operações sobre o modelo CIM) podem ser mapeadas para o protocolo HTTP, utilizando-se da linguagem XML.

[23] DISTRIBUTED MANAGEMENT TASK FORCE. **Specification for the Representation of CIM in XML, version 2.0.** July 20th, 1999. Disponível em http://www.dmtf.org/download/spec/xmls/CIM_XML_Mapping20.htm. Acessado em 22/12/2000.

Este documento define como os elementos do modelo CIM devem ser mapeados para uma representação na linguagem XML.

[24] DISTRIBUTED MANAGEMENT TASK FORCE. **XML As a Representation for Management Information – A White Paper.** September 15th, 1998. Disponível em http://www.dmtf.org/spec/xmlw.html. Acessado em 22/12/2000.

Contém uma visão geral da linguagem de *markup* XML, apresentando-a dentro do contexto de gerenciamento visado pela arquitetura WBEM.

[25] DISTRIBUTED MANAGEMENT TASK FORCE. **CIM Frequently Asked Questions.** Disponível em http://www.dmtf.org/spec/cimfaq.html. Acessado em 22/12/2000.

Este documento apresenta uma visão geral do modelo CIM e dos seus componentes, tais como os diversos CIM *schemas* e o CIM *meta-schema.*

[26] DISTRIBUTED MANAGEMENT TASK FORCE. **Common Information Model (CIM) Core Model, version 2.4.** August 30th, 2000. Disponível em http://www.dmtf.org/spec/cims.html. Acessado em 22/12/2000.

Este documento apresenta as principais classes definidas pela arquitetura WBEM para o gerenciamento de redes.

[27] DISTRIBUTED MANAGEMENT TASK FORCE. **Common Information Model (CIM) Specification, version 2.2.** June 14th, 1999. Disponível em http://www.dmtf.org/download/spec/cim_schema_v22.pdf. Acessado em 22/12/2000.

Este documento contém a especificação principal do modelo orientado a objetos CIM.

[28] FOWLER, D. **RFC 2496 - Definitions of Managed Objects for the DS3/E3 Interface Type.** January, 1999. Disponível em http://www.ietf.org/rfc/rfc2496.txt. Acessado em 22/12/2000.

Documento contendo a especificação da MIB para o gerenciamento de interfaces do tipo DS3/E3.

[29] FRIER, A., KARLTON, P., KOCHER, P. **The SSL 3.0 Protocol.** November, 1996.Disponível em http://home.netscape.com/eng/ssl3/ssl-toc.html. Acessado em 22/12/2000.

Especificação do protocolo de segurança *Secure Sockets Layer* (SSL), desenvolvido pela Netscape.

[30] GREENE, M., LUCIANI, J., WHITE, K. **RFC 2320 - Definitions of Managed Objects for Classical IP and ARP Over ATM Using SMIv2 (IPOA-MIB).** April 1998. Disponível em http://www.ietf.org/rfc/rfc2320.txt. Acessado em 22/12/2000.

Definição da MIB para o gerenciamento de redes ATM baseadas no padrão *Classical IP over ATM.*

[31] GUPTA, R. The "glue" of Networks: Looking at IP over ATM. **53 bytes: The ATM Forum Newsletter.** Vol. 7, N. 1, Fevereiro, 1999. Disponível em http://www.atmforum.com/atmforum/library/53bytes/53_2_99/53_2_99_02.html. Acessado em 22/12/2000.

Este artigo apresenta um comparativo entre as soluções de *LAN Emulation* e *Classical IP over ATM*

[32] HARRINGTON, D., PRESUHN, R., WIJNEN, B. **RFC 2271 - An Architecture for Describing SNMP Management Frameworks.** January 1998. Disponível em http://www.ietf.org/rfc/rfc2271.txt. Acessado em 22/12/2000.

Este documento apresenta uma visão geral da arquitetura do SNMP versão 3, definindo uma arquitetura modular para os gerentes e agentes SNMP.

[33] INTERNET ASSIGNED NUMBERS AUTHORITY. **SMI Network Management Private Enterprise Codes.** Disponível em http://www.isi.edu/in-notes/iana/assignments/enterprise-numbers. Acessado em 22/12/2000.

Contém a relação de todas as posições, dentro da árvore de nomeação definida pela SMI, (*iso.org.dod.internet.privates.enterprises*) para as empresas adicionarem as extensões de suas MIBs privativas.

[34] INTERNET ENGINEERING TASK FORCE. **RFC 2700 – Internet Official Protocol Standards.** August, 2000. Disponível em http://www.ietf.org/rfc/rfc1901.txt. Acessado em 22/12/2000.

Contém a relação de todas as especificações publicadas em RFCs, bem com o respectivo *status* de padronização de cada especificação dentro do processo do IETF.

[35] KASTELEIJN, W. **Web Based Management.** M.Sc. Thesis, University of Twente, NL, April, 1997. Disponível em ftp://ftp.cs.utwente.nl/pub/src/snmp/UT-THESIS/Kasteleijn.ps. Acessado em 22/12/2000.

Tese sobre gerenciamento web. Compara o gerenciamento web com o gerenciamento SNMP tradicional e define um protótipo para o gerenciamento de um *switch* ATM via web.

[36] KOVACH, S., CARVALHO, T.C.M.B. **Apostilas do curso de Tecnologia ATM do LARC.** Setembro, 2000.

Documentação do curso de tecnologia ATM, ministrado pelo LARC.

[37] LAUBACH, M., HALPERN, J. **RFC 2225 - Classical IP and ARP over ATM.** April 1998. Disponível em http://www.ietf.org/rfc/rfc2225.txt. Acessado em 22/12/2000.

