

MIB – MANAGEMENT INFORMATION BASE

1. INTRODUÇÃO.....	1
2. O QUE É UMA MIB?	1
3. MIB DA OSI.....	1
3.1 HIERARQUIA DE HERANÇA.....	2
3.2 HIERARQUIA DE NOMEAÇÃO	2
3.3 HIERARQUIA DE REGISTRO.....	2
4. MIB DA INTERNET	2
4.1 A ÁRVORE MIB II.....	2
4.2 O GRUPO SYSTEM.....	4
4.2.1 <i>Objetos de System para Gerenciamento de Configuração:</i>	4
4.2.2 <i>Objetos de System para Gerenciamento de Falhas:</i>	5
4.3 O GRUPO INTERFACES	5
4.3.1 <i>Objetos de Interfaces para Gerenciamento de Falhas:</i>	5
4.3.2 <i>Objetos de Interfaces para Gerência de Configuração</i>	6
4.3.3 <i>Objetos de Interfaces para Gerência de Performance</i>	6
4.3.4 <i>Objetos de Interfaces para Gerência de Contabilização</i>	7
4.4 O GRUPO ADDRESS TRANSLATION	8
4.5 O GRUPO IP	8
4.5.1 <i>Objetos de IP para Gerenciamento de Falhas</i>	8
4.5.2 <i>Objetos de IP para Gerenciamento de Configuração</i>	9
4.5.3 <i>Objetos de IP para Gerenciamento de Performance</i>	10
4.5.4 <i>Objetos de IP para Gerenciamento de Contabilização</i>	11
4.6 O GRUPO ICMP.....	11
4.7 O GRUPO TCP	12
4.7.1 <i>Objetos de TCP para Gerenciamento de Configuração</i>	12
4.7.2 <i>Objetos de TCP para Gerenciamento de Performance</i>	13
4.7.3 <i>Objetos de TCP para Gerenciamento de Contabilização</i>	14
4.7.4 <i>Objetos de TCP para Gerenciamento de Segurança</i>	14
4.8 O GRUPO UDP	15
4.8.1 <i>Objetos de UDP para Gerenciamento de Performance</i>	15
4.8.2 <i>Objetos de UDP para Gerenciamento de Contabilização</i>	15
4.8.3 <i>Objetos de UDP para Gerenciamento de Configuração</i>	16
4.8.4 <i>Objetos de UDP para Gerenciamento de Segurança</i>	16
4.9. O GRUPO EGP.....	16
4.9.1 <i>Objetos de EGP para Gerenciamento de Falhas</i>	16
4.9.2 <i>Objetos de EGP para Gerenciamento de Configuração</i>	17
4.9.3 <i>Objetos de EGP para Gerenciamento de Performance</i>	17
4.10 O GRUPO CMOT	18
4.11 O GRUPO TRANSMISSION	18
4.12 O GRUPO SNMP.....	18
4.12.1 <i>Objetos de SNMP para Gerenciamento de Falhas</i>	18
4.12.2 <i>Objetos de SNMP para Gerenciamento de Performance</i>	19
4.12.3 <i>Objetos de SNMP para Gerenciamento de Contabilização</i>	20
4.12.4 <i>Objetos de SNMP para Gerenciamento de Segurança</i>	20
4.12.5 <i>Objetos de SNMP para Gerenciamento de Configuração</i>	20
5. COMPARAÇÃO ENTRE A MIB DA OSI E A MIB DA INTERNET.....	20

MIB – Management Information Base

Joice Lee Otsuka (otsuka@inf.ufrgs.br)

Disponível em: <http://penta.ufrgs.br/gr952/trab1/2capa.html>, acessado em 30/09/2004.

1. Introdução

O conhecimento das MIB's (Base de Informações Gerenciáveis) e, principalmente, o conhecimento de como utilizar estas informações, são de fundamental importância na Gerência de Redes.

Este documento procura introduzir o conceito de MIB e apresentar os dois principais padrões de MIB, a MIB da OSI e a MIB da Internet, aprofundando mais neste último, no qual serão apresentados todos os objetos gerenciados e suas possíveis utilizações.

2. O que é uma MIB?

Antes de definir o que é uma MIB, será introduzido o conceito de objetos gerenciados.

Um 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.

Dessa forma, a MIB (Management Information Base) é o conjunto dos objetos gerenciados, que procura abranger todas as informações necessárias para a gerência da rede, possibilitando assim, a automatização de grande parte das tarefas de gerência.

Os padrões de gerenciamento OSI e Internet definiram MIBs que representam os objetos necessários para a gerência de seus recursos. Neste hipertexto serão apresentadas considerações sobre a MIB da OSI e a MIB Internet, bem como as diferenças entre as MIBs desses dois padrões.

3. MIB da OSI

O padrão OSI define três modelos para gerência de redes: o modelo organizacional, o modelo informacional e o modelo funcional. O modelo organizacional descreve a forma pela qual a gerência pode ser distribuída entre domínios e sistemas dentro de um domínio. O modelo funcional descreve as áreas funcionais e seus relacionamentos. Já o modelo informacional provê a base para a definição de objetos gerenciados e suas relações, classes, atributos, ações e nomes.

Na definição de objetos gerenciados é utilizada a orientação a objetos. Objetos com características semelhantes são agrupados em classes de objetos. Uma classe pode ser uma subclasse de outra, e a primeira herda todas as propriedades da segunda. Uma classe é definida pelos atributos da classe, pelas ações que podem ser invocadas, pelos eventos que podem ser relatados, pela subclasse a qual ela deriva e pela superclasse na qual ela está contida.

Para a definição dos objetos gerenciados devem-se considerar três hierarquias: hierarquia de herança, de nomeação e de registros usados na caracterização e identificação de objetos gerenciados.

A seguir descreveremos cada uma das hierarquias mencionadas acima.

3.1 Hierarquia de Herança

Também denominada hierarquia de classe, tem como objetivo facilitar a modelagem dos objetos, através da utilização do paradigma da orientação a objetos. Assim podem ser definidas classes, superclasses, subclasses. Trata-se de uma ferramenta para uma melhor definição de classes.

3.2 Hierarquia de Nomeação

A hierarquia de nomeação, também chamada hierarquia de *containment*, descreve a relação de "estar contido em" aplicado aos objetos. Um objeto gerenciado está contido dentro de um e somente um objeto gerenciado.

Um objeto gerenciado existe somente se o objeto que o contém existir, e dependendo da definição, um objeto só pode ser removido se aqueles que lhe pertencerem forem removidos primeiro.

