

**INSTITUTO FEDERAL DE SANTA CATARINA
IF-SC**

**CURSO DE PÓS-GRADUAÇÃO LATO SENSU EM DESENVOLVIMENTO DE
PRODUTOS ELETRÔNICOS**

IMPLEMENTAÇÃO DE UMA MIB SNMP PARA PABX INTELBRÁS

PEDRO PAULO DA SILVA

FLORIANÓPOLIS/SC

2009

**INSTITUTO FEDERAL DE SANTA CATARINA
IF-SC**

**CURSO DE PÓS-GRADUAÇÃO LATO SENSU EM DESENVOLVIMENTO DE
PRODUTOS ELETRÔNICOS**

IMPLEMENTAÇÃO DE UMA MIB SNMP PARA PABX INTELBRÁS

Monografia apresentada à coordenação do Curso de Pós-Graduação Lato Sensu em Desenvolvimento de Produtos Eletrônicos para a obtenção do título de Especialista em Desenvolvimento de Produtos Eletrônicos.

**FLORIANÓPOLIS/SC
2009**

Pedro Paulo da Silva

Monografia sob o título Implementação de Uma MIB SMMP para PABX Intelbrás, defendida por Pedro Paulo da Silva em 10 de julho de 2009 e aprovada pela banca examinadora constituída conforme abaixo:

Prof. Dr. Charles Borges de Lima

Prof. Dr. André Luis Dalcastangê

Prof. Msc. Leandro Schwarz

FLORIANÓPOLIS/SC

2009

AGRADECIMENTOS

Agradeço a todos aqueles que me incentivaram a não deixar em branco este trabalho: minha esposa Marta, minhas filhas Maria Helena e Carina e meus pais, Inácio e Oscarina. Agradeço a compreensão de todos porque durante muitos fins-de-semana eu não estive disponível para eles. Agradeço a Deus por tê-los comigo e por me permitir caminhar na direção do meu crescimento. Agradeço aos colegas de curso e aos professores pelo aprendizado que, certamente, extrapolou o conteúdo das aulas.

RESUMO

O presente trabalho descreve a criação de uma MIB SNMP para a linha Impacta de centrais telefônicas produzidas pela Intelbrás, incorporando a característica de gerenciamento remoto a estes equipamentos. Nele são apresentados o funcionamento do protocolo SNMP, as regras de construção de uma MIB e características das centrais Impacta. É apresentado também o documento formal da MIB implementada e o procedimento para sua incorporação ao agente *Net-SNMP*.

ÍNDICE DE FIGURAS

Figura 1. Arquitetura do Gerenciamento SNMP.....	4
Figura 2. A Arquitetura SNMP.....	9
Figura 3. Amostra da Árvore da MIB.....	10
Figura 4. A PDU GetRequest.....	11
Figura 5. A PDU SetRequest.....	12
Figura 6. Formato de uma PDU SNMP.....	12
Figura 7. Tela de Configuração de Ramais - Programador IMPACTA.....	27
Figura 8. Subárvore Impacta.....	29
Figura 9. A Ferramenta SimpleWeb.....	30

LISTA DE ABREVIATURAS E SIGLAS

- API : *Application Programming Interface* – Interface de Programação de Aplicação - é um conjunto de rotinas e padrões estabelecidos por um *software* para a utilização das suas funcionalidades por programas aplicativos.
- ASN.1 : *Abstract Syntax Notation One* – Notação de Sintaxe Abstrata Um – é uma linguagem abstrata utilizada na especificação formal de protocolos de comunicação.
- CCITT : *Comite Consultatif International Telegraphique et Telephonique* - Comitê Consultivo Internacional de Telefonia e Telegrafia (atualmente substituído pelo ITU-T) – órgão responsável pelo estudo técnico de questões relativas à operação, tarifação e elaboração de recomendações para padronizar as telecomunicações a nível mundial.
- CMIP : *Common Management Information Protocol* – Protocolo de Informações de Gerenciamento Comum – protocolo que define os procedimentos para a transformação de informações de gerenciamento e define a sintaxe para o serviço de gerenciamento do CMIS.
- CMIS : *Common Management Information Service* ou Serviço de Informações de Gerenciamento Comum – define os serviços para a operação de gerenciamento.
- CMISE : *Common Management Information Service Element* – Elemento de Serviço para Informações de Gerenciamento Comum – elemento de rede que implementa o CMIS.
- DoD : *U. S. Department of Defense* – Departamento de Defesa dos Estados Unidos da América
- IAB : *Internet Architecture Board* – Quadro de Arquitetura da Internet – órgão responsável pelo estudo e implementação de protocolos e soluções para a Internet.
- IANA : *Internet Assigned Numbers Authority* – Autoridade de Números Atribuídos na Internet – órgão da Internet responsável pela atribuição de números a protocolos, empresas etc.

- ISO : *International Organization for Standardization* – Organização Internacional para Padronização – órgão internacional responsável pela padronização e normatização de procedimentos e operações em diversas áreas do conhecimento e tecnologia humanos.
- ITU : *International Telecommunication Union* – União de Telecomunicações Internacional – organização internacional criada para padronizar e regulamentar as comunicações de uma maneira geral.
- ITU-T : Setor de Padronização de Telecomunicações do ITU (antigo CCITT)
- MIB : *Management Information Base* – Base de Informações de Gerenciamento – é um conjunto de objetos gerenciados definido de modo a abranger todas as informações necessárias para o gerenciamento de um determinado dispositivo. Existem as MIBs padrão da Internet e as MIBs definidas pelos fabricantes de dispositivos de rede, que possuem características específicas.
- MIB-II : A versão 2 da MIB padrão da Internet
- MRTG : *Multi Router Traffic Grapher* – Sistema Gráfico para Tráfego Multi-Roteador – ferramenta para analisar tráfego de rede de forma gráfica.
- NIST : *National Institute of Standards and Technology* – Instituto Nacional de Padrões e Tecnologia – agência governamental não regulatória da Administração de Tecnologia do Departamento de Comércio dos Estados Unidos.
- OID : *Object Identifier* – Identificador de Objetos – corresponde a uma seqüência numérica que identifica de forma universalmente única um objeto gerenciado. Esta seqüência numérica é obtida através da posição do objeto na árvore da Internet.
- OSI : *Open Systems Interconnection* – Interconexão de Sistemas Abertos – conjunto de padrões ISO relacionados à comunicação de dados.
- PABX : *Private Automated Branch Exchange* – Intercâmbio de Ramos Automatizado Privado – é um centro de distribuição telefônica pertencente a uma empresa que não inclua como sua atividade o fornecimento de serviços telefônicos ao público em geral.

- PDU : *Protocol Data Unit* – Unidade de Dados de Protocolo – corresponde a um bloco mínimo de informações trocadas como uma unidade entre dois elementos que se comunicam através de um protocolo.
- RFC : Request For Comments – Requisição por Comentários – é um documento que descreve os protocolos da Internet antes de se tornarem padrões.
- SGR : Sistema de Gerenciamento de Redes
- SMI : *Structure of Management Information* – Estrutura das Informações de Gerenciamento – documento que regulamenta a forma como uma MIB deve ser escrita.
- SNMP : *Simple Network Management Protocol* - Protocolo Simples de Gerenciamento de Redes – é um protocolo de aplicação de gerência típico das redes TCP/IP que padroniza o intercâmbio de informações entre dispositivos de rede.
- SNMPv : Versão 3 do SNMP
- 3
- TCP/IP : *Transmission Control Protocol/Internet Protocol* – Protocolo de Controle de Transmissão/Protocolo da Internet – é um conjunto de protocolos de comunicação entre dispositivos em rede. Seu nome se origina dos seus dois principais protocolos: TCP (*Transmission Control Protocol* ou Protocolo de Controle de Transmissão) e IP (*Internet Protocol* ou Protocolo da Internet)
- TI : Terminais Inteligentes – são terminais que permitem ao usuário a configuração da central telefônica aos quais estão ligados.
- UDP : *User Datagram Protocol* – Protocolo de Datagramas de Usuário – é um protocolo de camada de transmissão mais simples, sobre o qual está definido o SNMP.
- VoIP : Voice Over Internet Protocol – Voz Sobre o Protocolo da Internet – é uma tecnologia que permite estabelecer comunicações telefônicas através da Internet.

SUMÁRIO

RESUMO	iv
ÍNDICE DE FIGURAS	v
LISTA DE ABREVIATURAS E SIGLAS.....	vi
INTRODUÇÃO	1
CAPÍTULO I – GERENCIAMENTO DE REDES.....	4
1.1 INTRODUÇÃO.....	4
1.2 ARQUITETURA DO GERENCIAMENTO.....	5
1.2.1 Estação de Gerenciamento.....	5
1.2.2 Agente de Gerenciamento.....	6
1.2.3 Base de Informações de Gerenciamento.....	6
1.2.4 Protocolo de Gerenciamento de Redes.....	7
1.3 RESUMO.....	8
CAPÍTULO II – O PROTOCOLO SNMP.....	9
2.1 INTRODUÇÃO.....	9
2.2 A ARQUITETURA DO SNMP.....	10
2.3 A ESTRUTURA DE DADOS DO SNMP.....	10
2.4 O PROTOCOLO SNMP.....	11
CAPÍTULO III – BASE DE INFORMAÇÕES DE GERENCIAMENTO - MIB.....	15
3.1 INTRODUÇÃO.....	15
3.2 ESTRUTURA DAS INFORMAÇÕES DE GERENCIAMENTO.....	15
3.2.1 Nomes.....	16
3.2.1.1 A Subárvore directory.....	18
3.2.1.2 A Subárvore mgmt.....	18
3.2.1.3 A Subárvore experimental.....	18
3.2.1.4 A Subárvore Private.....	19
3.2.2 Sintaxe.....	19
3.2.2.1. Tipos Primitivos.....	20
3.2.2.2. Tipos Construtores.....	20
3.2.2.3. Tipos Definidos.....	20
3.2.3. CODIFICAÇÕES.....	21
3.3. OBJETOS GERENCIADOS.....	22
3.3.1. Orientação para Nomes de Objetos.....	22
3.3.2. Instâncias e Tipos de Objetos.....	23
CAPÍTULO IV – A MIB IMPACTA.....	26
4.1 INTRODUÇÃO.....	26
4.2 CRIAÇÃO DA MIB IMPACTA.....	26
4.2.1 As Centrais Impacta.....	26
4.2.2 O Gerenciamento das Centrais Impacta.....	28
4.2.3 As Variáveis de Gerenciamento.....	29
4.2.4 A MIB Impacta.....	31
4.2.5 A Implementação da MIB no Agente.....	43
4.2.5.1 A instalação do Net-SNMP.....	43
4.2.5.2 A inclusão da MIB Impacta no agente [10].....	43
4.2.5.3 Execução e Teste do Agente.....	45
4.2.5.4 Análise do Resultado.....	46
CONSIDERAÇÕES FINAIS.....	47
GLOSSÁRIO.....	49
REFERÊNCIAS.....	51

INTRODUÇÃO

No início dos anos 1980 houve uma enorme expansão no desenvolvimento de redes de computadores. Com ganhos de produtividade decorrentes da utilização da tecnologia de redes, as empresas começaram a instalar e expandir suas redes de computadores quase tão rapidamente quanto o surgimento de novas tecnologias e produtos. Em função disso, já naquela época, algumas companhias começaram a experimentar problemas com a utilização de tecnologias de redes diferentes e até mesmo incompatíveis.

Estes problemas ocorriam tanto na operação diária do gerenciamento das redes como no planejamento para a sua expansão. Isso porque cada nova tecnologia exigia seu próprio conjunto de profissionais capacitados que, por sua vez, defendiam as “suas” tecnologias como as mais adequadas para uma possível expansão. Naquela época a demanda de pessoal para o gerenciamento de redes grandes e heterogêneas já criava uma crise para muitas organizações.

Surgiu assim a necessidade de simplificar a estratégia para o gerenciamento das redes através da definição de um gerenciamento unificado e automatizado. Tal gerenciamento deveria facilitar a integração dos diversos ambientes e o planejamento da capacidade da rede não só com vistas à sua utilização ótima mas também, sua expansão. Com esta motivação, começaram a surgir grupos de estudo e pesquisa com o intuito de criar um protocolo de gerenciamento que atendesse às necessidades daquele ambiente. Os primeiros protocolos que surgiram a partir deste esforço foram o SNMP (*Simple Network Management Protocol*, ou Protocolo Simples de Gerenciamento de Redes) [1] e o CMIP (*Common Management Information Protocol*, ou Protocolo Comum de Informações de Gerenciamento) [2].

A idéia inicial do gerenciamento era dar suporte à operação e manutenção remotas da rede. Em virtude disso, os dispositivos que apresentavam característica de gerenciamento eram somente aqueles que proporcionavam a conectividade e o controle entre os diversos trechos da rede, como *hubs*, pontes (*bridges*), roteadores etc.

Com a descoberta das vantagens que o gerenciamento de redes proporcionava, as características de gerenciamento foram sendo incorporadas em outros dispositivos

ligados em redes mas não diretamente associados ao seu funcionamento: impressoras, fontes de energia ininterrupta (*no-break*), sistemas de controle de variáveis ambientais, etc. Hoje em dia, chegamos a uma condição em que é recomendado que todo dispositivo conectado a uma rede de comunicação seja gerenciável.