Especificação atual do padrão *Classical IP over ATM*, originalmente especificado na RFC

1577

[38] MCCLOGHRIE, K., HEINANEN, J., GREENE, W., et al. **RFC 2512 – Accounting Information for ATM Networks**. February, 1999. Disponível em http://www.ietf.org/rfc/rfc2512.txt. Acessado em 22/12/2000.

Especificação da MIB para Contabilização em redes ATM.

[39] McCLOGHRIE, K., KASTENHOLZ, F. **RFC 2233 - The Interfaces Group MIB.** November 1997. Disponível em http://www.ietf.org/rfc/rfc2233.txt. Acessado em 22/12/2000.

Este documento especifica uma extensão ao grupo Interfaces da MIB-II, uma vez que este grupo apresenta uma visão bastante simplista das interfaces dos dispositivos gerenciados, sendo inadequado a tecnologias onde as interfaces são divididas em sub-camadas (como, por exemplo, os circuitos virtuais existentes em redes X. 25 e ATM, ou então enlaces PPP sob interfaces seriais). Esta MIB é necessária para o gerenciamento de redes ATM.

[40] McDYSAN, D., SPOHN, D. **ATM Theory and Applications.** McGraw-Hill, June 1999.

Este livro apresenta uma visão completa da tecnologia ATM.

[41] NCSA. **The Common Gateway Interface**. Disponível em http://hoohoo.ncsa.uiuc.edu/cgi. Acessado em 22/12/2000.

Este documento contém a especificação do padrão CGI para comunicação entre servidores web e programas externos, visando a geração de páginas em tempo real.

[42] NOTO, M., SPIEGEL, E., TESINK, K et al. **Definitions of Supplemental Managed Objects for ATM Management - Draft**. November, 2000. Disponível em http://search.ietf.org/internet-drafts/draft-ietf-atommib-atm2-14.txt. Acessado em 22/12/2000.

Este documento define objetos complementares à AtoM MIB, visando um melhor gerenciamento de circuitos SVCs e SPVCs. È um trabalho em progresso, não constituindo um padrão do IETF propriamente dito.

[43] NOTO, M., SPIEGEL, E., TESINK, K. **RFC 2514 – Definitions of Textual Conventions and OBJECT-IDENTITIES for ATM Management**. February 1999. Disponível em http://www.ietf.org/rfc/rfc2514.txt. Acessado em 22/12/2000.

Este documento define os tipos de dados e macros específicos para a definição de MIBs ATM.

[44] OETIKER, TOBI. **RRDTool Documentation**. Disponível em http://ee-staff.ethz.ch/~oetiker/webtools/rrdtool. Acessado em 22/12/2000.

Neste endereço encontra-se a documentação *online* da ferramenta RRDTool, para o armazenamento de dados coletados periodicamente e geração de gráficos em scripts CGI.

[45] PAN, H. **SNMP-based ATM network management**. Artech House, 1998.

Este livro apresenta as principais MIBs SNMP padronizadas envolvidas no gerenciamento de redes ATM. Apresenta como o SNMP pode ser utilizado para o gerenciamento das diversas características das redes ATM, utilizando-se tanto das MIBs padronizadas como de MIBs

proprietárias.

[46] REDIGOLO, F. F., CARVALHO, T.C.M.B. Analyzing Emerging Web-Based Management Standards. **Anais Eletrônicos do III Info-Net – Congreso Internacional y Exposición de Informática e Internet**. Mendoza, Argentina, Junio, 1998.

Este artigo apresenta uma análise dos padrões de gerenciamento via *web* em desenvolvimento.

[47] ROSE, M., McCLOGHRIE, K. **RFC 1212 - Concise MIB Definitions**. March 1991. Disponível em http://www.ietf.org/rfc/rfc1212.txt. Acessado em 22/12/2000.

Este documento define recomendações para novas MIBs SNMP: são definidas recomendações sobre como deve ser uma especificação de uma nova MIB SNMP, de maneira que a especificação produzida seja sucinta (concisa), eliminando-se redundância de informações.

[48] ROSE, M., McCLOGHRIE, K. **RFC 1155 - Structure and Identification of Management Information for TCP/IP-based Internets**. May 1990. Disponível em http://www.ietf.org/rfc/rfc1155.txt. Acessado em 22/12/2000.

Define as regras de estruturação que as MIBs SNMP devem seguir. São definidas regras de nomeação, sintaxe da descrição, linguagem de descrição formal, estrutura de dados dos objetos, entre outros.

[49] ROSE, M., McCLOGHRIE, K. **RFC 1213 - Management Information Base for Network Management of TCP/IP-based internets: MIB-II**. March 1991. Disponível em http://www.ietf.org/rfc/rfc1213.txt. Acessado em 22/12/2000.

Este documento define a MIB-II, que contém os objetos de gerenciamento de implementação obrigatória para um agente SNMP.

[50] SPRENKELS, R., VAN DER WAAIJ, B., VAN BEIJNUM, B.J, PRAS, A. The Feasibility of Introducing ATM SVCs. **Proceedings of the European Conference on Networks and Optical Communications 1999**, IOS Press and AKM AG, Basel, June 1999.

Este artigo apresenta uma análise de viabilidade para a utilização de SVCs na rede ATM que liga os centros de pesquisa na Holanda. Como parte da análise, são analisados os padrões do IETF para gerenciamento de conexões SVCs, incluindo padrões para contabilização de utilização.