3.3 Hierarquia de Registro

A hierarquia de registro é usada para identificar de maneira universal os objetos, independentemente das hierarquias de heranças e nomeação. Esta hierarquia é especificada segundo regras estabelecidas pela notação ASN.1 (*Abstract Syntax Notation. One*).

Assim, cada objeto é identificado por uma sequência de números, correspondente aos nós percorridos desde a raiz, até o objeto em questão.

4. MIB da Internet

O RFC 1066 apresentou a primeira versão da MIB para uso com o protocolo TCP/IP, a MIB-I. Este padrão explicou e definiu a base de informação necessária para monitorar e controlar redes baseadas no protocolo TCP/IP. O RFC 1066 foi aceito pela IAB (Internet Activities Board) como padrão no RFC 1156.

O RFC 1158 propôs uma segunda MIB, a MIB-II, para uso com o protocolo TCP/IP, sendo aceita e formalizada como padrão no RFC 1213. A MIB-II expandiu a base de informações definidas na MIB-I.

No padrão Internet os objetos gerenciados são definidos em uma árvore de registro, equivalente a hierarquia de registro do padrão OSI, e que será descrita com maiores detalhes a seguir.

4.1 A árvore MIB II

A MIB II usa uma arquitetura de árvore, definida na ISO ASN.1, para organizar todas as suas informações. Cada parte da informação da árvore é um nó rotulado que contém:

- um identificador de objetos (OID): sequência de números separados por pontos;
- uma pequena descrição textual: descrição o nó rotulado

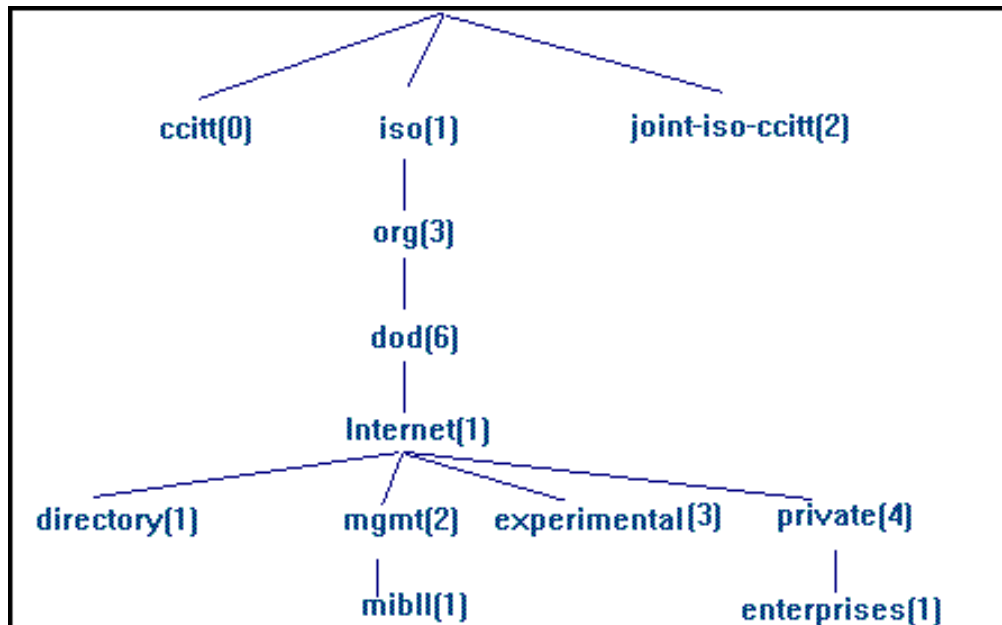
Exemplo:

directory(1)

identificador de objetos: 1.3.6.1.1

descrição textual: {internet 1}

Um nó rotulado pode ter sub-árvores contendo outros nós rotulados. Caso não tenha sub-árvores, ou nós folhas, ele conterá um valor e será um objeto.



O nó raiz da árvore MIB não tem nome ou número, mas tem três sub-árvores:

1. **ccitt(0)**, administrada pelo CCITT
2. **iso(1)**, administrada pela ISO
3. **joint-iso-ccitt(2)**, administrada pela ISO juntamente com o CCITT.

Sob o nó iso(1), estão outras sub-árvores, como é o caso da sub-árvore org(3), definida pela ISO para conter outras organizações. Uma das organizações que está sob a sub-árvore org(3) é o Departamento de Defesa dos EUA (DOD), no nó dod(6). A Internet(1) está sob o dod(6), e possui quatro sub-árvores:

- **directory(1)**: contém informações sobre o serviço de diretórios OSI (X.500)
- **mgmt(2)**: contém informações de gerenciamento, é sob esta sub-árvore que está o nó da mibII, com o identificador de objeto 1.3.6.1.2.1 ou { mgmt 1 }.
- **experimental(3)**: contém os objetos que ainda estão sendo pesquisados pela IAB.
- **private(4)**: contém objetos definidos por outras organizações.

Abaixo da sub-árvore mibII estão os objetos usados para obter informações específicas dos dispositivos da rede. Esses objetos são divididos em 11 grupos, que são apresentados na tabela abaixo:

<i>Grupos</i>	<i>Informações</i>
<u>system(1)</u>	Sistema de operação dos dispositivos da rede
interfaces(2)	interface da rede com o meio físico
<u>address translation(3)</u>	mapeamento de endereços IP em endereços físicos
<u>ip(4)</u>	protocolo IP

<u>icmp(5)</u>	protocolo ICMP
<u>tcp(6)</u>	protocolo TCP
<u>udp(7)</u>	protocolo UDP
<u>egp(8)</u>	protocolo EGP
<u>cmot(9)</u>	protocolo CMOT
transmission(10)	meios de transmissões
snmp(11)	protocolo SNMP

Cada objeto contido nos grupos apresentados na tabela acima é descrito no [RFC1213](#). A descrição dos objetos é dividida em cinco partes: o nome do objeto, a sintaxe abstrata do objeto, a descrição textual do significado do objeto, o tipo de acesso permitido ao objeto (*read-only*, *read-write*, *write-only* ou não acessível), e o estado do objeto (obrigatório, opcional, obsoleto). O exemplo abaixo apresenta a descrição do objeto sysDescr{Internet 1}.

OBJECT

sysDescr{Internet 1}

Syntax:

DisplayString(SIZE(0..255))

Definition: descrição textual da entidade: nome completo, versão,...

Access: read-only

Status: mandatory

4.2 O Grupo SYSTEM

Este grupo contém informações sobre o sistema no qual se encontra a entidade gerenciada. Muitos destes objetos são usados no gerenciamento de configuração e gerenciamento de falhas.

