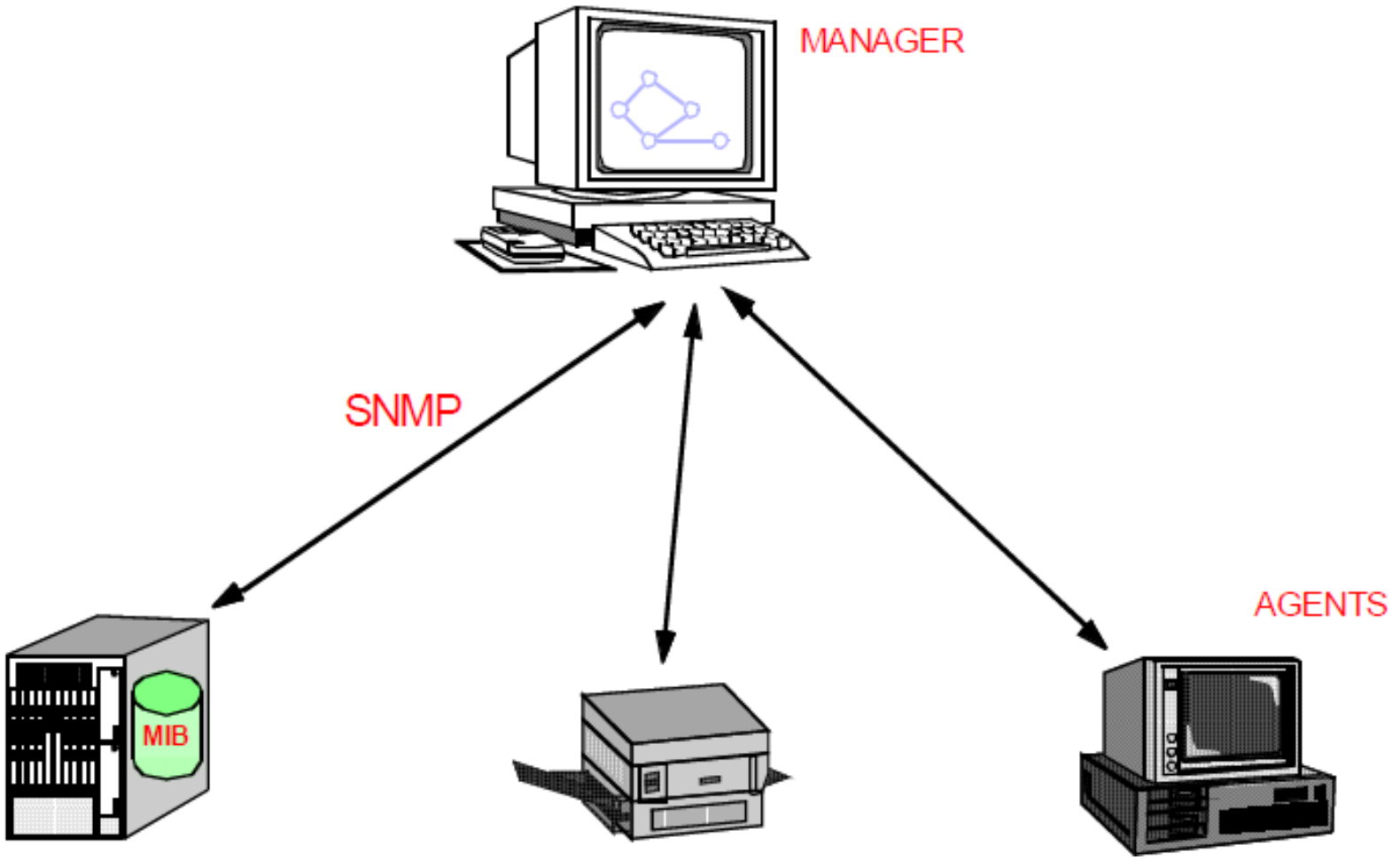
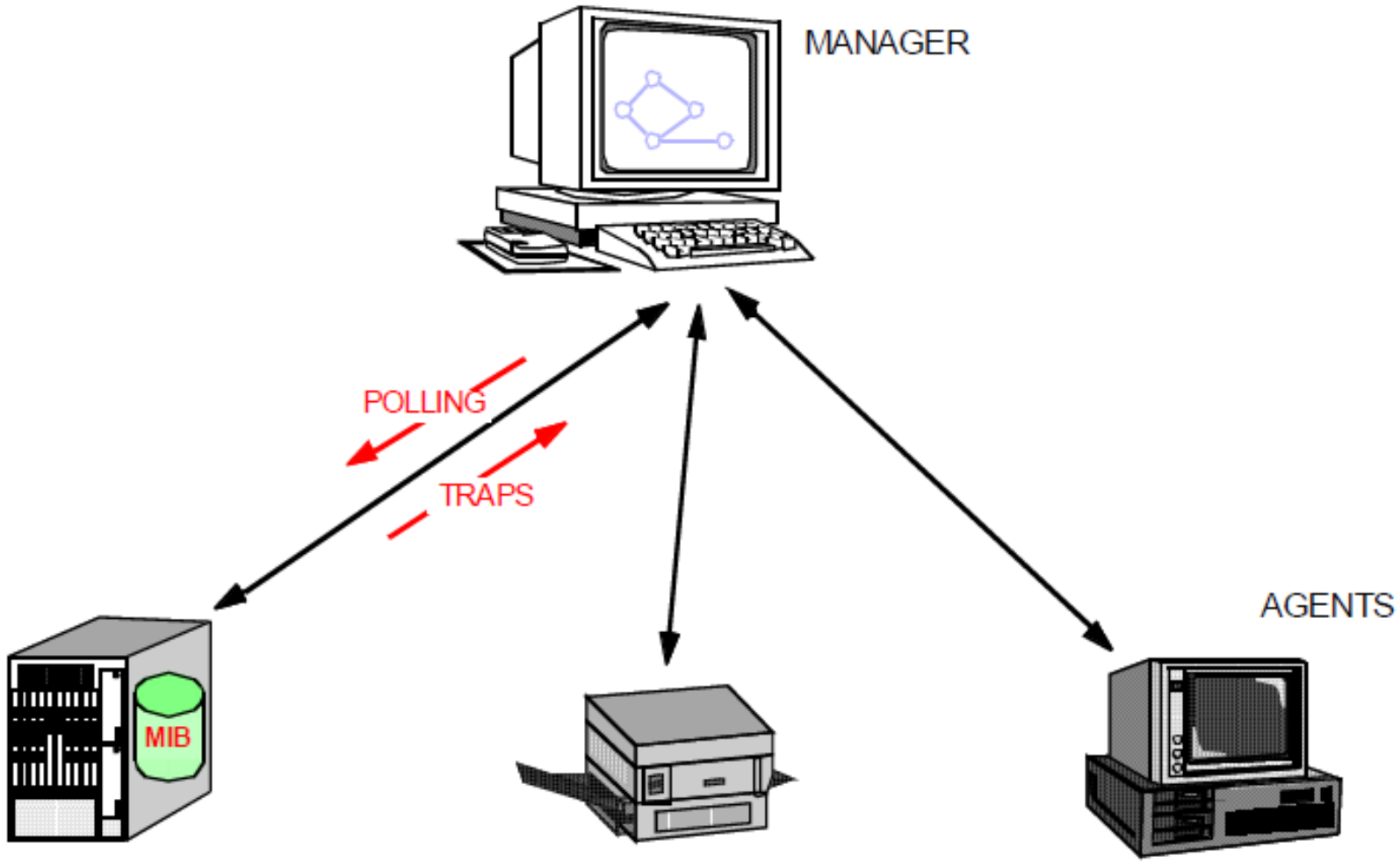
Histórico SNMP (Simple Network Management Protocol)