[51] STALLINGS, W. **SNMP, SNMPv2, SNMPv3 and RMON 1 and 2**. 3rd Edition. Reading, MA: Addison-Wesley, 1998

Este livro é uma referência completa sobre a arquitetura SNMP. Ele detalha os principais elementos desta arquitetura, assim como a sua evolução (SNMPv1, SNMPv2 e SNMPv3). Apresenta também as MIBs RMON e RMON2, de significância para o gerenciamento de tráfego em segmentos de redes locais.

[52] STALLINGS, W. SNMPv3: A Security Enhancement for SNMP. **IEEE Communications Surveys.** Fourth Quarter 1998, Vol.1, No.1. Disponível em http: www.comsoc.org pubs/surveys/4q98issue/stallings.html. Acessado em 22/12/2000.

Este artigo apresenta a arquitetura do SNMPv3 e seus módulos principais, apresentando em detalhes os mecanismos de segurança definidos nesta versão da arquitetura SNMP.

[53] SUN MICROSYSTEMS. **Java Management Extensions Instrumentation and Agent Specification, v1.0.** Julho, 2000. Disponível em http://java.sun.com/products/JavaManagement. Acessado em 22/12/2000.

Contém a especificação dos elementos relacionados aos dispositivos gerenciados em uma arquitetura JMX, agente e instrumentação.

[54] SUN MICROSYSTEMS. **Java Management Extensions While Paper.** Junho, 1999. Disponível em http://java.sun.com/products/JavaManagement/wp. Acessado em 22/12/2000.

Apresenta uma visão geral da arquitetura JMX, descrevendo de forma sucinta seus principais elementos.

[55] TESINK, K. **RFC 2558 - Definitions of Managed Objects for the SONET/SDH Interface Type.** March, 1999. Disponível em http://www.ietf.org/rfc/rfc1595.txt. Acessado em 22/12/2000.

Definição da MIB para o gerenciamento de enlaces físicos de redes ATM baseados no padrão Sonet/SDH.

[56] TESINK, K. **RFC 2515 - Definitions of Managed Objects for ATM Management.** February 1999. Disponível em http://www.ietf.org/rfc/rfc2515.txt. Acessado em 22/12/2000.

Evolução da RFC 1695, este documento atualiza a AToM MIB, incluindo melhorias na área de gerenciamento de tráfego. Utiliza-se das definições da RFC 2514.

[57] TESINK, K., BRUNNER, T. (Re)configuration of ATM Virtual Connections with SNMP. **The Simple Times.** Vol. 3, n.2, Aug. 1994. http://www.simple-times.org. Acessado em 22/12/2000.

Este artigo apresenta como a AToM MIB (RFC 1695), explicando como ela pode ser utilizada para o gerenciamento de circuitos virtuais ATM.

[58] WALL, Larry, SCHWARTZ, Randall, CHRISTIANSEN, Tom. **Programming Perl.** 2nd Edition, O'Reilly & Associates.

Documentação da linguagem Perl, apresenta a gramática e os elementos da linguagem a serem utilizados na programação.

[59] WILLIANS, R. Web Based Enterprise Management - W101. **DMTF 1999 Annual Conference.** June, 1999. Disponível em http://www.dmtf.org. Acessado em 22/12/2000.

Documentação de um *workshop* do DMTF, apresenta a arquitetura WBEM e seus principais elementos.

Apêndice I – ATM

A tecnologia ATM (*Asynchronous Transfer Mode*) foi a tecnologia escolhida pelo ITU-T (International Telecommunication Union – Telecommunication Standardization Sector) para prover suporte aos serviços B-ISDN (*Broad-Integrated Services Digital Network*). As suas principais características são [36]:

- Utiliza uma técnica de comutação derivada da comutação de pacotes e da comutação de circuitos, denominada comutação de células;
- Permite a alocação dinâmica de banda de acordo com a demanda;
- Suporta aplicações multimídia, podendo combinar dados, voz e vídeo, utilizando, como meios de transmissão, enlaces de alta velocidade.
- Suporta serviços com requisitos de banda de transmissão diferenciados.
- Atende aos requisitos de vários tipos de serviços, com taxas de bit constante (CBR – *Constant Bit Rate*) e variável (VBR – *Variable Bit rate*).
- Suporta tráfego em *burst* (rajada).
- Considera aplicações sensíveis a atraso e perda

A técnica de comutação de células utiliza pequenos pacotes de tamanho fixo, denominados células, de 53 *bytes*. Em uma rede ATM, todas as informações a serem transmitidas (dados, voz, vídeo, entre outros) são segmentadas nas células para, a seguir, serem multiplexadas em uma cadeia de bits para a transmissão através de um meio físico.

O ATM é uma tecnologia orientada à conexão: para o transporte das células, uma conexão (ou circuito) virtual deve ser estabelecido previamente. Há três tipos de conexões [40]:

- *PVCs (Permanent Virtual Connections)*: conexões virtuais estáticas, estabelecidas manualmente;
- *SVCs (Switched Virtual Connections):* conexões virtuais dinâmicas, automaticamente pelas aplicações que as utilizam;
- *SPVCs (Semi-Permanent Virtual Connections):* conexões virtuais híbridas, que combinam características de PVCs e SVCs.