4.2.1 Objetos de System para Gerenciamento de Configuração:

<i>Objeto</i>	<i>Informação usada no gerenciamento de configuração</i>
SysDescr	descrição do sistema
SysLocation	localização física do sistema
SysContact	pessoa responsável pelo sistema
SysName	nome do sistema

- **sysDescr** informa a descrição do sistema. Este dado pode ser útil tanto para gerenciar a configuração do dispositivo como para diagnosticar falhas.
- Os objetos **sysLocation**, **sysContact** e **sysName** informam respectivamente, a localização física do sistema, a pessoa de contato em caso de problemas, e o nome do dispositivo da rede. Estas informações são úteis quando há necessidade de contactar com alguém para um acesso físico a um dispositivo remoto.

4.2.2 Objetos de *System* para Gerenciamento de Falhas:

<i>Objeto</i>	<i>Informação usada no gerenciamento de falhas</i>
SysObjectID	fabricante do sistema
sysServices	qual camada de protocolo o sistema serve
sysUpTime	quanto tempo o sistema está operacional

- identificador de objeto encontrado em **sysObjectID** informa o fabricante do sistema. Este dado é importante quando para resolver um problema de um dispositivo, necessita-se conhecer o seu fabricante.
- o **sysServices** informa quais os níveis do modelo de referência da ISO, o dispositivo serve. Retorna a soma dos números de cada camada, usando, para cada camada, a fórmula $2^{(L-1)}$, onde L é o número da camada. Esta informação é útil para rastrear problemas quando a funcionalidade do dispositivo é desconhecida.
- o **sysUptime** informa a quanto tempo um sistema está funcionando. Através deste objeto, a aplicação de gerenciamento de falhas pode determinar se o sistema foi reiniciado desde a última vez que o objeto foi consultado.

4.3 O Grupo *Interfaces*

O Grupo *Interfaces* oferece dados sobre cada interface de um dispositivo gerenciável da rede. Essas informações são úteis para o gerenciamento de falhas, de configuração, de performance, e de contabilização.

O objeto **ifTable** contém informações sobre todas as interfaces de uma entidade.

4.3.1 Objetos de *Interfaces* para Gerenciamento de Falhas:

<i>Objeto</i>	<i>Informação usada no gerenciamento de falhas</i>
If AdminStatus	indica se a interface esta administrativamente up/down/test
ifOper Status	indica o status operacional da interface (up/down/test)
If Last Change	indica quando a interface mudou seu estado operacional

A combinação dos objetos **ifAdminStatus** e **ifOperStatus** determina o status da interface. A tabela abaixo apresenta as possíveis combinações.

ifAdminStatus:	Up(1)	Down(2)	Testing(3)
ifOperStatus:			
Up(1)	Operacional	N/A	N/A
Down(2)	Falha	Down	N/A
Testing(3)	N/A	N/A	Em teste

N/A - não aplicável.

O objeto **ifLastChange** informa quando a interface entrou no seu estado operacional atual.

4.3.2 Objetos de *Interfaces* para Gerência de Configuração

<i>Objeto</i>	<i>Informação usada no gerenciamento de configuração</i>
If Descr	nome da interface
ifType	tipo de interface
ifMtu	tamanho máximo do datagrama suportado pela interface
ifSpeed	largura de banda da interface
If AdminStatus	indica se a interface esta administrativamente up/down/test

- o objeto **ifDescr** nomeia a interface
- o objeto **ifType** atribui um tipo a interface
- o objeto **ifSpeed** é um medidor da velocidade da interface em bits por segundo. Útil quando deseja-se se saber velocidade atual de uma interface que aloca banda passante de acordo com a demanda de tráfego.
- o objeto **ifAdminStatus** permite que, através do comando SNMP **Set-Request**, configure-se remotamente a interface para on/off.

4.3.3 Objetos de *Interfaces* para Gerência de Performance

<i>Objeto</i>	<i>Informação usada no gerenciamento de performance</i>
If InDiscards	taxa de descartes de entrada
ifOutDiscards	taxa de descartes de saída
ifInErrors	taxa de erros de entrada
ifOutErrors	taxa de erros de saída
ifInOctets	taxa de bytes recebidos
ifOut Octets	taxa de bytes enviados
ifInUcastPkts	taxa de pacotes unicast recebidos
ifOutUcastPkts	taxa de pacotes unicast enviados
ifInNUcastPkts	taxa de pacotes no-unicast recebidos
ifOutNucastPkts	taxa de pacotes no-unicast enviados
ifInUnknownProtos	taxa de pacotes de protocolos desconhecidos recebidos
ifOutQLen	total de pacotes na fila de saída

- com os objetos ifInUcastPkts, ifOutUcastPkts, ifInNUcastPkts, ifOutNucastPkts, ifInErrors, ifOutErrors , podemos calcular as porcentagens de erro de entrada/saída.

- porcentagem de erro de entrada = $\text{ifInErrors} / (\text{ifInUcastPkts} + \text{ifInNUcastPkts})$
- porcentagem de erro de saída = $\text{ifOutErrors} / (\text{ifOutUcastPkts} + \text{ifOutNUcastPkts})$
- da mesma forma, com os objetos **ifInUcastPkts**, **ifOutUcastPkts**, **ifInNUcastPkts**, **ifOutNUcastPkts**, **ifInDiscards**, **ifOutDiscards**, podemos calcular as porcentagens de descartes de entrada/saída.
- o objeto **ifInUnknownProtos** informa o número de descartes realizados devido ao recebimento de pacotes de protocolo desconhecido. Portanto não há a detecção de nenhum problema, caso o valor dos objetos **ifInUnknownProtos** e **ifInDiscards** estiverem crescendo proporcionalmente.
- com os objetos **ifInOctets** e **ifOut Octets** pode-se calcular a taxa de utilização de uma interface:

Primeiro calcula-se o total de bytes recebidos e enviados em um intervalo de tempo entre x e y

$$\text{total de bytes} = (\text{ifInOctets}_y - \text{ifInOctets}_x) + (\text{ifOutOctets}_y - \text{ifOutOctets}_x)$$

depois calcula-se o total de bytes por segundo

$$\text{total de bytes por segundo} = \text{total de bytes} / (y - x)$$

calcula-se o total de bits por segundo

$$\text{total de bits por segundo} = \text{total de bytes por segundo} * 8$$

finalmente tem-se a taxa de utilização

$$\text{taxa de utilização} = (\text{total de bits por segundo}) / \text{ifSpeed}$$
- o objeto **ifOutQLen** indica se o dispositivo está tendo problemas em enviar dados para fora. Seu valor aumenta de acordo com o aumento do número de pacotes esperando para deixar a interface.
- os objetos **ifOutOctets** e **ifOutDiscards** juntos podem sinalizar um congestionamento na rede. Isto ocorre no caso de houver um aumento no valor do **ifOutDiscards** devido ao descarte de muitos pacotes que tentam deixar a interface, e uma diminuição do número total de bytes de saída, indicado pelo objeto **ifOutOctets**