O primeiro passo para tornar um dispositivo gerenciável é definir uma MIB com suas características. Com isso, o presente trabalho descreve a definição de uma Base de Informações de Gerenciamento (*MIB-Management Information Base*) para uma central telefônica produzida pela Intelbrás¹. Esta central da linha Impacta é dotada de interfaces seriais RS232 para o seu gerenciamento local e de uma conexão com rede de comunicação proporcionada por uma placa utilizada exclusivamente para o estabelecimento de comunicação telefônica via VoIP (*Voice-over-IP* ou voz sobre IP).

Embora favoreça o aspecto de segurança de acesso à central, a conexão RS232 é limitada em sua velocidade e não permite gerenciamento remoto. Isso porque a máquina que gerencia tem que estar próxima desta central telefônica. Este aspecto dificulta a inclusão das centrais Impacta em um sistema de gerenciamento centralizado.

A criação da MIB Impacta e do agente SNMP que a implementa, tem o principal objetivo de permitir o gerenciamento remoto e centralizado das centrais Impacta tanto no aspecto de sua configuração quanto no aspecto de seu monitoramento operacional. Assim, a definição da MIB Impacta é o passo mais importante para a criação do *Agente SNMP*. É nesta MIB onde estão delimitadas as estratégias e o objetivo do gerenciamento para estes dispositivos.

Este documento é apresentado em duas partes complementares. Inicialmente é feito um embasamento teórico nos três primeiros capítulos. Na seqüência, a descrição do trabalho realizado. No Capítulo I é apresentada uma introdução aos conceitos básicos de um sistema de gerenciamento de redes. Cada elemento que forma este tipo de sistema é apresentado. No Capítulo II é apresentado o SNMP, sua arquitetura, sua estrutura de dados e suas mensagens. No Capítulo III é feita uma descrição de como é definida uma MIB através da apresentação de uma estrutura para informações de gerenciamento. Finalmente, no Capítulo IV, é apresentada a implementação da MIB Impacta. Isso é feito através da apresentação das centrais Impacta, de uma descrição das suas características, descrição do seu sistema de gerenciamento tradicional e da apresentação do documento formal da MIB Impacta. Ao final, são tecidos alguns comentários sobre a implementação desta MIB em um agente SNMP.

1 Empresa fundada em 1976, atua nas áreas de telecomunicações, segurança eletrônica e informática com presença em todo o território nacional e em diversos países da América Latina e África. A Intelbrás é uma empresa 100% nacional e é líder no mercado brasileiro de centrais telefônicas.

CAPÍTULO I – GERENCIAMENTO DE REDES

1.1 INTRODUÇÃO

A área de gerência de redes foi inicialmente impulsionada pela necessidade de monitoramento e controle do universo de dispositivos que compõem as redes de comunicação. No início, havia apenas pequenas redes, cada uma operando dentro de sua própria área. Com o passar do tempo, essas redes foram se interligando para formar um sistema de comunicação mais complexo. À medida em que essa rede geral ia crescendo, tornava-se cada vez mais difícil de ser administrada. Percebeu-se, então, que havia a necessidade de um padrão para o gerenciamento de redes [3].

O primeiro padrão a ser definido foi o SNMP (*Simple Network Management Protocol* ou Protocolo Simples de Gerenciamento de Redes) [1]. Sua estrutura era simples pois a idéia era que ele fosse usado provisoriamente, enquanto outros protocolos mais bem estruturados fossem sendo projetados. O SNMP é, na verdade, um conjunto de padrões para gerenciamento que inclui um protocolo, uma especificação para a estrutura de dados e um conjunto de objetos de dados. Ele é definido como um protocolo de aplicação e utiliza o protocolo UDP, do conjunto de protocolos TCP/IP, como camada de transporte. O SNMP hoje já está na sua terceira versão oficial, SNMPv3, sendo que a quase totalidade dos dispositivos gerenciáveis é compatível com este padrão.

Um dos protocolos bem estruturados que estavam sendo definidos no início, era o CMIP (*Common Management Information Protocol* [2] ou Protocolo Comum de Informações de Gerenciamento). O CMIP é um protocolo de gerenciamento definido segundo o padrão *OSI*. Ele especifica como deve ser realizada a troca de informações entre o gerente e o agente no sistema de gerenciamento. Os tipos de informações a serem trocadas levam em conta o CMIS (*Common Management Information Service* ou Serviço de Informações de Gerenciamento Comum), que especifica o conjunto de serviços que os sistemas gerenciador e gerenciado poderão acessar. Juntos, CMIS e CMIP formam o que é chamado de CMISE (*Common Management Information Service Element* ou Elemento de Serviço de Informações de Gerenciamento Comum), que é uma aplicação da camada 7 do Modelo de Referência OSI.

A demora na conclusão da definição de protocolos como o CMIP associada à

crescente utilização das redes TCP/IP impulsionaram a popularização do SNMP e relegaram as opções mais complexas, como o CMIP, a um segundo plano.

Uma vez que a implementação deste trabalho destina-se a uma aplicação SNMP, este será o padrão utilizado na demonstração dos conceitos relacionados ao gerenciamento de redes.

1.2 ARQUITETURA DO GERENCIAMENTO

O SNMP foi criado sobre o modelo cliente/servidor de comunicação e é composto basicamente pelos seguintes elementos:

- Estação de Gerenciamento;
- Agente de Gerenciamento;
- Base de Informações de Gerenciamento (MIB);
- Protocolo de Gerenciamento de Redes.

A Figura 1 apresenta de forma esquemática a arquitetura do gerenciamento SNMP.

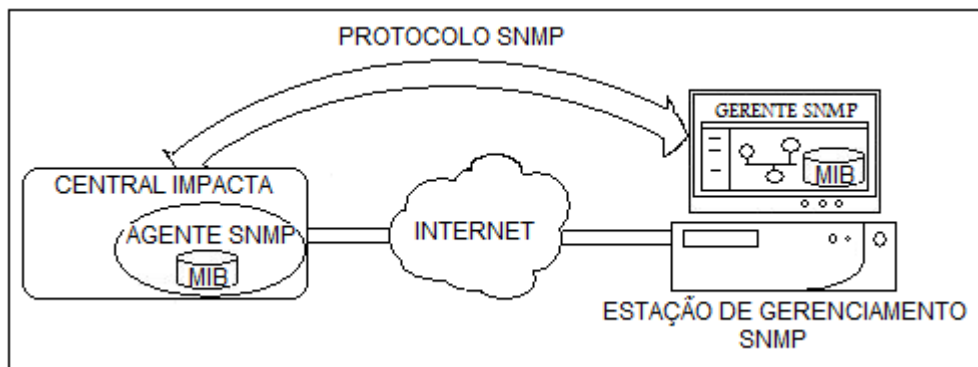


Figura 1. Arquitetura do Gerenciamento SNMP

1.2.1 Estação de Gerenciamento

Uma estação de gerenciamento é composta por um computador e um *software* aplicativo para gerenciamento, como, por exemplo, MRTG, HP OpenView, SNMPc etc, sendo este *software* o cliente das operações de gerenciamento.

Com esse perfil, a estação de gerenciamento serve como interface para o gerente

humano num sistema de gerenciamento de rede. Através dela, o gerente pode acessar as informações de cada dispositivo gerenciado, monitorando o seu estado operacional e, até mesmo, alterando variáveis que interferem neste estado operacional.

Além disso, uma estação de gerenciamento pode ser programada para reagir de forma automatizada a eventos reportados por um dispositivo gerenciado sem a interferência do gerente humano. Este tipo de funcionalidade favorece a continuidade da operação de uma rede.

A estação de gerenciamento passa a reconhecer as variáveis particulares de um equipamento após compilar a sua MIB.

1.2.2 Agente de Gerenciamento

O agente é um aplicativo que roda dentro de um dispositivo gerenciado e funciona como servidor nas operações do protocolo de gerenciamento. É dentro do agente que a MIB é implementada e onde cada variável passa a ter uma conexão com o dispositivo físico em que reside. Os recursos gerenciados são representados como objetos da MIB. Para tanto, o agente está acoplado logicamente ao *hardware* para que possa ser usado para o monitoramento e configuração do sistema.

O agente responde solicitações de informações e de ações da estação de gerenciamento e deve também prover assincronamente informações importantes que não foram solicitadas por esta estação. Estas informações assíncronas são alarmes que denunciam eventos importantes ou situações críticas dentro da operação do dispositivo gerenciado.

1.2.3 Base de Informações de Gerenciamento

Os recursos a serem gerenciados são representados como objetos e a coleção de objetos é referenciada como Base de Informações de Gerenciamento (MIB). A MIB é uma base de dados cuja estrutura é especificada pelo padrão SMI. Ela pode ser caracterizada como uma base de dados ativa, o que possibilita que os valores das suas variáveis sejam não só recuperados, como também alterados.

A MIB também é um documento formal padronizado onde estão definidas as variáveis que um determinado dispositivo implementa. Este documento é gerado para que uma estação de gerenciamento, após compilá-lo, incorpore as variáveis específicas de um determinado dispositivo na sua base de dados global. Dessa forma, todas as características gerenciáveis específicas que um fabricante incorpora ao seu equipamento poderão ser exploradas através de um sistema de gerenciamento padrão.

Neste trabalho, MIB pode significar tanto a base de dados específica implementada em um determinado dispositivo como o documento formal que a define.

Cada agente deve manter sua própria instância da MIB, relacionada com os objetos que estão sendo gerenciados sob o seu domínio. O RFC 1213 [4] define um conjunto padrão de variáveis utilizadas para o monitoramento e o controle de redes TCP/IP. Ainda assim, para cada novo dispositivo a ser gerenciado e que não tenha sido previsto o seu gerenciamento, deve ser definido um conjunto de novas variáveis que lhe sejam específicas e adequadas.

1.2.4 Protocolo de Gerenciamento de Redes

A forma de comunicação entre a estação de gerenciamento e o agente é definido pelo protocolo de gerenciamento de rede, no nosso caso, o SNMP [1]. O SNMP é definido como um protocolo de comunicação para informações gerenciais mas, também, corresponde a um conjunto de protocolos periféricos fundamentais para a sua definição. Como conjunto de protocolos, o SNMP é formado por:

- Um conjunto de especificações que determinam a estrutura e a identificação das informações gerenciais. Estas especificações estão dentro da SMI (*Structure of Management Information*) e são encontradas no RFC 1155 [5].
- Um protocolo de comunicação entre o gerente e o agente, o próprio SNMP (*Simple Network Management Protocol*). A especificação deste protocolo é apresentada no RFC 1157 [1].
- Uma base de informações gerenciais que especifica quais variáveis são mantidas pelos elementos de rede. Essa base de informações de gerenciamento é a MIB já descrita anteriormente.

1.3 RESUMO

Neste capítulo foram apresentados os elementos que compõem um sistema de gerenciamento de redes. Foram definidos o *gerente*, o *agente*, a MIB e o protocolo em si.

No próximos capítulos são apresentados o protocolo SNMP e a estrutura de uma MIB, que são a base para o desenvolvimento deste trabalho.

CAPÍTULO II – O PROTOCOLO SNMP

2.1 INTRODUÇÃO

SNMP é um protocolo desenvolvido para facilitar a leitura e escrita de pequenas quantidades de informação sobre uma rede, como simples números e *strings* de caracteres. O SNMP não é usado para grandes volumes de dados como, por exemplo, transferência de arquivos.

Um operação de leitura no SNMP é chamada de *GET*, ao passo que uma escrita é chamada *SET*. O destino de uma operação *GET* ou *SET* é chamada de um *objeto*. Um objeto é como um campo em um elemento de uma base de dados.

O SNMP trata objetos como se eles estivessem organizados em uma base de dados global conhecida como MIB. Uma maneira de facilitar o entendimento é imaginar o SNMP como sendo uma interface de programação de aplicativo (API) em um dispositivo. Ele esconde os detalhes da busca de objetos sob um espaço de nomes hierárquicos organizado. Um agente SNMP em cada sistema disponibiliza esta API. O *agente* é um processo ativado como um serviço do sistema e fica sempre disponível aguardando requisições SNMP na porta 161 do protocolo UDP. Sob a API, o agente pode interagir com o núcleo de um sistema ou com algum processo de aplicação que esteja rodando no sistema; ele pode também ler diversos trechos de informação de diferentes fontes e aplicar uma fórmula matemática para fornecer o valor definido em uma MIB. Um objeto na MIB pode também representar funcionalidade ao invés de dados; por exemplo, escrevendo em um objeto da MIB com um determinado valor pode reiniciar um determinado serviço [7].

Quando se fala em gerenciamento de redes, o SNMP é o protocolo padrão de fato. O SNMP proporciona monitoramento e gerenciamento remotos de redes desde o final dos anos 1980. É um protocolo com característica de fácil extensão, permitindo aos fabricantes incorporarem suas próprias funções de gerenciamento dentro de uma estrutura padronizada. Proporciona uma arquitetura muito simples e, em virtude da sua simplicidade, é difícil encontrar um equipamento de rede que não o implemente. Neste capítulo serão apresentados a arquitetura SNMP e o protocolo em si.

2.2 A ARQUITETURA DO SNMP

O SNMP segue o padrão de arquitetura cliente/servidor em que o Sistema de Gerenciamento de Redes (SGR) é o cliente para muitos agentes SNMP (servidores) distribuídos por toda a rede, como visto na Figura 2.