O modelo de referência do ATM é composto por 4 camadas, conforme pode ser visto na Figura I-1 [40]:

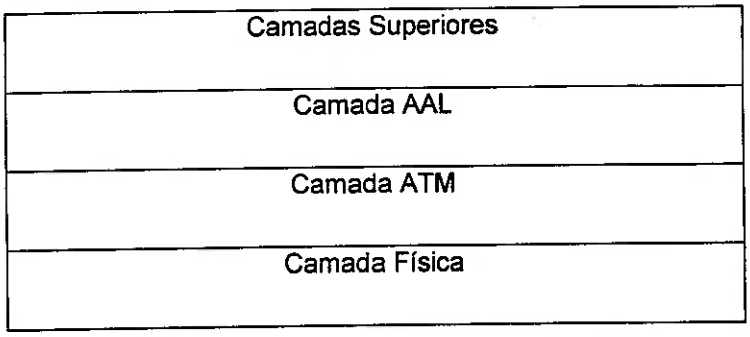


Figura I-1:Modelo de referência do ATM

A camada física é responsável pelo mapeamento das células nos quadros de transmissão dos meios de transporte físicos utilizados (por exemplo, em um enlace E3, as células devem ser mapeadas para os campos de informação dos quadros E3). O delineamento das células no quadro de transmissão e a verificação de erros no cabeçalho das células são também funções da camada física.

A camada ATM é a responsável pela geração das células, acrescentando o cabeçalho das células aos dados recebidos da camada superior e multiplexando as células para a transmissão (de maneira análoga, é responsável pela demultiplexação das células recebidas da camada física e extração dos dados destas células). A conversão de endereços ATM em identificadores das conexões virtuais e a comutação de células nos *switches* são outras funções desta camada.

A camada de adaptação AAL (*ATM Adaptation Layer*) é utilizada para suportar as funções requisitadas por camadas superiores. É responsável por segmentar os dados recebidos das camadas superiores em um tamanho apropriado para a área útil das células ATM, bem como para a remontagem do conteúdo das células recebidas em informação da camada superior.

Há 4 tipos de AALs, de acordo com os tipos de requisitos exigidos pelas camadas superiores [40]:

- AAL1: tráfego de dados em tempo real, com taxa de bit constante (por exemplo, para serviços de emulação de circuitos e voz sobre ATM);
- AAL2: tráfego de dados em tempo real, com taxa de bit variável (por exemplo, áudio e vídeo comprimido);
- AAL3/4: tráfego de dados sem requisitos de tempo, com taxa de bit variável (por exemplo, *Frame Relay* e X.25);

- AAL5: tráfego de dados sem requisitos de tempo, com taxa de bit variável (por exemplo, *LAN Emulation* e *Classical IP over ATM*) e encapsulamento com pequeno *overhead*.

Por fim, sobre a camada AAL encontram-se as camadas superiores, que fazem uso dos serviços de transmissão ATM. Como exemplos de camadas superiores, há o *LAN Emulation, Classical IP over ATM*, transmissão de voz e vídeo, entre outros.

I.1. Comutação

Conforme visto anteriormente, o ATM é uma tecnologia orientada à conexão, onde são utilizados conexões virtuais para o transporte de dados na rede. Uma conexão virtual ATM é dividida em dois níveis, denominados canal virtual ou VC (*Virtual Channel*) e via virtual ou VP (*Virtual Path*), que se relacionam da seguinte maneira: dentro do meio físico de transmissão há uma ou mais vias virtuais que, por sua vez, contém um ou mais canais virtuais (Figura I-2 [36]). A comutação pode ser realizada tanto a nível da via virtual como do canal virtual.

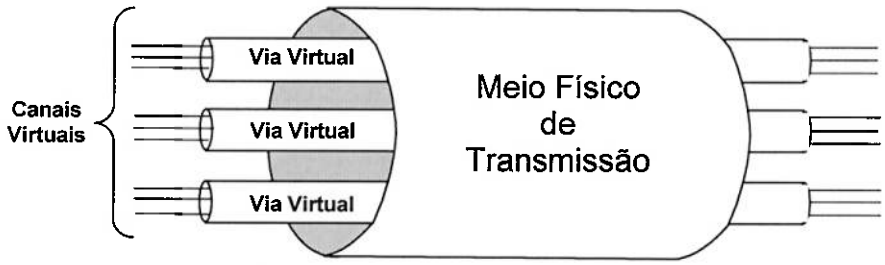


Figura I-2: Meio físico, vias virtuais e canais virtuais

Para entender os conceitos de comutação de VCs/VPs, é necessário entender os seguintes termos [36]:

- Canal Virtual ou VC: termo genérico utilizado para descrever o nível básico de conexão virtual ATM;
- Enlace de Canal Virtual ou VCL (*Virtual Channel Link*): transporte unidirecional de células ATM entre dois elementos ATM adjacentes como, por exemplo, um dispositivo terminal e um *switch* ATM ou dois *switches* ATM;
- Identificador de Canal Virtual ou VCI (*Virtual Channel Identifier*): valor numérico utilizado para identificar um VCL dentro de um dado VP;

- Conexão de Canal Virtual ou VCC (*Virtual Channel Conection*): concatenação de VCLs, que se estende entre dois pontos finais, onde a camada de adaptação é acessada. Provê transferência fim-a-fim de células ATM entre usuários ATM;
- Via Virtual ou VP (*Virtual Path*): conjunto de VCs que possuem os mesmos pontos finais;
- Enlace de Via Virtual ou VPL (*Virtual Path Link*): conjunto de VCLs entre dois elementos ATM adjacentes como, por exemplo, um dispositivo terminal e um *switch* ATM ou dois *switches* ATM;
- Identificador de Via Virtual ou VPI (*Virtual Path Identifier*): valor numérico utilizado para identificar um VPL;
- Conexão de Via Virtual ou VPC (*Virtual Path Conection*): concatenação de VCLs (ou seja, conjunto de VCCs), que se estende entre dois pontos finais, onde a camada de adaptação é acessada;

Tanto os VPLs como os VCLS são terminados em *switches*, sendo identificados, respectivamente, pelos VPIs e VCIs. Como os VCLs/VPLs possuem um escopo local, os identificadores VCI/VPI também possuem um significado local. Sendo assim, para as células que chegam de um dado VPL/VCL, o *switch* ATM deve decidir para qual enlace virtual estas células devem ser encaminhadas, alterando os identificadores correspondentes nos cabeçalhos das células (quando necessário) e encaminhando-as apropriadamente para o enlace virtual de destino, procedendo assim com a comutação.