4.3.4 Objetos de Interfaces para Gerência de Contabilização

<i>Objeto</i>	<i>Informação utilizada no gerenciamento de contabilização</i>
ifInOctets	taxa de bytes recebidos
ifOut Octets	taxa de bytes enviados
ifInUcastPkts	taxa de pacotes unicast recebidos
ifOutUcastPkts	taxa de pacotes unicast enviados
ifInNUcastPkts	taxa de pacotes no-unicast recebidos
ifOutNUcastPkts	Taxa de pacotes no-unicast enviados

- com os objetos **ifInOctets** e **ifOut Octets**, uma aplicação de gerenciamento de contabilização pode determinar o número de bytes enviados e recebidos em uma

interface. se a unidade de contabilização utilizada for pacotes ao invés de bytes, são utilizados os objetos **ifInUcastPkts**, **ifOutUcastPkts**, **ifInNUcastPkts**, **ifOutNUcastPkts** para calcular o número de pacotes recebidos e enviados.

4.4 O Grupo Address Translation

O grupo *Address Translation* não constitui mais um grupo separado. Seus objetos foram incorporados aos grupos de protocolos. Portanto, o uso de informações de conversão de endereços no gerenciamento de redes serão analisados em cada um desses grupos.

4.5 O Grupo IP

O IP é um protocolo de rede que utiliza um modo de serviço sem conexão para entregar datagramas. O grupo IP provê informações sobre o protocolo IP na entidade. Estas informações são subdivididas em quatro grupos:

1. objetos que informam erros e tipos dos pacotes IP vistos
2. tabela de informação sobre os endereços IP das entidades
3. tabela de roteamento IP da entidade
4. mapeamento de endereços IP para outros protocolos (substituindo o grupo Address Translation)

Os objetos do grupo IP podem ser aplicados ao gerenciamento de falhas, de configuração, de performance, e de contabilização.

4.5.1 Objetos de IP para Gerenciamento de Falhas

<i>Objeto</i>	<i>Informação usada para gerenciamento de falhas</i>
ipRoutTable	tabela de roteamento IP
ipNetToMediaTable	tabela de conversao de enderecos IP

Os objetos apresentados na acima são utilizados no gerenciamento de falhas. O ipRoutTable é uma tabela que possui as seguintes colunas:

- ipRouteDest: endereço IP do destino
- ipRouteIfIndex: número da interface
- ipRouteMetric1: métrica de roteamento #1
- ipRouteMetric2: métrica de roteamento #2
- ipRouteMetric3: métrica de roteamento #3
- ipRouteMetric4: métrica de roteamento #4
- ipRouteMetric5: métrica de roteamento #5
- ipRouteNextHop: próxima escala (endereço IP do roteador, usado para roteamento indireto)
- ipRouteType: tipo(direto, indireto, válido, inválido)
- ipRouteProto: mecanismo usado para determinar a rota
- ipRouteAge: idade da rota em segundos

- **ipRouteMask**: máscara da subrede para roteamento
- **ipRouteInfo**: ponteiro MIB para um protocolo de roteamento específico

Todos os objetos contidos no objeto **ipRouteTable** pode ser usado no gerenciamento de falhas. Por exemplo, eles podem ser usados para rastrear problemas de roteamento e de dispositivos que sinalizam informações de roteamento incorreto. Esses objetos permitem que a aplicação de gerenciamento de falhas produza a tabela de roteamento IP para um dispositivo e descubra rotas através da rede. Além disso os objetos **ipRouteType** e **IpRouteProto** informam como a informação de roteamento foi aprendida.

Assim como o **ipRouteTable**, o objeto **IpNetToMediaTable** é uma tabela com as seguintes entradas:

- **ipNetToMediaIfIndex**: número da interface
- **ipNetToMediaPhysAddress**: endereço do meio do mapeamento
- **ipNetToMediaNetAddress**: endereço IP do mapeamento
- **ipNetToMediaType**: como o mapeamento foi determinado (outro, inválido, dinâmico, estático)

Os objetos contidos no objeto **IpNetToMediaTable** informam o mapeamento de endereços IP para endereços em outros protocolos.

4.5.2 Objetos de IP para Gerenciamento de Configuração

<i>Objeto</i>	<i>Informação usada para gerenciamento de configuração</i>
IpFowarding	se o dispositivo é setado para IP
IpAddrTable	endereços IP do dispositivo
IpRouteTable	tabela de roteamento IP

- o objeto **ipForwarding** indica se a entidade está configurada para enviar (forward) datagramas IP.
- o objeto **IpAddrTable** é composto dos seguintes objetos:
 - **ipAdEntAddr**: o endereço IP desta entrada
 - **ipAdEntIfIndex**: número da interface
 - **ipAdEntNetMask**: máscara de sub-rede para endereços IP
 - **ipAdEntBcastAddr**: o bit menos significativo do endereço IP de broadcast

Estes dados são definidos na MIB somente para leitura (*read-only*), portanto a aplicação de gerenciamento de configuração poderá apenas consultá-los e não alterá-los.

- já o objeto **ipRouteTable** possui vários objetos definidos com permissão de leitura e escrita (*read-write*). Dessa forma, uma aplicação de gerenciamento de configuração pode entrar com novas rotas através do objeto **ipRouteDest** e mudar o tipo de uma rota com o objeto **ipRouteType**. Além disso é possível configurar-se as métricas de roteamento através dos objetos **ipRouteMetric1**, **ipRouteMetric2**, **ipRouteMetric3**, **ipRouteMetric4** e **ipRouteMetric5**.

4.5.3 Objetos de IP para Gerenciamento de Performance

<i>Objeto</i>	<i>Informação usada para gerenciamento de performance</i>
IpInReceives	taxa de datagramas de recebidos
IpInHdrErrors	taxa de erros de cabeçalho de entrada
IpInAddrErrors	taxa de erros de endereço de entrada
IpForwDatagrams	taxa de datagramas repassados
IpInUnknownProtos	taxa de datagramas de entrada para um protocolo desconhecido
IpInDiscards	taxa de datagramas de entrada descartados
IpInDelivers	taxa de datagramas de entrada entregues com sucesso
IpOutRequests	taxa de datagramas de saída (nao inclui os datagramas repassados)
IpOutDiscards	taxa de datagramas de saída descartados
ipOutNoRoutes	taxa de descartes ocorridos por falta de informacao de roteamento
ipRoutingDiscards	taxa de entradas de roteamento descartadas
ipReasmReqds	taxa de datagramas recebidos necessitando de remontagem
ipReasmOKs	taxa de datagramas com sucesso na remontagem
ipReasmFails	taxa de datagramas com falhas na remontagem
ipFragOKs	taxa de datagramas com sucesso na fragmentação
ipFragFails	taxa de datagrams com insucesso na fragmentação
ipFragCreates	taxa de fragmentos gerados

Os objetos listados na tabela acima podem ser usados no gerenciamento de performance. A seguir serão apresentadas algumas das formas que estes objetos podem ser utilizados para auxiliar no gerenciamento de performance.