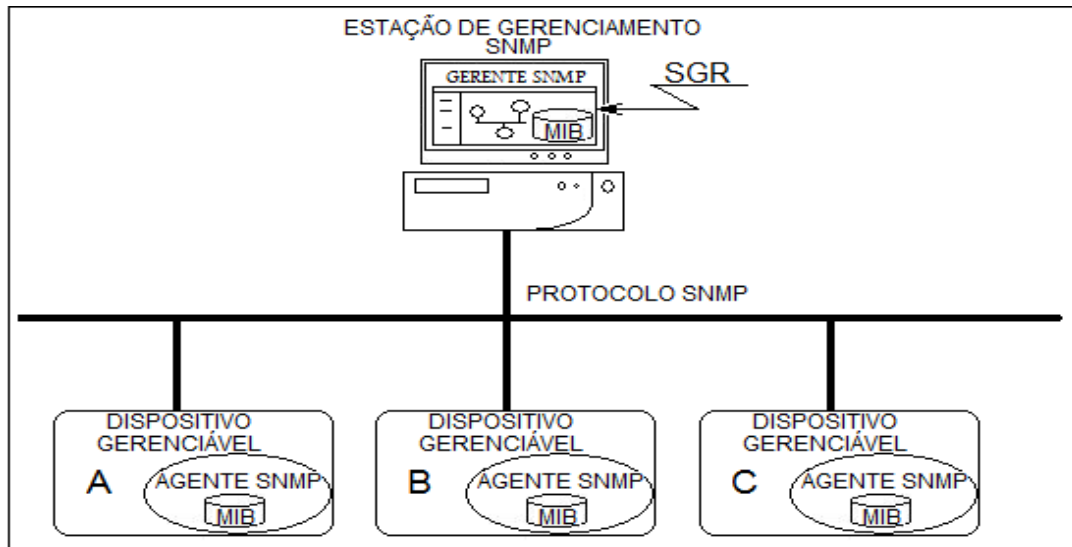


Figura 2. A Arquitetura SNMP.

Um dos problemas do gerenciamento de redes é o perpétuo aumento do número de equipamentos que devem ser gerenciados. Isto pode tornar-se um problema complexo até mesmo se os equipamentos forem similares. Sem o SNMP, o administrador da rede teria que monitorar e administrar manualmente cada um dos dispositivos. Com o SNMP, à medida que novos dispositivos são adicionados à rede, o SGR pode ser notificado e, então, rotineiramente passar a monitorar os seus parâmetros. Desta forma, erros ou falhas podem ser rapidamente identificados para que a integridade da rede seja preservada.

2.3 A ESTRUTURA DE DADOS DO SNMP

Talvez uma das grandes vantagens do SNMP seja a sua simplicidade. Um objeto dentro do SNMP é armazenado em uma árvore e organizado pelo seu identificador de objeto (OID—*Object Identifier*). Um OID é uma seqüência de números que identificam um objeto particular de forma universalmente única dentro da árvore. Por exemplo, uma coleção padrão de objetos dentro da MIB-II é o grupo *system*. Este grupo contém um número de objetos cujos valores são específicos para cada dispositivo que hospeda um

agente SNMP. Um desses objetos é a descrição do sistema (*system.SysDescr.0*). Este objeto é unicamente identificado pela seqüência ou OID *1.3.6.1.2.1.1.1.0*. Esta seqüência pode ser conferida através da árvore apresentada na Figura 3.

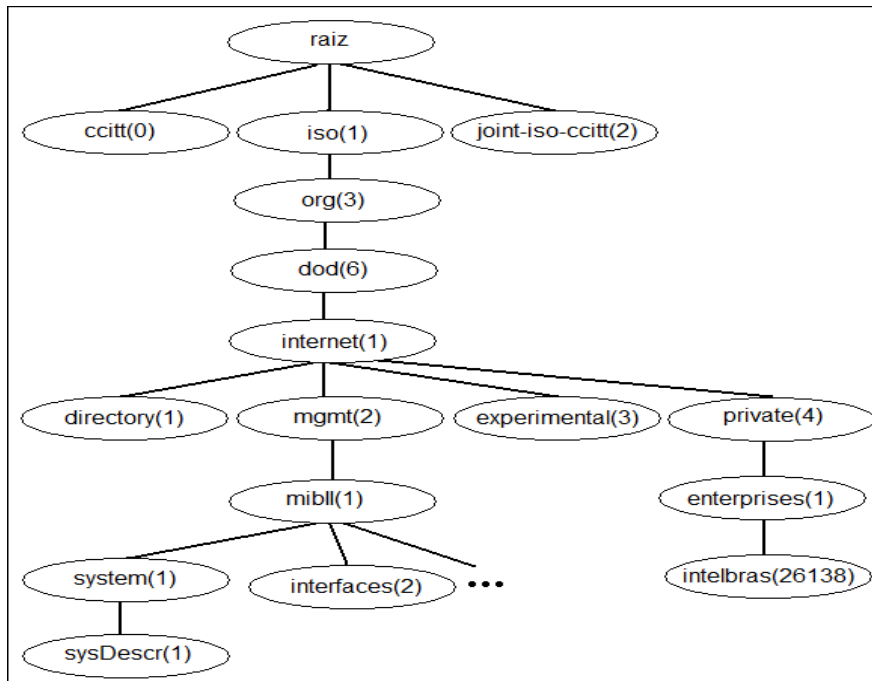


Figura 3. Amostra da Árvore da MIB.

O zero no final deste OID identifica esta variável como um escalar, ou seja, uma variável de instância única. Um escalar é um objeto que aparece apenas uma vez na árvore. Isso difere de objetos colunares que representam múltiplas instâncias de um objeto, tal como uma tabela de valores. Um zero na posição final representa uma “folha” (ou ponto final) da árvore.

Textualmente, a seqüência pode ser também representada como: *iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) system(1) sysDescr(1)*.

A árvore da MIB é muito ampla e profunda e cobre não somente dados de estações finais da rede mas também dados para equipamentos especializados tais como os roteadores da rede, *gateways* e *bridges*.

2.4 O PROTOCOLO SNMP

A versão 1 do SNMP proporciona cinco primitivas de comunicação básicas conhecidas como PDUs (*Protocol Data Units* ou Unidades de Dados do Protocolo). Dois tipos de PDUs são as requisições síncronas chamadas *GetRequest* e *GetNextRequest*.

As PDUs *GetRequest* tomam um identificador de objetos como parâmetro e resultam, como resposta, em uma PDU *GetResponse* do agente SNMP. A PDU *GetNextRequest* busca o objeto seguinte aquele usado como parâmetro, respeitando a ordem léxica da árvore de objetos, e novamente a resposta é uma PDU *GetResponse*. Estas PDUs implementam a função de leitura ou *GET*. O SNMP também fornece uma PDU assíncrona chamada *trap*. Uma *trap* é uma notificação de uma condição que é enviada diretamente para o SGR. Elas são usadas para notificar o SGR sobre eventos ou erros que não podem esperar até o próximo ciclo de interrogações.

Finalmente, a PDU *SetRequest* permite ao SGR mudar valores de objetos em um agente SNMP. Isso pode ser usado para comandar o agente SNMP para que ele execute certas ações. Esta PDU implementa a função de escrita ou *SET*. As Figuras 4 e 5 apresentam graficamente as primitivas de comunicação dentro da arquitetura.

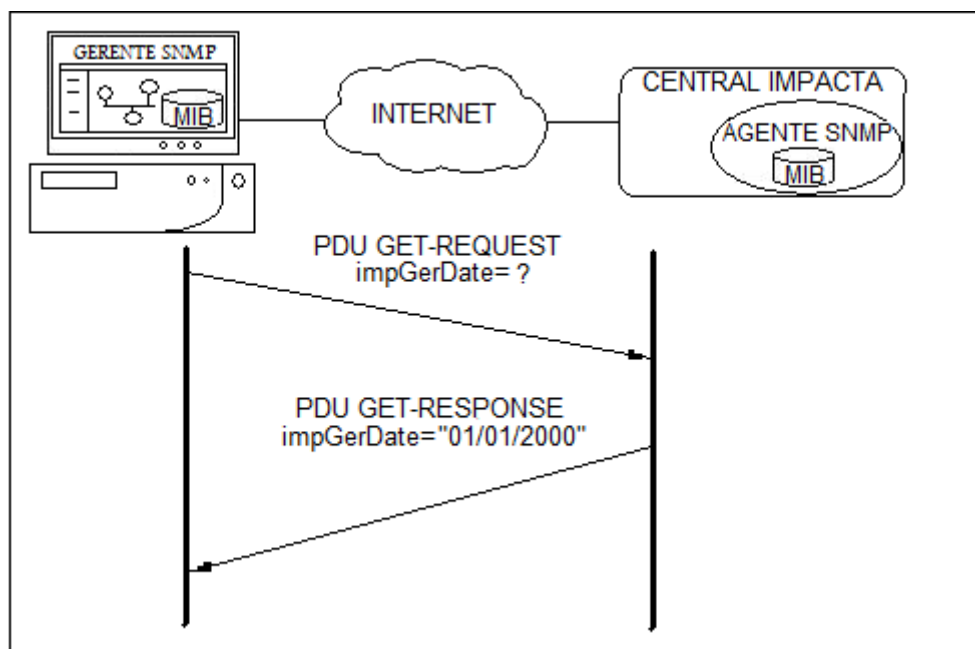


Figura 4. A PDU *GetRequest*.

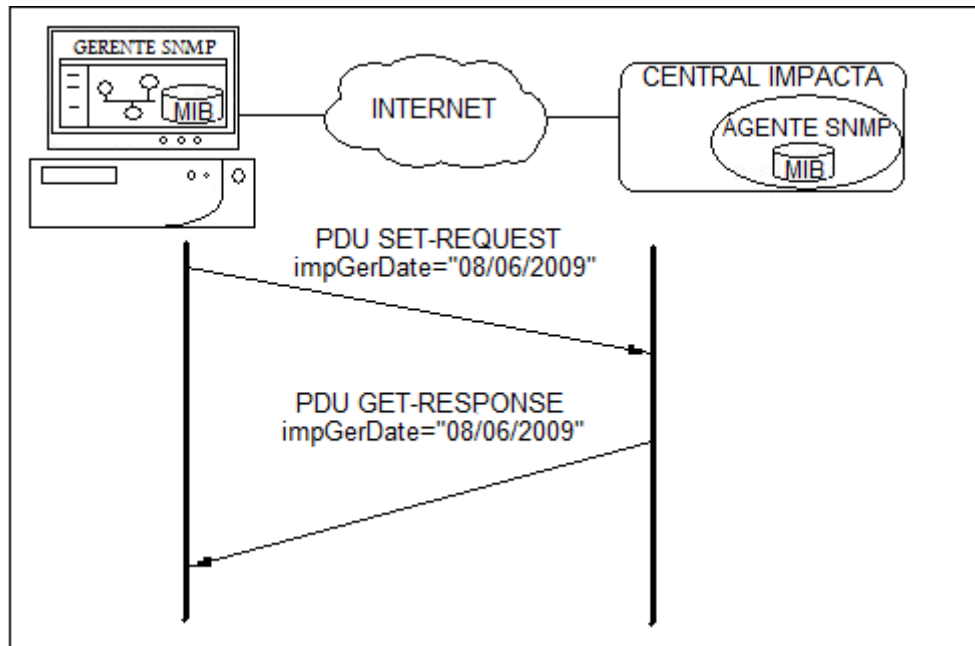


Figura 5. A PDU SetRequest.

Todas as PDUs seguem uma estrutura regular composta por tuplas Tipo-Tamanho-Valor (TTV). *Tipo* refere-se ao tipo de dados que forma o valor. *Tamanho* corresponde ao comprimento do “valor” em octetos e *Valor* é, efetivamente, a informação a ser transmitida. Uma PDU SNMP é formada por uma seqüência de TTVs em uma estrutura regular, como apresentado na Figura 6. Para simplificar a geração de respostas, o agente destino simplesmente traduz e preenche os dados requisitados como necessário. Isso é facilitado pelos TTVs nulos (NULL) que são colocados nas requisições no mesmo ponto em que os agentes deverão inserir dados nas respostas [9].

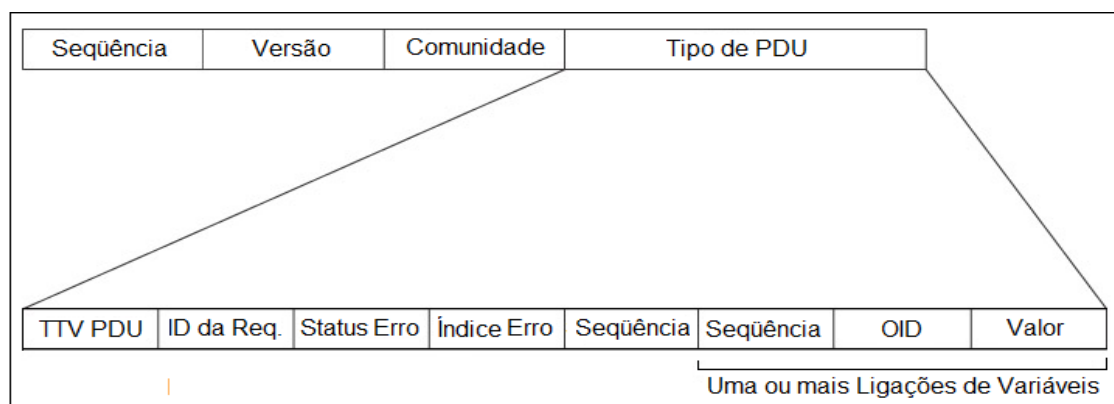


Figura 6. Formato de uma PDU SNMP.

Cada PDU pode tratar uma ou um conjunto de variáveis. Entretanto, deve-se ter em mente que o tamanho da resposta resultante não deve superar os 1472 bytes. Caso

isso aconteça, um erro é retornado ao gerente indicando que o pacote resultante ficou grande demais.