A comutação pode ser feita tanto ao nível de VPs como de VCs. Quando a comutação é feita ao nível de VPs, o *switch* ATM (denominado *switch* VP) utiliza apenas os VPIs, ignorando os VCIs. Desta maneira, todos os VCCs de um dado VPC são comutados de forma agregada, simplificando a função de comutação (Figura I-3).

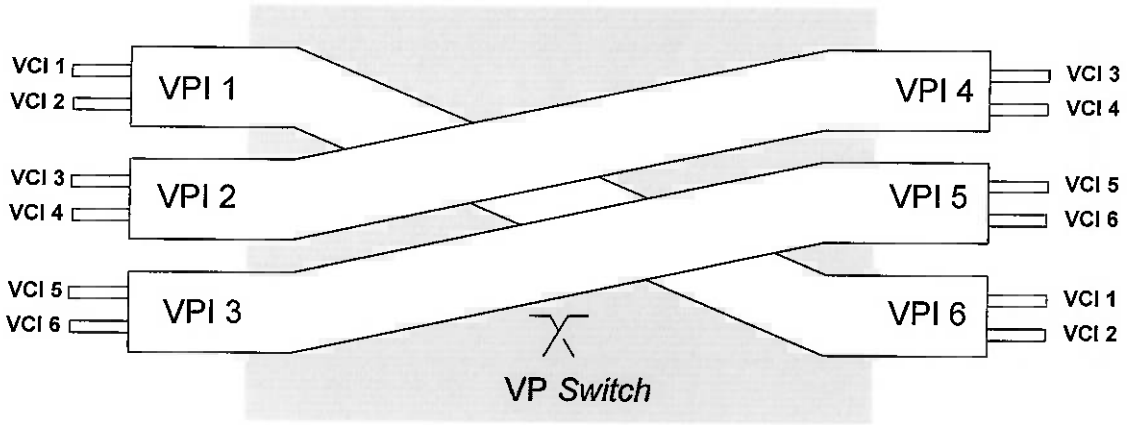


Figura I-3: Comutação de VPs

Quando a comutação é feita ao nível de VCs, o *switch* ATM (denominado *switch* VC) utiliza os valores de VPIs e VCIs para executar a comutação (Figura I-4).

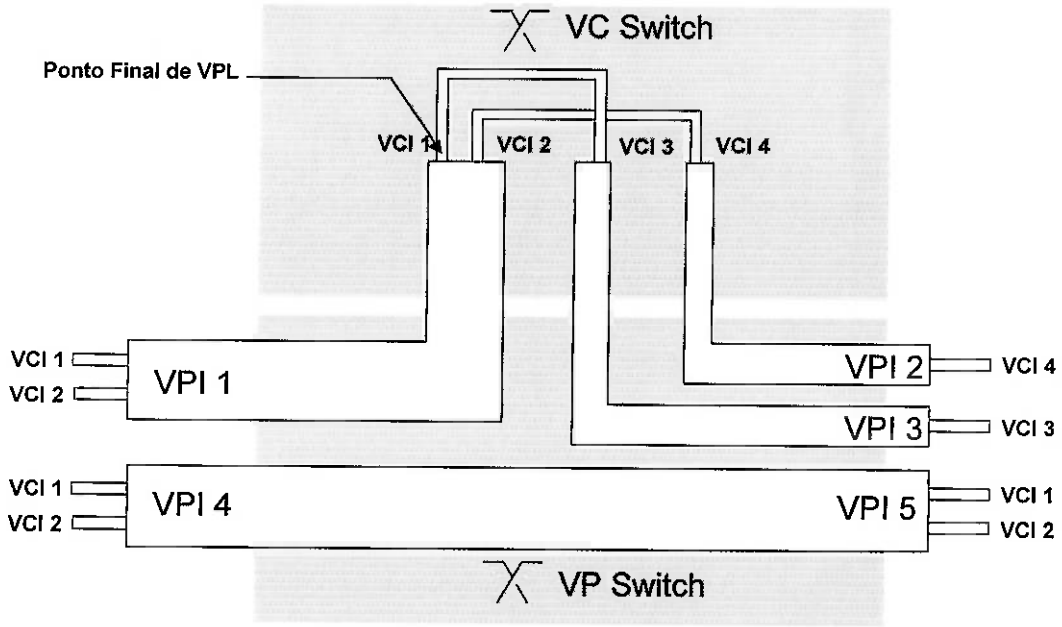


Figura I-4: Comutação de VCs

I.2 Interfaces UNI e NNI e células ATM

De acordo com os padrões, uma interface ATM define as características físicas, o formato da célula e sinalização nas conexões entre diferentes tipos de hardware, tais como: equipamentos ATM dos usuários (denominados dispositivos terminais ou TE – *Terminal Equipments*) e elementos de interconexão ATM (os *switches* ATM, também denominados IS – *Intermediate Systems*). Os principais tipos de interfaces são [36]:

- UNI (*User-Network Interface*): utilizada, dentro de uma rede privativa, entre um *switch* ATM e um dispositivo terminal, recebendo o nome de UNI Privativa. Pode ser usada também entre uma rede pública e uma privativa, tanto entre um dispositivo terminal e um *switch* ATM público como entre um *switch* ATM público e um privativo), recebendo o nome de UNI Pública (Figura I-5);
- NNI (*Network-Network Interface*): utilizada, dentro de uma rede privativa, entre *switches* ATM, recebendo o nome de NNI Privativa ou P-NNI. Pode ser usada também entre *switches* ATM de uma rede pública, recebendo o nome de NNI Pública (Figura I-5).