- os objetos **ipInReceives** e **ipOutRequest** juntamente com alguns objetos do grupo Interface permitem o cálculo da taxa de tráfego IP de entrada e saída de uma entidade:
 - percentagem de tráfego ip de entrada: $(ifInUcastPkts + ifInNUcastPkts) / ipInReceives$
 - percentagem de tráfego ip de saída: $(ifOutUcastPkts + ifOutNUcastPkts) / ipOutRequest$
- pode-se calcular a percentagem de erros de datagramas IP:
 - percentagem de erros de entrada: $(ipInDiscards + ipInHdrErrors + ipInAddrErrors) / ipInReceives$
 - percentagem de erros de saída: $(ipOutDiscards + ipOutHdrErrors + ipOutAddrErrors) / ipOutRequests$
- o aumento do valor do objeto **ipRoutingDiscards** pode significar que um dispositivo está descartando entradas válidas devido a falta de recursos.

- um freqüente aumento no valor do objeto **ipInUnknownProtos** pode causar problemas de performance, pois os recursos estarão sendo desperdiçados em checagens de erros e determinação do destino, sendo que por fim o datagrama será descartado por ser destinado a um protocolo da camada superior desconhecido.
- pode-se calcular a taxa de *forwarding* e de recepção de datagramas IP por um dispositivo em um intervalo tempo entre x e y em segundos.
 - taxa de forwarding = $(\text{ipForwDatagrams}_y - \text{ipForwDatagrams}_x) / (y - x)$
 - taxa de entrada = $(\text{ipInReceives}_y - \text{ipInReceives}_x) / (y - x)$

Tendo estas duas taxas pode-se determinar se o sistema está repassando os datagramas IP rápido o bastante para satisfazer os requerimentos da rede. Se os pacotes recebidos estão sendo repassados a outros sistemas, a taxa de entrada e taxa de forwarding devem ser iguais. No entanto para que o cálculo seja mais exato deve-se subtrair da taxa de entrada a taxa de erros e pacotes IP destinados ao próprio sistema.

4.5.4 Objetos de IP para Gerenciamento de Contabilização

<i>Objeto</i>	<i>Informação usada para gerenciamento de contabilização</i>
ipOutRequests	número de pacotes IP enviados
ipInReceives	número de pacotes IP recebidos
ipInDelivers	número de pacotes IP de entrada entregues com sucesso

A tabela acima contém os objetos do grupo IP usados para auxiliar no gerenciamento de contabilização;

- os objetos **ipOutRequests** e **ipInReceives** informam o número total de pacotes IP enviados e recebidos por uma entidade.
- o objeto **ipInDelivers** informa o número de pacotes IP entregues com sucesso a protocolos de camada superior e aplicações.

4.6 O Grupo ICMP

O ICMP é um protocolo que carrega mensagens de erro e controle para dispositivos IP. O grupo ICMP contém objetos que fornecem informações sobre o protocolo ICMP na entidade em questão. Todos os seus objetos são aplicados ao gerenciamento de performance. Os objetos do grupo ICMP são apresentados na tabela abaixo:

<i>Objeto</i>	<i>Informação usada para gerenciamento de performance</i>
icmpInMsgs	taxa de recebimento de mensagens
icmpInErrors	taxa de erros de entrada
icmpInDestUnreachs	taxa de mensagens de Destino não Alcançado recebidas
icmpInTimeExcds	taxa de mensagens de Tempo Excedido recebidas

icmpInParmProbs	taxa de mensagens de Problemas com Parametros recebidas
icmpInSrcQuenchs	taxa de mensagens de Fonte Apagada recebidas
icmpInRedirects	taxa de mensagens de Redirecionamento recebidas
icmpInEchos	taxa de mensagens de Ecos (requisicao) recebidas
icmpInEchoReps	taxa de mensagens de Respostas a Ecos recebidas
icmpInAddrMasks	taxa de mensagens de Requisicao de Mascaras de Enderecos recebidas
icmpInAddrMaskReps	taxa de mensagens de Resposta a Mascaras de Enderecos recebidas
icmpOutMsgs	taxa de saida de mensagens
icmpOutErrors	taxa de erros de saída
icmpOutDestUnreachs	taxa de mensagens de Destino não Alcançado enviadas
icmpOutTimeExcds	taxa de mensagens de Tempo Excedido enviadas
icmpOutParmProbs	taxa de mensagens de Problema com Parametros enviadas
icmpOutSrcQuenchs	taxa de mensagens de Origem Apagada enviadas
icmpOutRedirects	taxa de mensagens de Redirecionamento enviadas
icmpOutEchos	taxa de mensagens de Eco (requisicao) enviadas
icmpOutEchoReps	taxa de mensagens Respostas de Eco enviadas
icmpOutTimestamps	taxa de mensagens de requisicao de Timestamp requisicao enviadas
icmpOutTimestampReps	taxa de mensagens de respostas ao pedido de Timestamp enviadas
icmpOutAddrMasks	taxa de mensagens de Requisicao de Mascara de Enderecos enviadas
icmpOutAddrMaskReps	taxa de mensagens de Resposta de Mascara de Enderecos enviadas

4.7 O Grupo TCP

O TCP é um protocolo de transporte que provê conexões confiáveis entre aplicações. Muitas implementações do TCP incluem recursos adicionais para lidar com controle de fluxo, congestionamento da rede, e a retransmissão de segmentos perdidos.

O grupo TCP pode ajudar no gerenciamento de configuração , performance, de contabilização, e de segurança.

Este grupo é subdividido em dois grupos:

1. objetos gerais sobre o TCP no sistema
2. uma tabela de valores para cada conexão TCP corrente, a qual é alterada a cada começo e fim de uma conexão TCP.