CAPÍTULO III – BASE DE INFORMAÇÕES DE GERENCIAMENTO - MIB

3.1 INTRODUÇÃO

Para facilitar o entendimento do que é uma MIB, dois conceitos preliminares serão apresentados: RFC e *Objetos Gerenciados*.

RFCs (*Request for Comments* ou Requisições para Comentários) são documentos que descrevem cada um dos protocolos da Internet antes de serem considerados padrões. Eles são lançados como uma definição preliminar e como uma requisição para que a comunidade da Internet possa analisar um determinado protocolo fazendo críticas e sugestões antes que ele se torne um padrão.

Objeto gerenciado é a visão abstrata de um recurso real do sistema. Assim, todos os recursos da rede que devem ser gerenciados são modelados e as estruturas de dados resultantes são os objetos gerenciados. Os objetos gerenciados podem ter permissões para serem lidos ou alterados, sendo que cada leitura representará o estado real do recurso e cada alteração também será refletida no próprio recurso.

O termo MIB possui diferentes significados baseados no seu contexto. Geralmente, uma MIB descreve informações que podem ser obtidas e/ou modificadas através de um protocolo de gerenciamento de redes. Assim, uma MIB é uma coleção de objetos gerenciados que são definidos a partir dos objetivos de gerenciamento traçados para um determinado elemento de rede. MIB também significa o ramo da árvore da Internet onde está definida a MIB padrão do TCP/IP.

A primeira versão da MIB foi definida através do RFC 1066 [6]. Esta MIB definiu a base de informação necessária para monitorar e controlar redes baseadas na pilha de protocolos TCP/IP. A proposta inicial desta MIB era ser um documento único passível de utilização tanto pelo SNMP como pelo CMIP. Com a evolução dos trabalhos sobre estes dois protocolos, a comunidade da Internet considerou que não era mais possível manter as definições de uma MIB unificada. Assim, foi criada a segunda versão da MIB, a MIB-II, através do RFC1213 e que é considerada uma evolução da MIB padrão original.

3.2 ESTRUTURA DAS INFORMAÇÕES DE GERENCIAMENTO

Além dos documentos que especificam as variáveis da MIB e os seus significados,

um padrão separado define um conjunto de regras usadas para criar e identificar variáveis de uma MIB. Estas regras são conhecidas como a especificação *Estrutura das Informações de Gerenciamento* (SMI – *Structure of Management Information*) [5]. Para simplificar o protocolo de gerenciamento de redes, a SMI coloca restrições nos tipos de variáveis permitidos na MIB, especifica as regras para nomear estas variáveis e cria regras para definir tipos de variáveis. Por exemplo, o padrão SMI inclui definição de termos como *IpAddress*, que é definido como sendo uma *string* de 4 octetos, e *Counter*, definido como um inteiro na faixa de 0 a $2^{32}-1$, e especifica que eles são termos usados para definir variáveis da MIB. Mais importante, as regras na SMI descrevem como a MIB faz referência à tabelas e variáveis (por exemplo, a *Tabela de Roteamentos IP*) [8].

Cada tipo de objeto (*OBJECT-TYPE*) possui um nome, uma sintaxe e uma codificação. O nome é representado unicamente como um identificador de objeto (*OBJECT-IDENTIFIER*). Um identificador de objeto é um nome atribuído administrativamente. As políticas administrativas usadas para atribuir nomes são discutidas posteriormente neste trabalho.

A sintaxe para um tipo de objeto define a estrutura de dados abstrata correspondente àquele tipo de objeto. Por exemplo, a estrutura de um dado tipo de objeto deve ser um inteiro (*INTEGER*) ou *string* de octetos (*OCTET STRING*). Embora em geral deva ser permitido que qualquer construção em *ASN.1* esteja disponível para uso na definição da sintaxe de um tipo de objeto, a SMI propositalmente restringe as construções em *ASN.1* que podem ser usadas. Estas restrições são feitas somente por questão de simplicidade.

A codificação de um tipo de objeto é simplesmente como as instâncias daquele tipo são representadas usando a sintaxe. Implicitamente vinculada à noção de uma codificação e sintaxe de um objeto, é como o objeto é representado quando está sendo transmitido na rede. Nos itens a seguir, são apresentadas as regras básicas de codificação do *ASN.1*.

3.2.1 Nomes

Os nomes são usados para identificar os objetos gerenciados. A SMI especifica nomes que são hierárquicos por natureza. O conceito de identificador de objeto (*OBJECT-IDENTIFIER* ou, simplesmente, *OID*) é usado para modelar esta notação. Um identificador

de objeto pode ser usado para outros propósitos que não a nomeação de tipos de objetos. Por exemplo, cada padrão internacional possui um identificador de objeto atribuído a ele com propósito de identificação. Resumindo, identificadores de objetos são um meio de identificar alguns objetos, independente da semântica associada ao objeto (por exemplo, um documento de padrão, um ramo da árvore etc.).

Um identificador de objetos é uma seqüência de inteiros que transpõe uma árvore global. A árvore consiste de uma raiz conectada a um número de nodos rotulados através de terminais. Cada nodo pode, por sua vez, ter seus próprios filhos que também são rotulados. Neste caso, podemos dizer que o nodo é uma subárvore. Este processo pode continuar até um nível arbitrário de profundidade. A idéia central da noção de identificadores de objetos é o entendimento de que o controle administrativo dos significados atribuídos aos nodos pode ser delegado à medida que se transpõe a árvore. Um rótulo para um nodo é um par formado por uma breve descrição textual e um inteiro associado.

O nodo raiz não possui rótulo e possui três nodos filhos ligados diretamente a ele: um nodo é administrado pela *ISO (International Standardization Organization)*, com o rótulo *iso(1)*; outro nodo é administrado pelo *CCITT (International Telegraph and Telephone Consultative Committee)*, com o rótulo *ccitt(0)*; e o terceiro nodo é administrado conjuntamente pela *ISO* e pelo *CCITT*, com o rótulo *joit-iso-ccitt(2)*.

Abaixo do nodo *iso(1)*, a *ISO* designou uma subárvore para uso por outras organizações internacionais: *org(3)*. Dos nodos filhos presentes, dois foram atribuídos para o Instituto Nacional de Padrões e Tecnologia dos Estados Unidos (*NIST – National Institutes of Standards and Technology*). Uma destas subárvores foi transferida pelo *NIST* para o Departamento de Defesa dos Estados Unidos, *dod(6)*.

O *DoD* alocou um nodo para a comunidade internacional a ser administrada pelo Quadro de Atividades da Internet (*IAB – Internet Activities Board*), como segue:

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

Isto significa que a subárvore de identificadores de objetos da Internet inicia com o prefixo: *1.3.6.1*.

A SMI, como um padrão aprovado pelo *IAB*, especifica também a política sob a qual esta subárvore de identificadores de objetos é administrada. Neste nível da árvore,

quatro nodos estão presentes:

```

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

```

3.2.1.1 A Subárvore *directory*

A subárvore *directory*(1) é reservada para uso com um documento que discute como o diretório OSI pode ser usado na Internet.

3.2.1.2 A Subárvore *mgmt*

A subárvore *mgmt*(2) é usada para identificar objetos que são definidos em documentos aprovados pelo IAB. A administração da subárvore *mgmt*(2) é delegada pelo IAB à Autoridade de Números Atribuídos da Internet (IANA – *Internet Assigned Numbers Authority*). À medida que RFCs que definem novas versões da MIB padrão são aprovadas, a elas são atribuídos identificadores de objetos pelo IANA para a especificação dos objetos nelas definidos.

Por exemplo, à RFC que define a MIB padrão da Internet (MIB II) foi atribuído o documento de gerenciamento número 1. Assim, esta RFC usa o identificador de objeto

```
{ mgmt 1 }
```

ou

```
1.3.6.1.2.1
```

ao definir a MIB II.

3.2.1.3 A Subárvore *experimental*

A subárvore *experimental*(3) é usada para identificar objetos usados em experimentos da Internet. A administração desta subárvore é delegada pelo IAB à IANA.

Por exemplo, um experimentador poderia receber o número 17 e teria disponível o identificador de objeto

```
{ experimental 17 }
```

ou

```
1.3.6.1.3.17
```

para seu uso.

Como parte do processo de atribuição, a *IANA* pode fazer exigências sobre como aquela subárvore deve ser usada.

3.2.1.4 A Subárvore *Private*

A subárvore *private(4)* é usada para identificar objetos definidos unilateralmente. A administração da subárvore *private(4)* é delegada pelo *IAB* à *IANA* para a comunidade da Internet. Esta subárvore possui pelo menos um filho:

enterprises OBJECT IDENTIFIER ::= { private 1 }

A subárvore *enterprises(1)* é usada, entre outras coisas, para permitir a empresas que fornecem subsistemas de rede registrar modelos dos seus produtos.

Ao receber uma subárvore, a empresa pode, por exemplo, definir novos objetos da MIB na subárvore. Em adição, é extremamente recomendado que a empresa registre seus subsistemas de rede sob esta subárvore, de modo a proporcionar um mecanismo de identificação sem ambigüidade para uso em protocolos de gerenciamento. Por exemplo, a empresa “Intelbrás S/A” requereu um nodo sob a subárvore *enterprises* da *IANA*. Tal nodo recebeu o número:

1.3.6.1.4.1.26138

Dentro da “Intelbrás S/A”, que apresenta mais de uma linha de produtos, as centrais telefônicas (*pabx*) está registradas sob a subárvore 1 e sob esta, as centrais Impacta, definidas pela subárvore 1, com o nome:

1.3.6.1.4.1.26138.1.1

Assim, cada um dos objetos definidos na MIB das centrais Impacta tem este prefixo de forma a serem identificados de maneira universalmente única.

3.2.2 Sintaxe

A sintaxe é usada para definir o tipo de um determinado objeto. Tipos definidos na *ASN.1* são usados na declaração da sintaxe de um objeto, embora apenas um subconjunto de toda a gama de tipos definidos na *ASN.1* seja permitido.

Os tipos *ASN.1* podem ser classificados em três grupos: tipos primitivos, tipos construtores e tipos definidos.

3.2.2.1. Tipos Primitivos

Somente os tipos primitivos *ASN.1 INTEGER*, *OCTET STRING*, *OBJECT-IDENTIFIER* e *NULL* são permitidos. Estes tipos são algumas vezes referenciados como tipos “não agregados”.

Para um objeto definido como um inteiro enumerado, o valor 0 (zero) não deverá estar presente na lista de enumerações. O uso deste valor é proibido.

3.2.2.2. Tipos Construtores

O tipo construtor *ASN.1 SEQUENCE* é permitido, desde que seja usado para gerar tanto listas como tabelas.

Para listas, a sintaxe assume a forma:

```
SEQUENCE { <tipo1>, ..., <tipoN> }
```

onde cada <tipok> corresponde a um dos tipos primitivos listados no item anterior.

Para tabelas, a sintaxe assume a forma:

```
SEQUENCE OF <entrada>
```

onde <entrada> é resolvida para uma lista de construtores.

Listas e tabelas são algumas vezes referenciadas como tipos agregados.

3.2.2.3. Tipos Definidos

Novos tipos podem ser definidos desde que sejam muito utilizados, bastando para isso defini-los sobre tipos primitivos, listas, tabelas ou algum outro tipo definido anteriormente pelo criador de uma MIB.

- **NetworkAddress**

Este tipo representa um endereço de uma dentre muitas famílias de protocolo. No início, apenas uma família de protocolos foi implementada: a família Internet.

- **IpAddress**

Este tipo largamente empregado representa um endereço Internet de 32 *bits*. Ele é representado como um *string* de octetos de tamanho 4, na ordem de *bytes* da rede.

Quando este tipo *ASN.1* é codificado usando as regras básicas de codificação, somente o documento primitivo deve ser usado.

- **Counter**

Este tipo representa um inteiro não negativo que incrementa monotonicamente até alcançar um valor máximo e, então, salta para o início, continuando o incremento a partir do zero novamente.

- **Gauge**

Este tipo representa um número não negativo que pode ser incrementado ou decrementado mas que é limitado a um valor máximo. Para a versão 1 do SNMP este valor máximo é $2^{32}-1$ (4294967295 em decimal).

- **TimeTicks**

Este tipo representa um número inteiro não negativo que conta o tempo em centésimos de segundo desde uma época considerada. Quando um objeto com este tipo é definido em uma MIB, a descrição deste objeto identifica a época de referência.

- **Opaque**

Este tipo dá suporte à capacidade de passar uma sintaxe *ASN.1* arbitrária. Um valor é codificado usando as regras básicas de *ASN.1* em um *string* de octetos. Isso, então, é codificado com um *OCTET STRING*, em efeito “encapsulando duplamente” o valor *ASN.1* original.

Note que uma implementação precisa apenas ser capaz de aceitar e reconhecer dados codificados opacamente. Ele não precisa ser capaz de desencapsular os dados e então interpretar o seu conteúdo.

Além disso, note que pelo uso do tipo *ASN.1 EXTERNAL*, outras codificações diferentes de *ASN.1* podem ser usadas em dados codificados opacamente.

3.2.3. CODIFICAÇÕES

Uma vez que uma instância de um tipo de objeto tenha sido identificada, o seu valor pode ser transmitido aplicando-se as regras de codificação básica do *ASN.1* à

sintaxe para o tipo de objeto.

3.3. OBJETOS GERENCIADOS

Uma definição de tipo de objeto consiste de cinco campos:

OBJECT:

Um nome textual chamado de descritor do objeto (*OBJECT DESCRIPTOR*), para o tipo de objeto, junto como o seu identificador de objeto (*OBJECT-IDENTIFIER*).