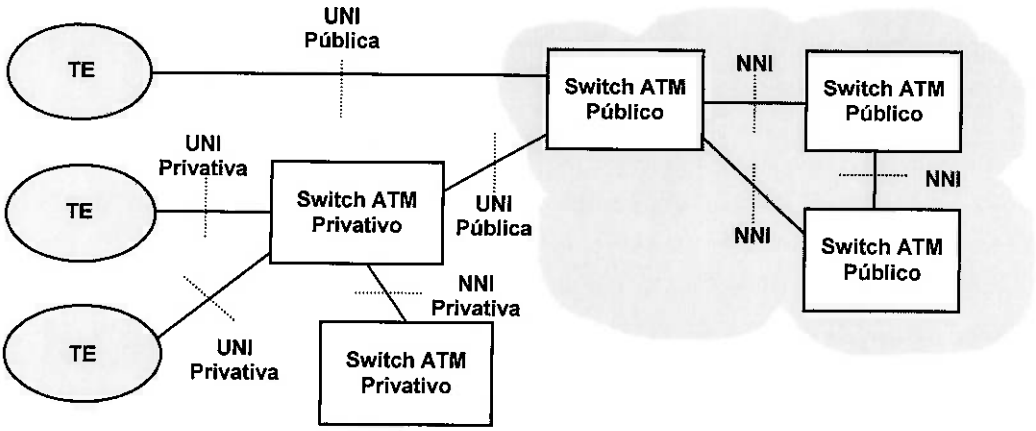


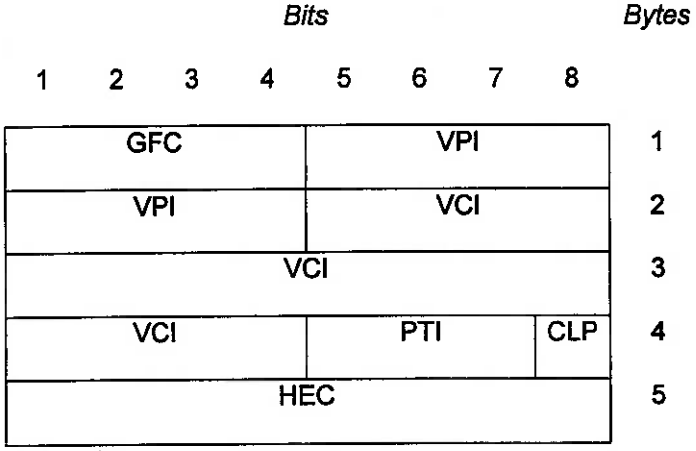
Figura I-5: Posição das Interfaces UNI e NNI

I.3 Células ATM

Para a transmissão de dados, o ATM opera com células de tamanho fixo de 53 *bytes*, consistindo de um cabeçalho de 5 *bytes* e um campo de informação de 48 *bytes*. Uma das vantagens da utilização de células pequenas com tamanho fixo é a redução do tempo de espera em fila das células de alta prioridade pois, como todas as mensagens são segmentadas

em células pequenas, mensagens longas não causam atrasos das mensagens curtas. Além disso, a comutação é mais eficiente, uma vez que o tamanho fixo das células simplifica a implementação dos *switches* ATM.

São definidos dois formatos de cabeçalho das células ATM, usados respectivamente nas interfaces UNI e NNI. O formato do cabeçalho da célula ATM utilizado na UNI e na NNI podem ser vistos, respectivamente, na Figura I-6 e Figura I-7 [36]:



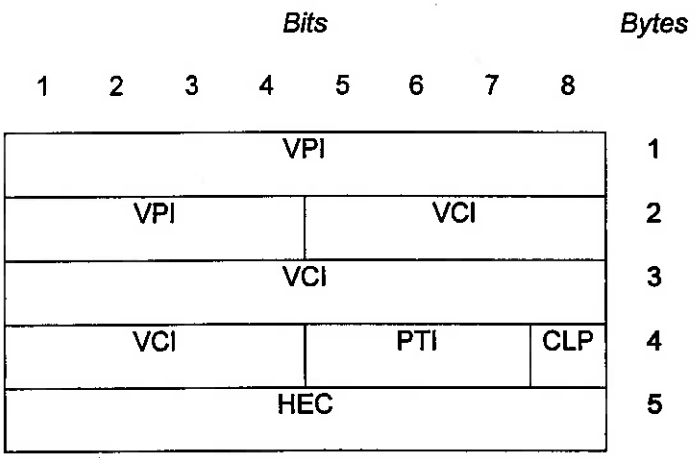
Legenda:

CLP – *Cell Loss Priority* PTI – *Payload type Identifier*

GFC – *Generic Flow Control* VCI – *Virtual Channel Identifier*

HEC – *Header Error Control* VPI – *Virtual Path Identifier*

Figura I-6: Cabeçalho de uma célula na Interface UNI



Legenda:

CLP – *Cell Loss Priority* VCI – *Virtual Channel Identifier*

HEC – *Header Error Control* VPI – *Virtual Path Identifier*

PTI – *Payload type Identifier*

Figura I-7: Cabeçalho de uma célula na Interface NNI

O cabeçalho das células consiste dos seguintes campos:

- GFC (*Generic Flow Control*): 4 *bits*, usado para o controle de fluxo e para o controle de acesso ao meio na interface UNI; tais funções não estão, ainda, totalmente definidas;
- VCI: 8 *bits* (UNI) e 12 *bits* (NNI), identifica o VP ao qual pertence a célula. O maior número de *bits* na interface NNI visa suportar um maior número de VPs dentro da rede;
- VPI: 16 *bits*, identifica o VC ao qual a célula pertence;
- PTI (*Payload Type Identifier*): 3 *bits*, identifica o tipo de informação contida no campo de informação da célula (por exemplo, se o seu terceiro *bit* for igual a "0", a célula contém dados do usuário e, se for igual a "1", a célula contém informações de gerenciamento e manutenção);
- CLP (*Cell Loss Priority*): 1 *bit*, indica se a célula pode ser descartada no caso de congestionamento da rede. O valor "0" indica um célula de alta prioridade que não deve ser descartada;
- HEC (*Header Error Control*): 8 *bits*, corresponde a um campo de controle de erro do cabeçalho, calculado sobre os demais 32 *bits* do cabeçalho utilizando o polinômio gerador X8 + X2 + X + 1. Permite detectar múltiplos erros de *bits* e corrigir um único erro de bit.

I.4 LAN Emulation (LANE)

O serviço de *LAN Emulation* (LANE [40]) visa habilitar que aplicações existentes possam acessar uma rede ATM, através de protocolos tradicionais, como se estes protocolos estivessem utilizando uma rede local. O *LAN Emulation*, que trabalha no nível da camada MAC (*Media Access Control*), utiliza o conceito de uma LAN Emulada ou ELAN (*Emulated LAN*). Uma ELAN agrupa logicamente estações e outros dispositivos de rede, de maneira análoga ao agrupamento determinado por segmentos de rede físicos. Membros de uma ELAN podem se comunicar livremente; porém, para a comunicação entre diferentes ELANs são necessários dispositivos de interconexão, como uma ponte ou um roteador.

I.4.1 Componentes do padrão LANE

O LANE baseia-se em um modelo cliente/servidor. Cada ELAN é composta por um grupo de clientes, denominados LECs (*LAN Emulation Clients*) e três servidores, o LECS (*LAN Emulation Configuration Server*), o LES (*LAN Emulation Server*) o BUS (*Broadcast and Unknown Server*). Possuem as seguintes funções [40]:

- LEC: é o *software* utilizado nos dispositivos terminais ATM para que eles possam se comunicar dentro de uma ELAN. Entre suas funções destacam-se a transmissão de dados e a resolução de nomes. Além de um endereço ATM, um LEC possui um ou mais endereços MAC, dependendo do seu tipo: um LEC *proxy* representa os endereços MAC de outros dispositivos, atuando como uma ponte, enquanto que um LEC não-*proxy* representa um dispositivo terminal, com um único endereço MAC. Deve haver um LEC para cada ELAN à qual pertence o dispositivo ATM;
- LES: é o responsável pelo coordenação de uma ELAN. Entre as suas principais funções destacam-se o registro de clientes LANE na ELAN e a resolução de endereços MAC para endereços ATM. Deve haver um LES para cada ELAN;
- BUS: é o responsável pelas transmissões em *broadcast* em uma ELAN. Como não existe *broadcast* em redes ATM, o BUS emula esta função: o dado de *broadcast* é enviado para o BUS, que redistribui a todos os LECs da ELAN. Deve haver um BUS para cada ELAN;
- LECS: é o responsável pela configuração inicial dos LECs. Um LEC pode, em sua inicialização, requisitar ao LECS informações de configuração, tais como: a ELAN a qual o LEC irá pertencer, o endereço ATM do LES responsável por esta ELAN, entre outros. Um LECS pode atender a diversas ELANs.

I.4.2 Conexões virtuais utilizadas

Para prover os serviços de LAN Emulation, diversas conexões virtuais são utilizadas, conforme pode ser visto na Tabela I-1 [40].

Nome VCC	**Pontos terminais da Conexão**	**Sentido**	**Tipo**
Configuration Direct VCC	LECS-LEC	Bidirecional	Ponto-a-ponto
Control Direct VCC	LES-LEC	Bidirecional	Ponto-a-ponto
Control Distribute VCC	LES-LEC	Unidirecional	Ponto-multiponto
Multicast Send VCC	BUS-LEC	Bidirecional	Ponto-a-ponto
Multicast Forward VCC	BUS-LEC	Unidirecional	Ponto-multiponto
Data Direct VCC	LEC-LEC	Bidirecional	Ponto-a-ponto

Tabela I-1: VCCs utilizados pelo serviço de LANE

As funções de cada VCC são apresentadas a seguir:

- *Configuration Direct VCC*: estabelecida pelo LEC durante sua inicialização, é utilizada para se obter informações de configuração junto ao LECS;
- *Control Direct VCC*: estabelecido pelo LEC em sua inicialização, é utilizado para o registro do LEC na ELAN, para a resolução de endereços, entre outros;
- *Control Distribute VCC*: estabelecido pelo LES, esta conexão virtual ponto-multiponto liga o LES a todos os clientes da ELAN. É utilizada para a resolução de nomes, quando o LES não conhece o endereço MAC requisitado por um cliente;
- *Multicast Send VCC*: estabelecido pelo LEC em sua inicialização, é utilizado para o envio de dados de *multicast* e *broadcast* para os clientes da ELAN;
- *Multicast Forward VCC*: estabelecido pelo BUS, esta conexão virtual ponto-multiponto liga o BUS a todos os clientes da ELAN. É utilizada pelo BUS para redistribuir dados recebidos de um dado LEC para todos os LECs da rede;
- *Data Direct VCC*: estabelecido por um LEC, é utilizado para a transmissão de dados de *unicast* entre dois LECs.