4.7.1 Objetos de TCP para Gerenciamento de Configuração

<i>Objeto</i>	<i>Informação usada para gerenciamento de configuração</i>
tcpRtoAlgorithm	algoritmo utilizado para determinar o "time out" de retransmissão de octetos TCP não confirmados
tcpRtoMin	valor mínimo permitido para o "time-out" de retransmissão TCP, em milissegundos
tcpRtoMax	valor máximo permitido para o "time-out" de retransmissão TCP, em milissegundos
tcpMaxConn	limite de conexões que podem ser abertas pela entidade de transporte do dispositivo
tcpCurrEstab	número de conexões de transporte corretamente abertas

- o objeto **tcpRtoAlgorithm** permite a configuração do algoritmo de retransmissão de octetos TCP não confirmados. A configuração inadequada pode resultar em congestionamento na rede ou distribuição injusta de banda passante. Consultando-se frequentemente os objetos **tcpRtoMin**, **tcpRtoMax** e **tcpRtoAlgorithm** pode-se verificar se a configuração está adequada ao ambiente de seu sistema de rede.
- o objeto **tcpMaxConn** ajuda a configurar a rede para suportar o número de conexões TCP remotas necessárias. Este número pode ser calculado observando-se o objeto **tcpCurrEstab** que informa o número de conexões TCP estabelecidas no momento.

4.7.2 Objetos de TCP para Gerenciamento de Performance

<i>Objeto</i>	<i>Informação usada para gerenciamento de performance</i>
tcpAttemptFails	número de tentativas de conexão falhadas
tcpEstabResets	número de reinicializações de conexões estabelecidas
tcpRetransSegs	número de segmentos retransmitidos
tcpInErrs	número de pacotes recebidos com erro
tcpOutRsts	número de vezes que a entidade tentou reinicializar uma conexão
tcpInSegs	taxa de segmentos TCP recebidos
tcpOutSegs	taxa de segmentos TCP enviados

- pela observação do objeto **tcpAttemptFails** pode-se medir a confiabilidade da rede, onde um número menor de falhas indicam uma rede mais confiável.
- pela observação do objeto **tcpEstabResets** também pode-se medir a confiabilidade da rede, sendo que quanto maior o número de conexões estabelecidas reinicializadas, menos confiável é a rede.
- o objeto **tcpRetransSegs** informa o número de segmentos TCP que o sistema está retransmitindo, esta informação pode indicar se uma entidade está tendo que fazer várias retransmissões para garantir a confiabilidade.

- o objeto **tcpInErrs** indica o número de segmentos recebidos com erro. O aumento deste objeto pode ser causado pelo encapsulamento incorreto dos segmentos pelo sistema de origem, alguma rede repassando os segmentos com erro, ou outras razões.
- o objeto **tcpOutRsts** informa o número de vezes que a entidade tentou reinicializar uma conexão..
- os objetos **tcpInSegs** e **tcpOutSegs** permitem a checagem da taxa de segmentos TCP que entram e saem da entidade.

4.7.3 Objetos de TCP para Gerenciamento de Contabilização

<i>Objeto</i>	<i>Informação usada para gerenciamento de contabilização</i>
tcpActiveOpens	número de vezes que o sistema abriu uma conexão
tcpPassiveOpens	número de vezes que o sistema recebeu um pedido de abertura de conexão
tcpInSegs	número total de segmentos TCP recebidos
tcpOutSegs	número total de segmentos TCP emitidos
tcpConnTable	tabela das conexões TCP correntes

- os objetos **tcpActiveOpens** e **tcpPassiveOpens** informa o número total de vezes em que uma conexão foi feita de ou para um sistema, respectivamente.
- os objetos **tcpInSegs** e **tcpOutSegs** juntos contam os segmentos TCP que entram e saem da entidade, respectivamente.
- o objeto **tcpConnTable** é uma tabela com as atuais conexões TCP, e contém os seguintes campos:
 - tcpConnState: estado da conexão
 - tcpConnLocalAddress: endereço TCP local
 - tcpConnLocalPort: endereço IP local
 - tcpConnRemAddress: endereço TCP remoto
 - tcpConnRemPort: endereço IP remoto
- O campo **tcpConnRemAddress** determina o endereço do sistema remoto que está conectado à entidade. A consulta frequente a este campo permite o conhecimento de quais sistemas usam os recursos da rede e durante quanto tempo.
Muitas aplicações TCP usam portas bem definidas, tornando possível determinar-se quais aplicações estão fazendo ou recebendo conexões TCP.

4.7.4 Objetos de TCP para Gerenciamento de Segurança

As informações da tabela **tcpConnTable** também podem ser usados para gerenciamento de segurança, pois permite o conhecimento dos sistemas que acessam recursos via TCP. O tempo de *polling* influenciará grandemente na eficiência do gerenciamento, pois um intruso pode levar apenas alguns segundos para pegar as informações que deseja e fechar a conexão. Se nenhum *poll* for feito neste intervalo, o intruso não será detectado.

4.8 O Grupo UDP

O UDP é um protocolo de transporte que, ao contrário do TCP, não garante segurança e nem estabelece conexões, ao invés disso ele usa um fluxo de datagramas para transportar as informações. O grupo UDP possui um número limitado de objetos. O grupo TCP pode ajudar no gerenciamento de performance, de contabilização, e de configuração.

Este grupo é subdividido em dois grupos:

1. objetos gerais sobre o UDP nesta entidade
2. entradas sobre as aplicações UDP, que estão recebendo datagramas correntemente, na entidade em questão

4.8.1 Objetos de UDP para Gerenciamento de Performance

<i>Objeto</i>	<i>Informação usada para gerenciamento de performance</i>
udpInDatagrams	taxa de datagramas recebidos
udpOutDatagrams	taxa de datagramas enviados
udpNoPorts	taxa de datagramas que não foram enviados para uma porta válida
udpInErrors	taxa de datagramas UDP recebidos com erro

- a consulta periódica aos objetos **udpInDatagrams** e **udpOutDatagrams** pode determinar a taxa de entrada e saída de datagramas.
- o objeto **udpNoPorts** informa quando a entidade está recebendo datagramas de uma aplicação inválida. Uma taxa alta desses datagramas pode resultar em problemas de performance.

4.8.2 Objetos de UDP para Gerenciamento de Contabilização

<i>Objeto</i>	<i>Informação usada para gerenciamento de contabilização</i>
udpInDatagrams	número total de datagramas UDP recebidos
udpOutDatagrams	número total de datagramas UDP enviados
udpTable	portas UDP recebendo datagramas correntemente

- os objetos **udpInDatagrams** e **udpOutDatagrams** determinam quantos datagramas UDP foram recebidos e enviados pela entidade.
- o objeto **udpTable** é composto pelos seguintes campos:
 - udpLocalAddress: endereço IP local
 - udpLocalPort: porta UDP local

Como o UDP não é um protocolo baseado em conexão, as entradas da tabela acima são válidas somente para o período em que a aplicação escuta uma porta.