SYNTAX:

A sintaxe abstrata para o tipo de objeto. Deve ser resolvida para uma instância do tipo *ObjectSyntax* que será definido na seqüência.

ACCESS:

Uma das opções somente leitura (*read-only*), leitura e escrita (*read-write*), somente escrita (*write-only*) ou não acessível (*not-accessible*).

STATUS:

Uma das opções obrigatório (*mandatory*), opcional (*optional*) ou obsoleto (*obsolete*). Se for obrigatório, o objeto tem que fazer parte da implementação. Se for opcional, cabe ao implementador a escolha de incluir o objeto ou não e, se for obsoleto, o objeto não deve ser implementado.

DESCRIPTION:

Uma descrição textual da semântica do tipo de objeto. As implementações devem assegurar que suas instâncias do objeto preencham esta definição uma vez que estas definições serão usadas em MIBs de múltiplos ambientes de fabricantes diferentes. Como tal, é vital que objetos possuam significados consistente através de todas as máquinas.

3.3.1. Orientação para Nomes de Objetos

Nenhum tipo de objeto na MIB padrão da Internet deve usar o zero como parte do seu identificador de objetos. Este valor foi reservado para uso em futuras extensões.

Cada descritor de objeto correspondente a um tipo de objeto na MIB deverá ser único e ser um *string* mnemônico passível de impressão. Esta restrição promove uma linguagem comum para humanos, para usar quando a MIB estiver em discussão, e também para facilitar mapeamento de tabelas para interface de usuário.

3.3.2. Instâncias e Tipos de Objetos

Um tipo de objeto é a definição de uma classe de objetos gerenciados; é declarativo por natureza. Em contraste, uma instância de objeto é uma instanciação de um tipo de objeto que foi associada a um valor. Por exemplo, a noção de uma entrada em uma tabela de roteamento pode ser definida na MIB. Tal noção corresponde a um tipo de objeto; entradas individuais em uma tabela de roteamento particular que existe em um determinado momento são instâncias de objeto daquele tipo de objeto.

Uma coleção de tipos de objeto é definida na MIB. Cada um destes tipos de objeto é nomeado unicamente pelo seu identificador de objeto e também possui um nome textual, que é seu descritor de objeto. Os meios pelos quais instâncias de objeto são referenciadas não estão definidos na MIB. Referência a instâncias de objeto são alcançadas por um mecanismo específico do protocolo: é responsabilidade de cada protocolo de gerenciamento aderir ao SMI para definir este mecanismo.

Um tipo de objeto pode ser definido na MIB de forma tal que uma instância daquele tipo de objeto representa uma agregação de informação também representada por instâncias de algum número de tipos de objetos “subordinados”. Por exemplo, suponha que os seguintes tipos de objeto estejam definido na MIB:

```

atIndex OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write.
    STATUS  mandatory.
    DESCRIPTION
        "O número da interface para o endereço físico."
    ::= { atEntry 1 }

atPhysAddress OBJECT-TYPE
    SYNTAX  OCTET STRING
    ACCESS  read-write.
    STATUS  mandatory.
    DESCRIPTION
        "O endereço físico dependente do meio físico utilizado na interface."
    ::= { atEntry 2 }

atNetAddress OBJECT-TYPE
    SYNTAX  NetworkAddress
    ACCESS  read-write.

```

```

STATUS mandatory.
DESCRIPTION
  "O endereço de rede correspondente ao endereço físico dependente
  do meio."
 ::= { atEntry 3 }

```

Assim, um quarto tipo de objeto pode também ser definido na MIB:

```

atEntry OBJECT-TYPE
  SYNTAX NetworkAddress
  ACCESS read-write.
  STATUS mandatory.
  DESCRIPTION
    "O endereço de rede correspondente ao endereço físico dependente do
    meio."
 ::= { atTable 1 }

```

```

atEntry OBJECT-TYPE
  SYNTAX AtEntry
  ACCESS not-accessible
  STATUS deprecated
  DESCRIPTION
    "Cada entrada contém uma equivalência entre os endereços de rede
    (NetworkAddress) e o endereço físico."
  INDEX { atIfIndex, atNetAddress }
 ::= { atTable 1 }

```

onde

```

AtEntry ::=
  SEQUENCE {
    atIfIndex
      INTEGER,
    atPhysAddress
      PhysAddress,
    atNetAddress
      NetworkAddress
  }

```

Cada instância deste tipo de objeto compreende informações representadas por instâncias dos três tipos de objetos apresentados anteriormente. Um tipo de objeto definido desta forma é chamado de *lista*.

Da mesma forma, tabelas podem ser formada por agregação de uma lista de tipos. Por exemplo, um quinto tipo de objeto também pode ser definido na MIB:

```

atTable OBJECT-TYPE
  SYNTAX SEQUENCE OF AtEntry
  ACCESS not-accessible
  STATUS deprecated
  DESCRIPTION
    "As tabelas de tradução de endereços contém as equivalências entre os
    endereços de rede e endereços físicos. Algumas interfaces não utilizam
    tabelas de tradução para determinar equivalência de endereços (por

```

exemplo, DDN-X.25 possui um método algorítmico); se todas as interfaces forem deste tipo então a tabela de tradução de endereços fica vazia, ou seja, possui zero entradas.”
 ::= { at 1 }

de modo que cada instância de um objeto *atTable* compreende a informação representada pelo conjunto de tipos de objeto *atEntry* que coletivamente constitui uma dada instância do objeto *atTable*, ou seja, uma dada tabela de tradução de endereços.

Considere como alguém poderia referenciar um simples objeto dentro de uma tabela. Continuando com o exemplo anterior, pode-se nomear o tipo de objeto

```
{ atPhysAddress }
```

e especificar, usando um mecanismo específico de protocolo, a instância de objeto

```
{ atNetAddress } = { internet "10.0.0.52" }
```

Este par de tipo de objeto com instância de objeto se referiria a todas as instâncias do *atPhysAddress* que é parte de qualquer entrada de alguma tabela de tradução de endereços para a qual o valor de *atNetAddress* associado é { internet “10.0.0.52” }.

Para continuar com este exemplo, considere como poderia ser referenciado um objeto agregado (lista) dentro da tabela. Nomeando o tipo de objeto

```
{ atEntry }
```

e especificando a instância de objeto

```
{ atNetAddress } = { internet "10.0.0.52" }
```

refere-se a todas as instâncias de entradas na tabela para as quais o valor do *atNetAddress* associado é { internet “10.0.0.52” }.

Cada protocolo de gerenciamento deve prover um mecanismo para acessar tipos de objetos simples (não agregados). Cada protocolo de gerenciamento especifica se suporta ou não acesso a tipos de objetos agregados. Além disso, o protocolo deve especificar quais instâncias são “retornadas” quando um par tipo de objeto/instância refere-se a mais de uma instância de um tipo.

Para assegurar suporte a uma variedade de protocolos de gerenciamento, toda a informação pela qual instâncias de um determinado tipo de objeto possa ser distinguido um do outro é representada por instâncias dos tipos de objetos definidos na MIB.

CAPÍTULO IV – A MIB IMPACTA

4.1 INTRODUÇÃO

O projeto do gerenciamento de um elemento de rede através do SNMP pode ser dividido em três etapas bem definidas: a criação da MIB, a incorporação da MIB ao agente SNMP e a incorporação da MIB ao sistema de gerenciamento.

A criação da MIB é um trabalho de modelagem de um dispositivo baseado nas suas características operacionais interessantes do ponto de vista do seu gerenciamento. Exige do projetista um conhecimento do funcionamento básico deste equipamento e um senso de organização que permita a criação de uma estrutura simples mas completa.

A incorporação da MIB ao agente SNMP é o processo de criar cada uma das variáveis de uma MIB dentro do agente. Isso proporciona as condições para que o agente possa responder às solicitações de um gerente relativas àquelas variáveis. Além disso, criar as variáveis no agente significa implementar os mecanismos de ação sobre o sistema, correspondentes a cada operação de protocolo sobre as variáveis da MIB.

A incorporação da MIB ao sistema de gerenciamento é feita através da compilação da MIB pelo aplicativo utilizado para o gerenciamento. Os *softwares* de gerenciamento possuem compiladores de MIB que lhes permitem acrescentar a subárvore de variáveis de um determinado equipamento ao seu sistema.

O foco deste trabalho foi a modelagem das centrais Impacta para o gerenciamento SNMP, ou seja, a criação da sua MIB e a sua incorporação ao agente *Net-SNMP*. Este processo de criação é descrito neste capítulo.

4.2 CRIAÇÃO DA MIB IMPACTA

4.2.1 As Centrais Impacta

A linha de centrais telefônicas Impacta foi criada com características híbridas para atender tanto ambientes corporativos de grande como de pequeno porte. O grande atrativo destas centrais é a possibilidade de conexão via VoIP (*Voice over IP*, ou Voz sobre o protocolo IP) que permite a interconexão remota entre duas ou mais centrais, numa configuração ponto-a-ponto.

Por serem híbridas, suportam troncos analógicos e troncos digitais de forma

configurável. Também disponibilizam ramais analógicos para a conexão de aparelhos telefônicos convencionais, máquinas de *fax* e *modems*, e ramais digitais para a conexão de terminais inteligentes (*T/s*), que permitem configurar e operar as centrais utilizando toda a gama de serviços disponibilizados ao usuário.

A linha de centrais Impacta é apresentada em 4 modelos:

Impacta16:

- Configuração mínima: 2 troncos analógicos e 4 ramais (analógicos e/ou digitais)
- Configuração máxima: 4 troncos analógicos, 12 ramais (analógicos e/ou digitais)

Impacta68:

- Configuração mínima: 2 troncos analógicos e 4 ramais (analógicos e/ou digitais)
- Configuração máxima:
 - primeira opção - 30 troncos digitais, 6 troncos analógicos, 32 ramais (24 analógicos e 8 digitais);
 - segunda opção - 8 troncos analógicos, 32 ramais (24 analógicos e 8 digitais).

Impacta140:

- Configuração mínima: 8 troncos analógicos e 16 ramais (analógicos e/ou digitais)
- Configuração máxima:
 - primeira opção - 60 troncos digitais, 8 troncos analógicos, 48 ramais (analógicos e/ou digitais);
 - segunda opção - 60 troncos digitais e 80 ramais (analógicos e/ou digitais).

Impacta220:

- Configuração mínima: 8 troncos analógicos e 16 ramais (analógicos e/ou digitais)

- Configuração máxima:
 - primeira opção - 60 troncos digitais, 8 troncos analógicos, 128 ramais (analógicos e/ou digitais);
 - segunda opção - 60 troncos digitais e 160 ramais (analógicos e/ou digitais).

4.2.2 O Gerenciamento das Centrais Impacta

O Gerenciamento das centrais Impacta atualmente é feito através do *software Programador IMPACTA*. Este *software* foi desenvolvido pela equipe da Intelbrás e permite uma configuração completa das centrais. A Figura 7 apresenta a tela de configuração de ramais do *Programador IMPACTA*.

O *Programador IMPACTA* é compatível com todas as opções de centrais Impacta, permitindo a geração de relatórios e gráficos que facilitam o controle total do sistema telefônico. Recebe informações de chamadas efetuadas pelo ramal como tempo, duração, número discado, preço da ligação etc. Além da característica de monitoramento da central, o *Programador IMPACTA* também permite a configuração operacional de todo o sistema telefônico, priorizando ramais, definindo grupos, configurando conexões VoIP etc. O sistema formado pelo *software Programador IMPACTA* mais o micro que o hospeda é conectado à central através de uma linha serial RS232.

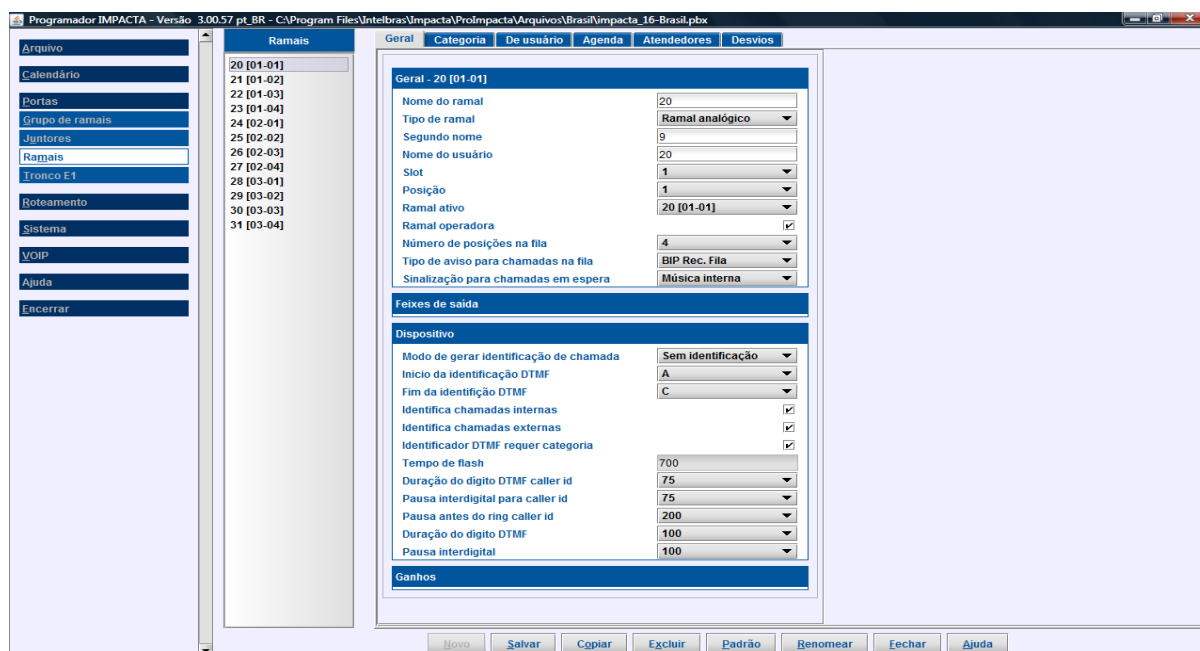


Figura 7. Tela de Configuração de Ramais - Programador IMPACTA

Embora favoreça o aspecto de segurança de acesso à central, a conexão RS232 é limitada em sua velocidade e não permite gerenciamento remoto. Isso porque a máquina que gerencia tem que estar próxima desta central telefônica. Este aspecto dificulta a inclusão das centrais Impacta em um sistema de gerenciamento centralizado.