I.4.3 Funcionamento do Protocolo

Na inicialização de um LEC, ele busca o LECS da rede para obter as informações necessárias à sua configuração, através do *Configuration Direct VCC* [40]. O LECS devolve ao LEC o nome e tipo da ELAN, o tamanho de quadro utilizado na ELAN e o endereço ATM do LES responsável pela ELAN, entre outros.

A partir do endereço do LES recebido, o LEC estabelece o *Control Direct VCC* com o LES, requisitando o seu registro na ELAN. A partir deste pedido de registro, o LES armazena o endereço ATM e MAC do LEC, para resoluções de endereço futuras, e acrescenta o LEC no *Control Distribute* VCC.

Após registrado na ELAN, o LEC envia para o LES um pedido de resolução para o endereço de *broadcast* da rede (ou sej.a para o endereço do BUS). Quando a resposta com o endereço do BUS é recebida, o LEC estabelece o *Multicast Send VCC* com o BUS, que responde acrescentando o LEC no *Multicast Forward VCC*.

Para a transmissão de um quadro *unicast*, o LEC verifica em seu cache de endereços LE_ARP se há uma entrada com o endereço ATM associado ao endereço MAC de destino. Caso não haja, o LEC envia para o LES uma requisição para a resolução de endereços, através do *Control Direct VCC*. Se o LES tiver os endereços necessários, a resolução é respondida; caso contrário ele encaminha o pedido de resolução para todos os LECs da ELAN, através do *Control Distribute* VCC, e, quando a resolução for respondida ao LES, ele encaminha a resposta ao LEC que iniciou a resolução. Quando o LEC com o quadro a ser transmitido recebe do LES a resposta do seu pedido de resolução de endereço, ele estabelece com o LEC destino um *Data Direct VCC* para a transmissão do quadro.

Para a transmissão de um quadro de *broadcast* ou *multicast*, o LEC envia o quadro ao BUS, utilizando o *Multicast Send VCC*. O BUS, por sua vez, retransmite o quadro para todos os LECs da ELAN, através do *Multicast Forward VCC*, emulando assim o comportamento de uma LAN para transmissão em *broadcast*.

I.5 Classical IP over ATM (CLIP)

O serviço de *Classical IP over ATM* (CLIP) visa habilitar que aplicações existentes possam acessar uma rede ATM, através do protocolo TCP/IP **[40]**.

Em uma rede ATM com CLIP as sub-redes IPs são lógicas e são denominadas LIS (*Logical IP Subnetwork*). Membros de uma LIS podem se comunicar livremente, como se pertencessem à mesma rede local. As diferentes LIS são interconectadas através de roteadores, para permitir a comunicação entre elementos pertencentes a LIS distintas.

Como no ATM não há uma funcionalidade que permita o broadcast, o protocolo ARP (*Address Resolution Protocol*) tradicional não pode ser utilizado. A resolução de endereços em uma LIS é feita através de um servidor ATMARP, que faz o mapeamento de endereços IPs em endereços ATM.

I.5.1 Componentes do CLIP

O CLIP baseia-se em um modelo cliente/servidor. Cada LIS é composta por um grupo de clientes, denominados clientes CLIP e um servidor, o servidor ATMARP. Possuem as seguintes funções [40]:

- Cliente CLIP: é o responsável pela integração dos dispositivos terminais ATM em uma LIS, permitindo a comunicação destes dispositivos dentro de uma LIS. A principal função do cliente CLIP é a transmissão de dados: através da adição de um cabeçalho LLC (*Logical Link Control* - padrão IEEE 802.2) os pacotes IP são preparados para o encapsulamento em PDUs (*Protocol Data Units*) da camada de adaptação AAL5. Deve haver um cliente CLIP para cada LIS à qual pertence o dispositivo ATM;
- Servidor ATMARP: é o responsável pelo coordenação de uma LIS. Entre as suas principais funções destacam-se o registro de clientes CLIP na LIS e a resolução de endereços IP para endereços ATM, através da manutenção de uma base de dados com os endereços IP e ATM de todos os clientes da LIS;

I.5.2 Funcionamento do Protocolo

O CLIP pode utilizar tanto PVCs como SVCs para a comunicação entre os elementos de uma LIS. De acordo com o tipo de conexão utilizada, o funcionamento do protocolo é diferente, devendo ser visto em separado.

I.5.2.1 Utilização de PVCs

Para a comunicação entre dois dispositivos terminais de uma LIS utilizando PVCs, deve-se configurar manualmente o PVC entre os dispositivos ATM que irão se comunicar. Para a transmissão dos pacotes IP, é necessário mapear o endereço IP destino para o PVC que liga o dispositivo ATM de origem ao dispositivo ATM de destino.

Esta resolução é feita utilizando uma requisição ATMARP inversa (*InATMARP_Request*). Para saber o endereço IP e ATM do dispositivo remoto (na outra ponta do PVC), o cliente CLIP envia uma requisição *InATMARP_Request; a* resposta *InATMARP_Reply* enviada pelo dispositivo remoto contém as informações necessárias desejadas. Executando este procedimento para todos os PVCs configurados, um cliente CLIP mapeia o endereços IP de cada dispositivo remoto ao PVC a ser utilizado para a transmissão.