4.8.3 Objetos de UDP para Gerenciamento de Configuração

Checando o objeto **udpTable**, pode-se determinar se as aplicações da entidade estão setadas corretamente. Por exemplo, se é conhecido que uma entidade tem uma aplicação que oferece impressão remota em uma determinada porta, esta configuração pode ser facilmente verificada usando o objeto **udpTable**.

4.8.4 Objetos de UDP para Gerenciamento de Segurança

O objeto **udpTable** também pode ser usado para gerenciamento de segurança. Pode-se checar este objeto para assegurar que uma entidade não executou uma determinada aplicação. Por exemplo, se determinarmos que uma determinada aplicação escuta requisições por uma porta UDP específica. A ferramenta de gerenciamento pode checar o objeto **udpTable** de todos os sistemas para verificar se esta porta UDP local está sendo escutada.

4.9. O Grupo EGP

O EGP é um protocolo que informa a um dispositivo de rede IP como alcançar outras redes IP. Ele não informa a rota completa para a outra rede, mas ela permite que um dispositivo saiba em que direção que a rede existe. Redes IP podem ser agrupadas em áreas lógicas chamadas *sistemas autônomos*. Um *sistema autônomo* geralmente é uma rede e suas subredes associadas ou uma coleção de redes e subredes sob uma mesma administração. Dois dispositivos de rede em dois *sistemas autônomos* distintos podem compartilhar informações de alcançabilidade via EGP.

Os dispositivos de rede que comunicam com o EGP entre *sistemas autônomos* são chamados *vizinhos EGP*. Cada processo EGP tem uma relação um-para-um com cada vizinho. Cada vizinho EGP conversa um protocolo *hello* que periodicamente informa outros vizinhos que ele ainda está ativo. Quando o sistema consulta a informação de alcançabilidade do vizinho, ele está fazendo um *EGP poll*.

Os objetos do grupo EGP são subdivididos em dois grupos:

1. informações sobre o EGP nesta entidade.
2. uma tabela de entradas contendo informações sobre cada vizinho EGP.

Os objetos do grupo EGP podem ser aplicados ao gerenciamento de falhas, de configuração, de performance, e de contabilização.

4.9.1 Objetos de EGP para Gerenciamento de Falhas

<i>Objeto</i>	<i>Informação usada para gerenciamento de falhas</i>
egpNeighState	estado de cada vizinho EGP
egpNeighStateUps	número de vezes que um vizinho EGP entrou no estado UP
egpNeighStateDows	número de vezes que um vizinho EGP entrou no estado DOWN

Os objetos listados acima estão contidos na tabela de vizinhos EGP (objeto **egpNeighTable**) .

O estado de um vizinho EGP pode prover informações de como a informação de roteamento é injetada no sistema autônomo. A aplicação de gerenciamento de falhas pode usar o objeto **egpNeighState** para conhecer o estado atual de um vizinho EGP. Se o vizinho EGP está no estado **up**, então ele deverá estar enviando informações sobre alcançabilidade de redes ao processo EGP local.

Sabendo-se quando um vizinho entra no estado **up** pode sinalizar sobre novas informações de roteamento que devem entrar no sistema autônomo. Já saber quando o vizinho para a comunicação, e entra no estado **down** pode ser útil na resolução de problemas de roteamento.

4.9.2 Objetos de *EGP* para Gerenciamento de Configuração

<i>Objeto</i>	<i>Informação usada para gerenciamento de configuração</i>
egpNeighState	estado de cada vizinho EGP
egpNeighAddr	endereço IP do vizinho EGP
egpNeighAs	sistema autônomo do vizinho EGP
egpNeighIntervalHello	intervalo entre retransmissões de comandos Hello
egpNeighIntervalPoll	intervalo entre retransmissões de comandos Poll
egpNighMode	modo de polling desta entidade EGP
egpNeighEventTrigger	permite iniciar ou finalizar uma comunicação
egpAs	sistema autônomo local

- o objeto **egpAs** informa o número do sistema autônomo da entidade EGP local.
- os demais objetos estão contidos no objeto **egpNeighTable** e dizem respeito a configuração de um vizinho específico .
- o objeto **egpNeighEventTrigger** pode ser usado para iniciar e encerrar uma comunicação com um vizinho EGP (já existente). Este objeto permite o controle do processo EGP do sistema. Este é o único objeto do grupo EGP que pode ser setado pelo engenheiro de rede através de um comando SNMP Set-Request.

4.9.3 Objetos de *EGP* para Gerenciamento de Performance

<i>Objeto</i>	<i>Informação usada para gerenciamento de performance</i>
egpInMsgs	taxa de mensagens recebidas
egpInErrors	taxa de mensagens recebidas com erro
egpOutMsgs	taxa de mensagens enviadas
egpOutErrors	taxa de mensagens nao enviadas devido a ocorrencia de erros
egpNeighInMsgs	taxa de mensagens recebidas deste vizinho EGP
egpNeighInErrs	taxa de mensagens recebidas com erro deste vizinho EGP
egpNeighOutMsgs	taxa de mensagens enviadas a este vizinho EGP
egpNeighOutErrs	taxa de mensagens não enviadas a este vizinho EGP devido a erros
egpNeighInErrMsgs	taxa de mensagens de erro recebidas deste vizinho EGP
egpNeighOutErrMsgs	taxa de mensagens de erro enviadas a este vizinho EGP

- os objetos **egpInMessages** e **egpOutMessages** permitem calcular a taxa de mensagens EGP que entram e saem da entidade. Geralmente esta taxa será insignificante, mas em alguns momentos de instabilidade da rede entre os vizinhos EGP, essa taxa pode aumentar e influenciar na performance da entidade.
- o aumento do valor dos objetos **egpInErrors** e **egpOutErrors** geralmente coincide com o aumento número de mensagens recebidas e enviadas pela entidade. Se uma mensagem é recebida com erro e uma resposta válida não é enviada, o vizinho EGP originador deverá retransmitir a mensagem. Quando a entidade não pode enviar mensagens EGP válidas devido a limitações de recursos, o valor do objeto **egpOutErrors** irá aumentar. Conseqüentemente, quando a taxa de **egpInMessages** aproxima-se da taxa de **egpOutErrors**, a entidade provavelmente estará tendo dificuldades em construir e enviar mensagens EGP.
- da mesma forma usando os objetos **egpNeighInMsgs**, **egpNeighInErrs**, **egpNeighOutMsgs** e **egpNeighOutErrs** permite o cálculo da taxa de entrada e saída de mensagens e erros de cada vizinho.