A criação da MIB Impacta e do agente SNMP que a implementa, tem o principal objetivo de permitir o gerenciamento remoto e centralizado das centrais Impacta tanto no aspecto de sua configuração quanto no aspecto de seu monitoramento operacional.

Embora a criação desta MIB tenha a meta de ser a mais completa possível, atendendo a todas as características das centrais Impacta, algumas características gerenciáveis destas centrais são de difícil modelagem dentro de uma MIB SNMP. Isso porque exigem referências cruzadas entre tabelas, o que torna mais complexa a implementação do agente e até mesmo a utilização do gerenciamento. Na primeira versão da MIB Impacta, tais características foram deixadas disponíveis apenas para o sistema de gerenciamento convencional da central.

4.2.3 As Variáveis de Gerenciamento

As variáveis de gerenciamento definidas nesta versão 1.0 da MIB Impacta estão focadas no monitoramento e configuração dos ramais e dos troncos (ou juntores) das centrais Impacta. Isso porque algumas características do equipamento não são facilmente modeláveis em uma MIB e, também, porque existem funções que estão sendo redefinidas em um novo projeto da placa VoIP, atualmente em execução. Para a criação das variáveis da MIB o *Programador IMPACTA* foi utilizado como referência.

Sob o ramo *enterprises* da MIB, foi definida a subárvore Intelbrás (26138) a ser administrada pela empresa (ver Figura 3). Abaixo do nodo *Intelbrás* foi definido um nível para as linhas de produtos da empresa onde os *pabx* correspondem à subárvore 1. Abaixo do nodo *pabx* são definidas as linhas de centrais gerenciáveis que a Intelbrás produz ou produzirá no futuro. A linha Impacta é definida pela subárvore 1. A MIB Impacta foi, por sua vez, dividida em 3 subárvores, que correspondem aos grupos de variáveis definidos: *geral(1)*, *ramais(2)* e *juntores(3)*. A Figura 8 apresenta a subárvore das centrais Impacta a partir do nodo *private*.

No grupo *geral*, encontram-se as variáveis relativas a toda a central: descrição do sistema, data e hora, modelo da central sendo gerenciada, número máximo de ramais e

número máximo de jutores para este modelo etc.

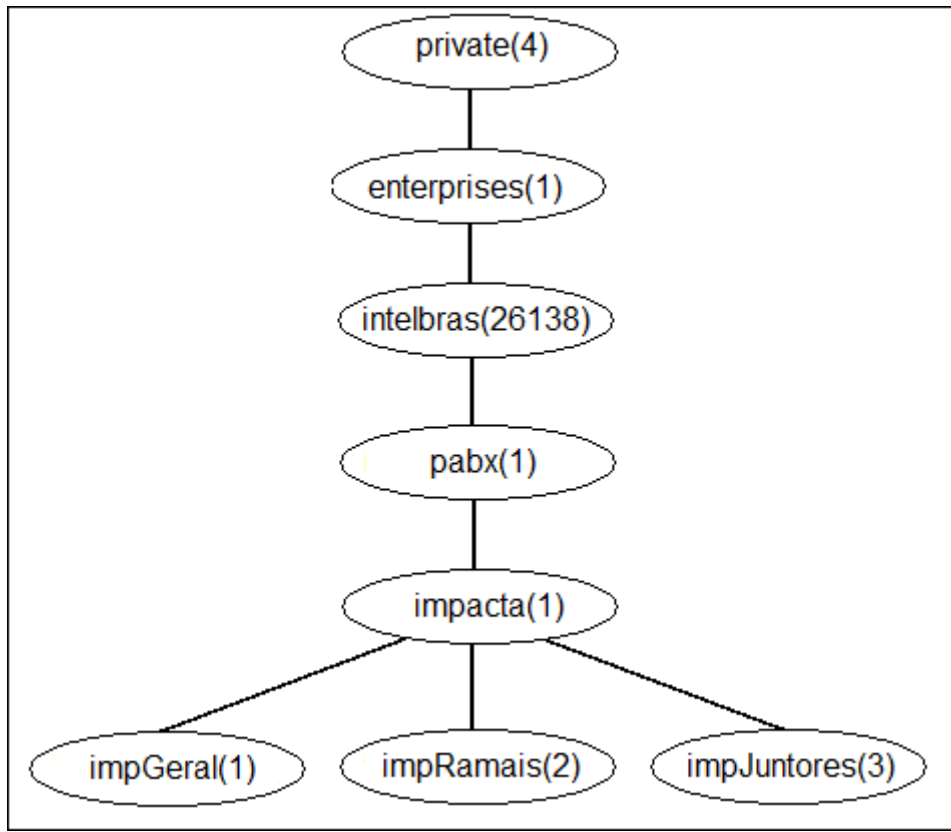


Figura 8. Subárvore Impacta

No grupo *ramais* é definida uma variável de instância única para informar o número de ramais efetivamente instalados na central e uma tabela em que cada entrada corresponde a um determinado ramal. Para cada ramal são definidas 27 variáveis para a sua configuração e monitoramento. Estas variáveis indicam se o ramal está presente ou não, a sua localização no sistema, o seu nome, a sua categoria, o seu tipo etc.

No grupo *jutores*, também é definida uma variável de instância única para informar o número de jutores efetivamente instalados na central e uma tabela onde cada entrada corresponde a um determinado jutor (ou tronco). Para cada tronco são definidas 14 variáveis que permitem determinar se o tronco está presente ou não, qual a direção do tronco, qual o tipo de atendimento, o tipo de discagem, tipo de identificação etc.

O documento formal resultante é apresentado no item 4.2.4.

4.2.4 A MIB Impacta

A MIB Impacta apresentada a seguir foi validada pela ferramenta *SimpleWeb MIB Module Validation* da universidade de Twente, na Holanda, disponibilizada na internet pelo endereço “<http://wwwsnmp.cs.utwente.nl/ietf/mibs/validate/>” (Figura 9).

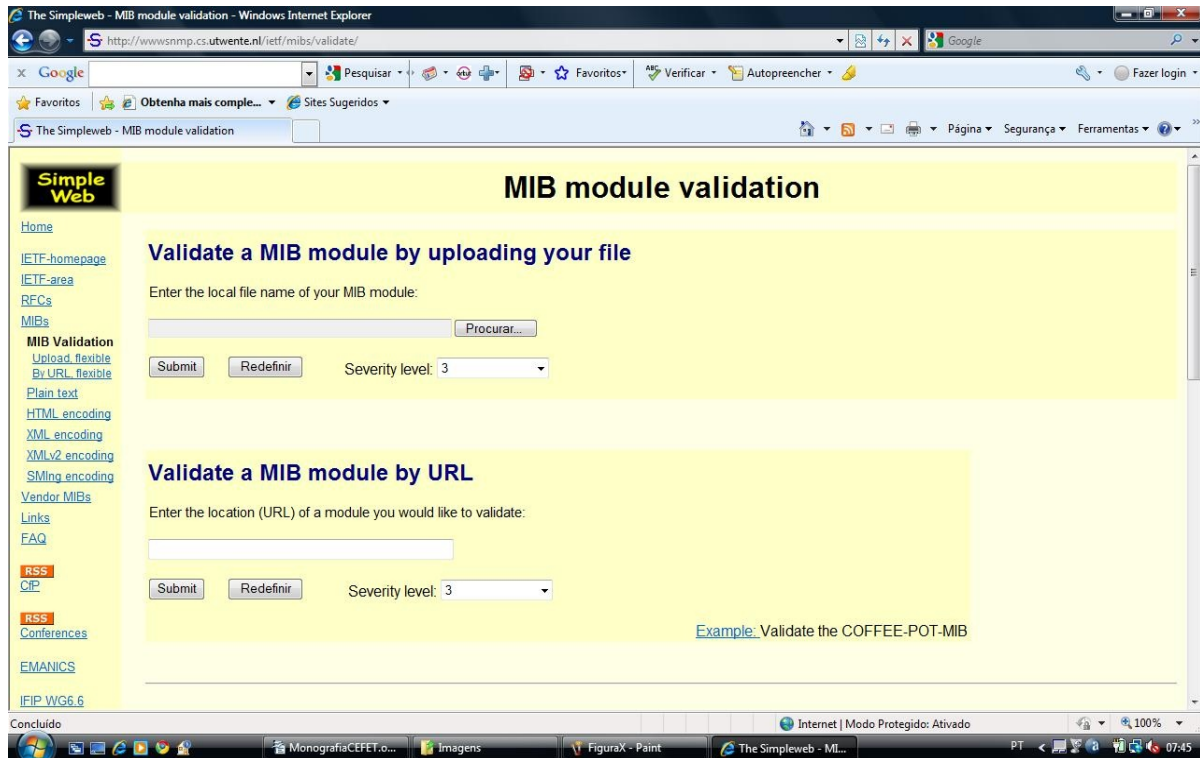


Figura 9. A Ferramenta *SimpleWeb*

A validação da MIB visa eliminar erros antes de submetê-la ao sistema de gerenciamento e verificar se a sua implementação está dentro do padrão.

```
-- MIB-IMPACTA { iso org(3) dod(6) internet(1) private(4) enterprises(1) 26138 }

MIB-IMPACTA DEFINITIONS ::= BEGIN
-- Título: MIB do PABX Impacta da Intelbrás
-- Versão: 1.0.0
-- Data: 25/02/2008
-- Por: Pedro Paulo da Silva (mat: 043210)
-- Comentários: Desenvolvida para a Intelbrás S/A
--

IMPORTS
MODULE-IDENTITY, enterprises
FROM SNMPv2-SMI
DisplayString, PhysAddress
```

```

FROM RFC1213-MIB
OBJECT-TYPE      FROM RFC1155-SMI;

-- Alguns Compiladores de MIB necessitam destas duas linhas:
-- enterprises OBJECT IDENTIFIER ::=
--             { iso org(3) dod(6) internet(1) private(4) 1 }

intelbras MODULE-IDENTITY
  LAST-UPDATED "200905300000Z"
  ORGANIZATION "www.intelbras.com.br"
  CONTACT-INFO
    "postal:    BR 101, km 100
      Sao Jose - SC

      email:    pedro.paulo@intelbras.com.br"
  DESCRIPTION
    "Definicoes da MIB Impacta"
  REVISION     "200905300000Z"
  DESCRIPTION
    "Primeira Versao"
  ::= { enterprises 26138}

pabx          OBJECT IDENTIFIER ::= { intelbras 1 }
impacta       OBJECT IDENTIFIER ::= { pabx 1 }

impGeral      OBJECT IDENTIFIER ::= { impacta 1 }
impRamais     OBJECT IDENTIFIER ::= { impacta 2 }
impJuntores   OBJECT IDENTIFIER ::= { impacta 3 }

-- Informações Gerais Relacionadas ao PABX Impacta da Intelbrás.
--   impGeral OBJECT IDENTIFIER ::= { impacta 1 }

impGerSysDescr OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..255))
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Uma descricao do Agente SNMP do PABX Impacta da Intelbras, incluindo
    versao/revisao do software."
  ::= { impGeral 1 }

impGerDate OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..79))
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "A data atual mantida pelo relógio de tempo real do agente SNMP do PABX
    Impacta. Esta data eh fornecida no formato dd/mm/aaaa."
  ::= { impGeral 2 }

impGerTime OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..79))
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "A hora atual mantida pelo relógio de tempo real do agente SNMP do PABX
    Impacta. Esta hora eh fornecida no formato hh:mm:ss."
  ::= { impGeral 3 }

impGerModelo OBJECT-TYPE
  SYNTAX INTEGER {
    impacta16(1),
    impacta68(2),
    impacta140(3),
    impacta220(4)
  }