4.10 O Grupo CMOT

O Grupo CMOT existe somente por razões históricas. O CMOT é um protocolo que ajuda na transição do SNMP para o CMIS/CMIP. No entanto, embora a definição para o CMOT exista, nenhum trabalho significativo tem sido feito em cima deste protocolo desde algum tempo, e logo não há nenhum objeto neste grupo.

4.11 O Grupo Transmission

O Grupo *transmission* provê informações sobre o meio específico que forma a base das interfaces no sistema. Quando os padrões Internet para gerenciar vários tipos de meios forem definidos, este grupo será o prefixo para estas informações.

4.12 O Grupo SNMP

Os objetos do grupo SNMP podem ser aplicados em todas as cinco áreas de gerenciamento. Aplicações de gerenciamento de falhas observando só problemas SNMP podem achar útil conhecer o número de erros SNMP e sua frequência, enquanto aplicações de gerenciamento performance podem calcular a taxa de pacotes SNMP entrando e deixando a entidade. Já aplicações de gerenciamento de contabilização podem usar os objetos SNMP para encontrar o número de pacotes SNMP enviados ou recebidos pela entidade. E por fim, alguns objetos do grupo SNMP podem ajudar no gerenciamento de configuração e segurança.

4.12.1 Objetos de SNMP para Gerenciamento de Falhas

<i>Objeto</i>	<i>Informação usada para gerenciamento de falhas</i>
snmpInASNParseErrs	total de mensagens recebidas com erros ASN
snmpInTooBigs	total de mensagens recebidas com erro "too big"
snmpInNoSuchNames	total de mensagens recebidas com erro "noSuchName"
snmpInBadValues	total de mensagens recebidas com erro "badValue"

snmpInReadOnlys	total de mensagens recebidas com erro "readOnly"
snmpInGenErrs	total de mensagens recebidas com erro "genErr"
snmpOutTooBigs	total de mensagens enviadas com erro "too big"
snmpOutNoSuchNames	total de mensagens enviadas com erro "noSuchName"
snmpOutBadValues	total de mensagens enviadas com erro "badValue"
snmpOutGenErrs	total de mensagens enviadas com erro "genErr"

Os objetos listados na tabela acima informam erros referentes a mensagens SNMP. Esses erros não indicam erros na rede em si, mas pode informar que a entidade não está manipulando os pacotes SNMP apropriadamente. O número e tipos de erros também podem indicar que a entidade está recebendo pacotes SNMP com erros dos dispositivos da rede. A solução para esses erros geralmente está na configuração do gerente ou agente SNMP. Se a re-configuração não diminuir o número de erros, o problema provavelmente residirá na implementação do gerente ou agente SNMP.

4.12.2 Objetos de *SNMP* para Gerenciamento de Performance

<i>Objeto</i>	<i>Informação usada para gerenciamento de performance</i>
snmpInPkts	taxa de pacotes SNMP recebidos
snmpOutPkts	taxa de pacotes SNMP enviados
snmpInTotalReqVars	taxa de Get/Get-Next-Requests recebidas
snmpInTotalSetVars	taxa de Set-Requests recebidas
snmpInGetRequests	taxa de Get-Requests recebidas
snmpInGetNexts	taxa de Get-Next-Requests recebidas
snmpInSetRequests	taxa de Set-Requests recebidas
snmpInGetResponses	taxa de Get-Responses recebidas
snmpInTraps	taxa de Traps recebidas
snmpOutGetRequests	taxa de Get-Requests enviadas
snmpOutGetNexts	taxa de Get-Next-Requests enviadas
snmpOutSetRequests	taxa de Set-Requests enviadas
snmpOutGetResponses	taxa de Get-Responses enviadas
snmpOutTraps	taxa de Traps enviadas

Como qualquer outra atividade da entidade, o SNMP pode afetar a performance do sistema. Caso deseja-se conhecer a porcentagem de recursos uma entidade está usando para manipular o SNMP, pode-se calcular a taxa de pacotes SNMP recebidos ou enviados, usando os objetos **snmpInPkts** e **snmpOutPkts**.

Os demais objetos listados na tabela acima permitem que se conheçam os tipos de pacotes SNMP que a entidade está manipulando.

4.12.3 Objetos de SNMP para Gerenciamento de Contabilização

<i>Objeto</i>	<i>Informação usada para gerenciamento de contabilização</i>
snmpInPkts	taxa de pacotes SNMP recebidos
snmpOutPkts	taxa de pacotes SNMP enviados
snmpInTraps	taxa de traps recebidas
snmpOutTraps	taxa de traps enviadas

Os objetos **snmpInPkts**, **snmpOutPkts**, **snmpInTraps**, e **snmpOutTraps** permitem calcular o total de pacotes e traps recebidos e enviados. Estas informações podem ser úteis no gerenciamento de contabilização.

4.12.4 Objetos de SNMP para Gerenciamento de Segurança

<i>Objeto</i>	<i>Informação usada para gerenciamento de segurança</i>
snmpInBadCommunityNames	total de pacotes com uma community string incorreta
snmpInBadCommunityUses	total de pacotes com community string que não permite a operação requisitada

- o objeto **snmpInBadCommunityNames** conta o número de vezes que um usuário ou aplicação, na tentativa de comunicar-se com o SNMP de uma entidade, não informou a *community string* correta.
- o objeto **snmpInBadCommunityUses** conta o número de vezes que um pacote SNMP é recebido contendo uma *community string* que não permite a operação requisitada.

4.12.5 Objetos de SNMP para Gerenciamento de Configuração

<i>Objeto</i>	<i>Informação usada para gerenciamento de configuração</i>
snmpEnableAuthenTraps	indica se o agente SNMP pode enviar traps

5. Comparação entre a MIB da OSI e a MIB da Internet

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

A diferença entre estas duas MIB's reside nas hierarquias usadas para representar os objetos. Na MIB da ISO são definidas três hierarquias: hierarquia de herança, hierarquia de nomeação e hierarquia de registro.

A hierarquia de herança ou de classes está relacionada às propriedades associadas a um determinado objeto. Dentro desta hierarquia diz-se que objetos da mesma classe possuem propriedades similares.

No caso da Internet não são usados os conceitos de classes de objetos e seus respectivos atributos. São definidos tipos de objetos. A definição de tipo de objetos contém cinco campos: nome textual com o respectivo identificador de objeto (OBJECT IDENTIFIER), uma sintaxe ASN.1, uma descrição do objeto, o tipo de acesso e o status.

A hierarquia de nomeação ou de *containment* é usada para identificar instâncias de objetos. Este tipo de hierarquia não é definido no caso da Internet.

Finalmente tem-se a hierarquia de registro que é especificada em ambos padrões.