```

```

MAX-ACCESS read-only
STATUS current
DESCRIPTION
"O modelo de PABX Impacta que esta entidade representa."
 ::= { impGeral 4 }

impGerNumMaximoRamais OBJECT-TYPE
SYNTAX INTEGER {
    r12(1),
    r32(2),
    r80(3),
    r160(4)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Descreve o numero maximo de ramais dependendo do modelo de PABX
utilizado: 12 ramais para Impacta16, 32 ramais para Impacta68,
80 ramais para Impacta140 e 160 ramais para Impacta220."
 ::= { impGeral 5 }

impGerNumMaximoJuntores OBJECT-TYPE
SYNTAX INTEGER {
    j4(1),
    j36(2),
    j68(3)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Descreve o numero maximo de juntores dependendo do modelo de PABX
utilizado: 4 juntores para Impacta16, 36 juntores para Impacta68 e
68 juntores para os modelos Impacta140 e Impacta220."
 ::= { impGeral 6 }

-- InformaÃ§Ãµes relacionadas aos ramais dos PABX Impacta.
-- impRamais OBJECT IDENTIFIER ::= { impacta 2 }

impRamaisInstalados OBJECT-TYPE
SYNTAX INTEGER (4..160)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Descreve o numero de ramais instalados neste PABX. Este numero eh,
no minimo 4 e no maximo o valor da variavel impGerNumMaximoRamais."
 ::= { impRamais 1 }

impRamaisTable OBJECT-TYPE
SYNTAX SEQUENCE OF ImpRamaisEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Uma tabela contendo, no maximo, 128 entradas (128 ramais do modelo
Impacta220), que pode ser esparsa e que eh usada para acessar
variaveis dos ramais do PABX."
 ::= { impRamais 2 }

impRamaisEntry OBJECT-TYPE
SYNTAX ImpRamaisEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Uma entrada da tabela que representa um ramal do PABX. Um
determinado ramal do PABX eh identificado de forma inequivoca
atraves do parametro impRTIndex."
INDEX { impRTIndex }

```

```

 ::= { impRamaisTable 1 }

ImpRamaisEntry ::=
 SEQUENCE {
   impRTIndex
     INTEGER,
   impRTPresent
     INTEGER,
   impRTNome
     INTEGER,
   impRTTipo
     INTEGER,
   impRTSlot
     INTEGER,
   impRTPosicao
     INTEGER,
   impRTModoIdentDeChamada
     INTEGER,
   impRTInicioIdentDTMF
     INTEGER,
   impRTFinalIdentDTMF
     INTEGER,
   impRTIdentChamadasInternas
     INTEGER,
   impRTIdentChamadasExternas
     INTEGER,
   impRTCategoriaParaIdentDTMF
     INTEGER,
   impRTTempoDeFlash
     INTEGER,
   impRTDuracaoDigitoDTMFIdent
     INTEGER,
   impRTPausaInterdigitalIdent
     INTEGER,
   impRTPausaAntesDoRingIdent
     INTEGER,
   impRTDuracaoDigitoDTMF
     INTEGER,
   impRTPausaInterdigital
     INTEGER,
   -----
   -- Categoria
   -----
   impRTChamadaInternaFazRecebe
     INTEGER,
   impRTAcessoExternoInternacional
     INTEGER,
   impRTAcessoExternoNacional
     INTEGER,
   impRTAcessoExternoRegional
     INTEGER,
   impRTAcessoExternoLocal
     INTEGER,
   impRTCelularExternoInternacional
     INTEGER,
   impRTCelularExternoNacional
     INTEGER,
   impRTCelularExternoRegional
     INTEGER,
   impRTCelularExternoLocal
     INTEGER
 }

impRTIndex OBJECT-TYPE
 SYNTAX INTEGER (1..4096)
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

```

```

"Indice que identifica uma entrada da tabela. Esta variavel podera
assumir valores de 1 ateh impRamaisInstalados."
::= { impRamaisEntry 1 }

impRTPresent OBJECT-TYPE
    SYNTAX INTEGER {
        true(1),
        false(2)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
    "Permite identificar se o ramal representado por este indice esta
    presente ou nao."
    ::= { impRamaisEntry 2 }

impRTNome OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
    "Permite definir um novo nome para um ramal. Corresponde a um numero
    que sera aplicado ao ramal por questao de organizacao do sistema do
    usuario. "
    ::= { impRamaisEntry 3 }

impRTTipo OBJECT-TYPE
    SYNTAX INTEGER {
        analogico(1),
        digital(2)
    }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
    "Permite configurar o sistema para refletir o tipo de ramal
    efetivamente instalado: analogico ou digital."
    ::= { impRamaisEntry 4 }

impRTSlot OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
    "Retorna o slot onde este ramal esta instalado."
    ::= { impRamaisEntry 5 }

impRTPosicao OBJECT-TYPE
    SYNTAX INTEGER (1..4)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
    "Retorna a posicao do ramal dentro do slot onde este ramal esta
    instalado."
    ::= { impRamaisEntry 6 }

impRTModoIdentDeChamada OBJECT-TYPE
    SYNTAX INTEGER {
        semIdentificacao(1),
        fskBell(2),
        dtmf(3),
        fsj23Mdmf(4),
        fsk23Sdmf(5)
    }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
    "Permite configurar e retornar o modo de identificador de chamadas
    para este ramal."

```

```

::= { impRamaisEntry 7 }

impRTInicioIdentDTMF OBJECT-TYPE
    SYNTAX  INTEGER {
                a(1),
                b(2),
                c(3),
                d(4),
                asterisco(5),
                susenido(6)
            }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Permite configurar e retornar o caractere de inicio da identificacao
        de chamadas
        para este ramal."
::= { impRamaisEntry 8 }

impRTFinalIdentDTMF OBJECT-TYPE
    SYNTAX  INTEGER {
                a(1),
                b(2),
                c(3),
                d(4),
                asterisco(5),
                susenido(6)
            }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Permite configurar e retornar o caractere de final da identificacao
        de chamadas
        para este ramal."
::= { impRamaisEntry 9 }

impRTIdentChamadasInternas OBJECT-TYPE
    SYNTAX  INTEGER {
                ativado(1),
                desativado(2)
            }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Permite ativar ou desativar a funcao de identificador de chamadas
        internas."
::= { impRamaisEntry 10 }

impRTIdentChamadasExternas OBJECT-TYPE
    SYNTAX  INTEGER {
                ativado(1),
                desativado(2)
            }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Permite ativar ou desativar a funcao de identificador de chamadas
        externas."
::= { impRamaisEntry 11 }

impRTCategoriaParaIdentDTMF OBJECT-TYPE
    SYNTAX  INTEGER {
                ativado(1),
                desativado(2)
            }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION

```

```

"Permite ativar ou desativar a vinculacao de categoria do ramal para a
funcao de identificador de chamadas."
::= { impRamaisEntry 12 }

impRTTempoDeFlash OBJECT-TYPE
SYNTAX INTEGER
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Retorna o tempo de flash das centrais impacta. Nao eh passivel de
configuracao."
::= { impRamaisEntry 13 }

impRTDuracaoDigitoDTMFIdent OBJECT-TYPE
SYNTAX INTEGER (10..500)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite configurar a duracao do digito DTMF no identificador de
chamadas. O valor padrao para esta variavel eh 75."
::= { impRamaisEntry 14 }

impRTPausaInterdigitalIdent OBJECT-TYPE
SYNTAX INTEGER (10..500)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite configurar a pausa entre digitos para o identificador de
chamadas. O valor padrao para esta variavel eh 75."
::= { impRamaisEntry 15 }

impRTPausaAntesDoRingIdent OBJECT-TYPE
SYNTAX INTEGER (100..1000)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite configurar a pausa antes do ring para o identificador de
chamadas. O valor padrao para esta variavel eh 200."
::= { impRamaisEntry 16 }

impRTDuracaoDigitoDTMF OBJECT-TYPE
SYNTAX INTEGER (50..500)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite configurar a duracao do digito DTMF. O valor padrao para esta
variavel eh 100."
::= { impRamaisEntry 17 }

impRTPausaInterdigital OBJECT-TYPE
SYNTAX INTEGER (50..500)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite configurar a duracao da pausa entre digitos na discagem. O
valor padrao para esta variavel eh 100."
::= { impRamaisEntry 18 }

-- Categoria
impRTChamadaInternaFazRecebe OBJECT-TYPE
SYNTAX INTEGER {
    ativado(1),
    desativado(2)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite ativar ou desativar a funcao de realizar e receber chamadas

```

```

    internas. Quando desativado, significa que o ramal fica inoperante."
::= { impRamaisEntry 19 }

impRTAcessoExternoInternacional OBJECT-TYPE
    SYNTAX INTEGER {
        ativado(1),
        desativado(2)
    }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Permite ativar ou desativar a permissao para realizacao de chamadas
        Internacionais."
::= { impRamaisEntry 20 }

impRTAcessoExternoNacional OBJECT-TYPE
    SYNTAX INTEGER {
        ativado(1),
        desativado(2)
    }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Permite ativar ou desativar a permissao para realizacao de chamadas
        Nacionais."
::= { impRamaisEntry 21 }

impRTAcessoExternoRegional OBJECT-TYPE
    SYNTAX INTEGER {
        ativado(1),
        desativado(2)
    }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Permite ativar ou desativar a permissao para realizacao de chamadas
        Regionais."
::= { impRamaisEntry 22 }

impRTAcessoExternoLocal OBJECT-TYPE
    SYNTAX INTEGER {
        ativado(1),
        desativado(2)
    }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Permite ativar ou desativar a permissao para realizacao de chamadas
        externas Locais."
::= { impRamaisEntry 23 }

impRTCelularExternoInternacional OBJECT-TYPE
    SYNTAX INTEGER {
        ativado(1),
        desativado(2)
    }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Permite ativar ou desativar a permissao para realizacao de chamadas
        para celulares Internacionais."
::= { impRamaisEntry 24 }

impRTCelularExternoNacional OBJECT-TYPE
    SYNTAX INTEGER {
        ativado(1),
        desativado(2)
    }

```



```

MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite ativar ou desativar a permissao para realizacao de chamadas
para celulares Nacionais."
::= { impRamaisEntry 25 }

impRTCelularExternoRegional OBJECT-TYPE
SYNTAX INTEGER {
    ativado(1),
    desativado(2)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite ativar ou desativar a permissao para realizacao de chamadas
para celulares Regionais."
::= { impRamaisEntry 26 }

impRTCelularExternoLocal OBJECT-TYPE
SYNTAX INTEGER {
    ativado(1),
    desativado(2)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite ativar ou desativar a permissao para realizacao de chamadas
para celulares Locais."
::= { impRamaisEntry 27 }

-- Este grupo consiste de uma tabela que representa as variáveis
-- associadas aos juntores de um PABX Impacta.
-- impJuntores OBJECT IDENTIFIER ::= { impacta 3 }

impJuntoresInstalados OBJECT-TYPE
SYNTAX INTEGER (2..68)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Descreve o numero de juntores instalados neste PABX. Este numero eh,
no minimo 2 e no maximo o valor da variavel impGerNumMaximoJuntores."
::= { impJuntores 1 }

impJuntoresTable OBJECT-TYPE
SYNTAX SEQUENCE OF ImpJuntoresEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Tabela que contem informacao de todos os juntores instalados neste
PABX."
::= { impJuntores 2 }

impJuntoresEntry OBJECT-TYPE
SYNTAX ImpJuntoresEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Entrada da tabela que um juntor deste PABX."
INDEX { impJTIndex }
::= { impJuntoresTable 1 }

ImpJuntoresEntry ::=
SEQUENCE {
    impJTIndex

```

```

        INTEGER,
    impJTPresent
        INTEGER,
    impJTDirecao
        INTEGER,
    impJTTipoDeAtendimento
        INTEGER,
    impJTTipoDeDiscagem
        INTEGER,
    impJTTipoDeIdentificacao
        INTEGER,
    impJTSinalizacaoDeProgressoDeChamada
        INTEGER,
    impJTDetectarTomDeChamada
        INTEGER,
    impJTDiscarSemTomDeLinha
        INTEGER,
    impJTAtenderDuranteRingAtivo
        INTEGER,
    impJTBloqueioDeChamadaSainte
        INTEGER,
    impJTBloqueioUrgenteDeChamadaSainte
        INTEGER,
    impJTBloqueioDeChamadaEntrante
        INTEGER,
    impJTBloqueioUrgenteDeChamadaEntrante
        INTEGER
    }

impJTIndex OBJECT-TYPE
    SYNTAX  INTEGER (1..68)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Indice que identifica uma entrada da tabela e, conseqüentemente, um
        juntor do PABX. Esta variavel podera assumir valores de 1 ateh
        impJuntoresInstalados."
    ::= { impJuntoresEntry 1 }

impJTPresent OBJECT-TYPE
    SYNTAX  INTEGER {
        true(1),
        false(2)
    }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Permite identificar se o juntor representado por este indice esta
        presente ou nao."
    ::= { impJuntoresEntry 2 }

impJTDirecao OBJECT-TYPE
    SYNTAX  INTEGER {
        saida(1),
        entrada(2),
        bidirecional(3)
    }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Permite identificar se o juntor representado por este indice esta
        presente ou nao."
    ::= { impJuntoresEntry 3 }

impJTTipoDeAtendimento OBJECT-TYPE
    SYNTAX  INTEGER {
        semSinalizacao(1),

```

```

                invertPolaridade(2)
            }
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite definir o tipo de atendimento feito pelo juntor."
 ::= { impJuntoresEntry 4 }

impJTTipoDeDiscagem OBJECT-TYPE
SYNTAX INTEGER {
    pulso(1),
    dtmf(2),
    mfcFwd(3),
    mfcBwd(4),
    fskBell(5)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite definir o tipo de discagem feita pelo juntor."
 ::= { impJuntoresEntry 5 }

impJTTipoDeIdentificacao OBJECT-TYPE
SYNTAX INTEGER {
    semIdentificacao(1),
    fskBell(2),
    dtmf(3),
    mfcA(4),
    mfcB(5),
    mfcC(6)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite definir o tipo de identificacao de chamada feita pelo
 juntor."
 ::= { impJuntoresEntry 6 }

impJTSinalizacaoDeProgressoDeChamada OBJECT-TYPE
SYNTAX INTEGER {
    timeout(1),
    s425Hz(2),
    dualTone(3)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite definir o tipo de sinalizacao de progresso de chamada para
 este juntor."
 ::= { impJuntoresEntry 7 }

impJTDetectarTomDeChamada OBJECT-TYPE
SYNTAX INTEGER {
    ativado(1),
    desativado(2)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite ativar ou desativar a deteccao de tom de chamada."
 ::= { impJuntoresEntry 8 }

impJTDiscarSemTomDeLinha OBJECT-TYPE
SYNTAX INTEGER {
    ativado(1),
    desativado(2)
}
MAX-ACCESS read-write

```

```

STATUS current
DESCRIPTION
"Permite ativar ou desativar a discagem sem tom de linha."
::= { impJuntoresEntry 9 }

impJTAtenderDuranteRingAtivo OBJECT-TYPE
SYNTAX INTEGER {
    ativado(1),
    desativado(2)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite ativar ou desativar a funcao de atendimento durante o ring
ativo para este juntor."
::= { impJuntoresEntry 10 }

impJTBloqueioDeChamadaSainte OBJECT-TYPE
SYNTAX INTEGER {
    ativado(1),
    desativado(2)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite bloquear ou liberar as chamadas saintes atraves deste
juntor."
::= { impJuntoresEntry 11 }

impJTBloqueioUrgenteDeChamadaSainte OBJECT-TYPE
SYNTAX INTEGER {
    ativado(1),
    desativado(2)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite fazer um bloqueio urgente para as chamadas saintes neste
juntor."
::= { impJuntoresEntry 12 }

impJTBloqueioDeChamadaEntrante OBJECT-TYPE
SYNTAX INTEGER {
    ativado(1),
    desativado(2)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite bloquear ou liberar as chamadas entrantes neste juntor."
::= { impJuntoresEntry 13 }

impJTBloqueioUrgenteDeChamadaEntrante OBJECT-TYPE
SYNTAX INTEGER {
    ativado(1),
    desativado(2)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Permite fazer um bloqueio urgente para as chamadas entrantes neste
juntor."
::= { impJuntoresEntry 14 }

```

END

4.2.5 A Implementação da MIB no Agente

No projeto da nova plataforma de comunicação VoIP das centrais Impacta foi adotada uma solução *Linux*. Em função disso, a adoção de soluções de *software* para a implementação de todo o sistema tinha como pré-requisito a compatibilidade com este sistema operacional.

Para o agente SNMP, por sugestão de consultores da Universidade Federal do Rio Grande do Sul, foi adotado o pacote *Net-SNMP*. O *Net-SNMP* é um conjunto de ferramentas destinado à implementação e utilização do SNMP em um sistema de gerenciamento. Fazem parte deste conjunto de ferramentas aplicativos para monitoramento, configuração e desenvolvimento de soluções SNMP (agentes).

Dentre estas ferramentas, pela conveniência deste trabalho, um destaque maior deve ser dado ao aplicativo *mib2c*. O *mib2c* é um aplicativo que gera um código em linguagem C a partir de um arquivo MIB fornecido como parâmetro. Isso simplifica a incorporação de uma nova MIB ao agente, deixando ao implementador apenas a tarefa de inclusão da interface de acesso ao sistema de *hardware* da plataforma alvo.

Neste item será feita uma descrição da implementação da MIB Impacta no agente *Net-SNMP*. Esta implementação foi feita em um microcomputador utilizando o sistema operacional *Linux*, distribuição *Ubuntu 8.04*.

4.2.5.1 A instalação do *Net-SNMP*

O pacote *Net-SNMP* está disponível no endereço <http://net-snmp.sourceforge.net>. A versão utilizada neste trabalho foi a 5.4.2.1. Como o objetivo deste trabalho é o desenvolvimento e não apenas a utilização das ferramentas, foi baixado todo o código fonte do *net-snmp* (*net-snmp-5.4.2.1.tar.gz*). Em seguida, este pacote foi descompactado em um diretório que passou a ser o diretório de trabalho para o desenvolvimento da MIB Impacta no agente.

4.2.5.2 A inclusão da MIB Impacta no agente [10]

Após ser instalado, o *Net-SNMP* a inclusão da MIB Impacta no agente pode ser descrita nos passos enumerados a seguir.

1. Primeiramente, copiar o arquivo da MIB Impacta (MIB-IMPACTA.txt) para os diretórios `<diretório de trabalho>/net-snmp-5.4.2.1/mibs`.

2. Definir e alterar as variáveis de ambiente necessárias à operação do agente:

```
$export MIBS=ALL          #disponibilizar todas as MIBs que o sistema
                          #implementa
$export MIBDIRS=/usr/local/share/snmp/mibs #tornar público o
                          #diretório das MIBs para o agente traduzir os
                          #identificadores de objetos em nomes
$export PATH=/usr/local/bin:$PATH      #tornar público o diretório
                          #onde se encontram os aplicativos como
                          #snmpget, snmpset, snmpwalk, snmptranslate
                          #etc.
```

3. Dentro do diretório *<diretório de trabalho>/net-snmp-5.4.2.1/mibs* O aplicativo *mib2c* é executado para a geração do código C base da implementação. A linha de comando para esta operação é

```
$mib2c MIB-IMPACTA:impacta
```

onde *MIB-IMPACTA* corresponde ao arquivo da MIB Impacta (*MIB-IMPACTA.txt*) e *impacta* corresponde ao ramo da árvore MIB para o qual o código será gerado.

Antes da efetiva geração dos arquivos destino, o usuário é questionado sobre o padrão de código a ser adotado

1. *ucd-snmp style code*
2. *Net-SNMP style code*

Por simplicidade, neste trabalho foi adotada a geração de código no estilo *ucd-snmp*.

O resultado desta operação é a criação de dois arquivos de código C fonte: *impacta.h* e *impacta.c*. Para que estes arquivos possam ser incorporados ao agente, uma alteração mínima deve ser implementada para evitar erro de compilação. Para tanto, no arquivo *impacta.c* devem ser criadas as variáveis globais inteiras *VAR* e *VALUE*. A constante *TABLE_SIZE* também deve ser substituída por um valor numérico conveniente para cada uma das duas tabelas da MIB Impacta.

4. Copiar os arquivos fonte gerados (*impacta.h* e *impacta.c*) para

```
<diretório de trabalho>/net-snmp-5.4.2.1/agent/mibgroup/
```

5. Dentro do diretório *<diretório de trabalho>/net-snmp-5.4.2.1/*, executar o aplicativo de configuração básica do *Net-SNMP* para que ele inclua a nova MIB Impacta.

```
$/configure --with-mib-modules="impacta"
```

Ao longo deste processo, o usuário é questionado sobre a versão padrão do SNMP

que o sistema deve considerar. Em virtude do padrão adotado para a implementação da MIB, a versão padrão considerada nesta implementação é a 2c. Outras informações são solicitadas ao usuário como endereço de correio eletrônico de uma pessoa para contato, nome do sistema e locais onde devem ser armazenados os arquivos de *log* e de informações persistentes.

```
Default version of SNMP to use (3): 2
System Contact Information (root@): pedropaulo\_pps@hotmail.com
System Location (Unknown): CEFET-IFSC
Location to write logfile (/var/log/snmpd.log):
Location to write persistent information (/var/net-snmp):
```

6. Para a compilação e instalação do agente SNMP basta executar a seqüência de comandos

```
$make clean      #para eliminar códigos objeto gerados anteriormente
$make           #para compilar todo o Net-SNMP
$make install   #para instalar o Net-SNMP
```

4.2.5.3 Execução e Teste do Agente

Uma vez instalado o agente SNMP, resta executá-lo e testar a sua resposta às variáveis da MIB *Impacta*. A execução do agente é feita através do comando

```
$/usr/local/sbin/snmpd -c /usr/local/share/snmp/snmpd.conf
```

onde a opção *-c* indica o arquivo de configuração a ser utilizado (neste caso, o arquivo */usr/local/shar/snmp/snmpd.conf*).

O teste do agente pode ser feito através de dois aplicativos: *snmpwalk* e *snmpget*. O *snmpwalk* permite ao usuário ter um retorno de todas as variáveis de uma MIB, ou conjunto de MIBs, implementada no agente, dependendo do nível de profundidade da árvore MIB do identificador de objetos fornecido como parâmetro. Um exemplo de utilização seria

```
$snmpwalk -c public localhost 1.3.6.1.4.1.26138.1.1
```

sendo:

-c public - a definição da comunidade de leitura para este agente *snmp*.

localhost - a indicação da máquina onde está o agente

1.3.6.1.4.1.26138.1.1 - indicação do ramo *impacta* dentro da árvore *MIB*

A resposta a este comando seria a apresentação de uma lista de identificadores de objetos com os seus respectivos valores, correspondentes a cada uma das variáveis da MIB Impacta.

Com o *snmpget* é possível ler uma única variável de uma MIB. Sendo assim, o identificador de objetos fornecido deve conter toda a informação da variável, incluindo a instância.

```
$snmpget -c public localhost 1.3.6.1.4.1.26138.1.1.1.1.0
```

sendo:

```
-c public - a definição da comunidade de leitura para este agente snmp.  
localhost - a indicação da máquina onde está o agente  
1.3.6.1.4.1.26138.1.1.1.1.0 - descrição da variável impGerSysDescr
```

4.2.5.4 Análise do Resultado

O resultado dos testes de implementação da MIB Impacta demonstrou que o agente foi ativado já incluindo a subárvore da MIB Impacta. Entretanto, existe ainda algum trabalho a ser feito sobre o sistema. Isso porque ambos os aplicativos, *snmpwalk* e *snmpget*, retornam apenas a primeira variável da MIB Impacta. Este problema está sendo depurado para que a implementação possa ser apresentada à empresa.

CONSIDERAÇÕES FINAIS

O trabalho aqui descrito está inserido em um projeto mais amplo de reimplementação da placa de rede das centrais Impacta. Esta nova placa de rede foi projetada sobre uma plataforma *ARM9* rodando *Linux* o que permitiu a incorporação de diversas aplicações para gerenciamento, monitoramento e segurança do sistema. Uma destas aplicações é o SNMP.

A modelagem preliminar da central Impacta para a criação do documento formal da MIB foi um trabalho baseado no ambiente de gerenciamento já utilizado para este tipo de equipamento. Existem ainda algumas características a serem incorporadas à MIB e, para tanto, está sendo feito um estudo de como implementá-las tendo como fator limitante a complexidade de tais variáveis e a limitação das definições formais da MIB. Este trabalho, sem dúvida, é o que tem demandado maior tempo de elaboração.

Uma vez definida a MIB, a depuração do documento formal aconteceu sem maiores problemas. O mecanismo de compilação de MIBs disponibilizado na Internet pela universidade de *Twente* (Holanda) mostrou-se útil para que fosse eliminada uma etapa preliminar na geração do código do agente SNMP. De outra feita, a depuração seria realizada pelo aplicativo de geração automática do código do agente com número maior de iterações.

Aproveitando a disponibilidade de aplicativos de código aberto oferecidos para a plataforma *Linux*, o agente foi implementado através do conjunto de ferramentas *Net-SNMP*. A distância entre a implementação manual de um agente SNMP, como era realizada há uma década, e a sua configuração através de um conjunto de ferramentas como este é medida em tempo de desenvolvimento que é significativamente reduzido. O aplicativo *mib2c* praticamente cria todo o código necessário à incorporação da MIB ao agente deixando ao programador apenas a tarefa de incorporar o mecanismo de conexão entre cada variável e o seu correspondente no sistema alvo.

Tal mecanismo de conexão está sendo definido como uma API que permite o acesso ao *hardware* da central, dando significado concreto a cada variável definida na MIB.

Existe ainda algum trabalho de depuração a ser feito no que diz respeito ao funcionamento do agente SNMP com relação à MIB Impacta. Isso porque o agente não

está percorrendo todas as variáveis da MIB quando solicitado. Este trabalho de depuração é de menor monta quando comparado ao trabalho de incorporação da API da central Impacta ao agente SNMP.

A implementação de um agente SNMP não é uma tarefa que acaba ao ser concluído este trabalho. Sempre com vistas a melhorar a característica dos produtos oferecidos ao mercado e ouvindo a opinião dos clientes, o sistema estará submetido a melhora constante começando pela redefinição da MIB através de um conjunto mais completo de variáveis.

REFERÊNCIAS

- [1] RFC 1157. **A Simple Network Management Protocol (SNMP)**. Maio 1990.
- [2] ITU-T X.700. **Management Framework for Open Systems Interconnection (OSI) for CCITT Applications – Data Communication Networks (Study Group VII)**. 1991.
- [3] **Aspectos Introdutórios Acerca do Protocolo CMIP**. 2009. <<http://www.gta.ufrj.br/grad/CMIP.html>>.
- [4] RFC 1213. **Management Information Base for Network Management of TCP/IP-based internets: MIB-II**. Março 1991
- [5] RFC 1155. **Structure and Identification of Management Information for TCP/IP-based Internets**. Maio 1990.
- [6] RFC 1066. **Management Information Base for Network Management of TCP/IP-based internets**. Agosto 1988.
- [7] Smith, B.; Hardin, J.; Phillips, G. **Linux Appliance Design A hands-on guide to building Linux Appliances**. No Starch Press. San Francisco. 2007.
- [8] Comer, D. **Internetworking With TCP/IP Volume 1: Principles, Protocols, and Architecture**. 5th edition. Prentice Hall. New Jersey. 2006.
- [9] Jones, M. **TCP/IP Application Layer Protocols for Embedded Systems**. Charles River Media. Massachusetts. 2002.
- [10] Perkins, D. **Understanding SNMP MIBs**. Revisão 1.1.7. Setembro 1993.