- padrão oficial (RFCs 3311-3318) Dez. 2002
- SNMPv3 (RFCs 2271-2275) Abr. 99
- RMON 2 Jan. 97
- Revisão SNMPv2 -SNMPv2c (RFCs 1901-1910) Jan. 96
- SNMPv2 (RFCs 1441-1450) Abr. 93
- RMON Nov. 91
- MIB-II Mar. 91
- SNMP versão 1 Mai. 90
- SNMP e CMOT desvinculados Ago. 89
- Working Groups SNMP e CMOT (CMIP over TCP) Fev. 88

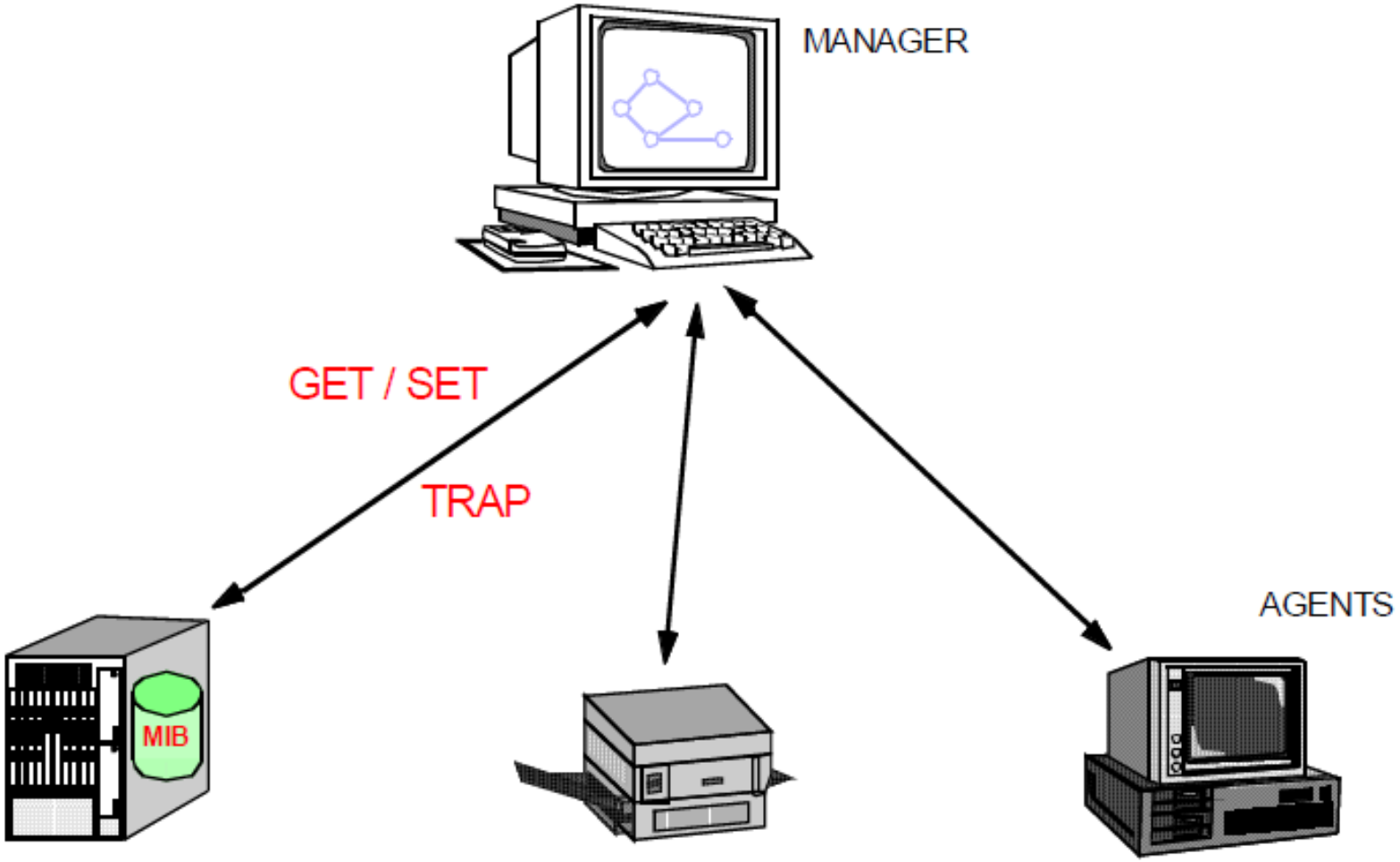
Arquitetura SNMP e seus componentes



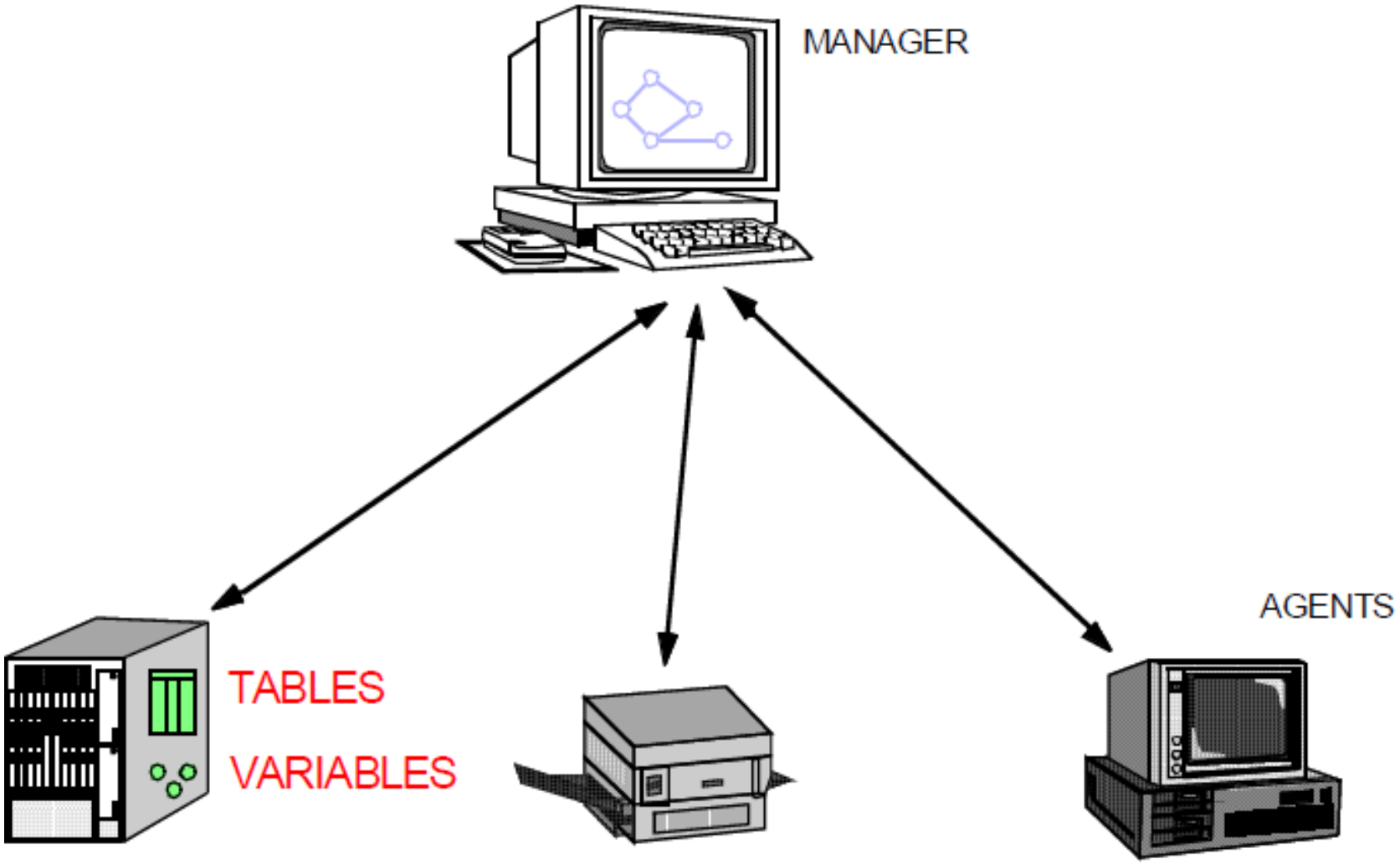
Arquitetura SNMP e seus componentes



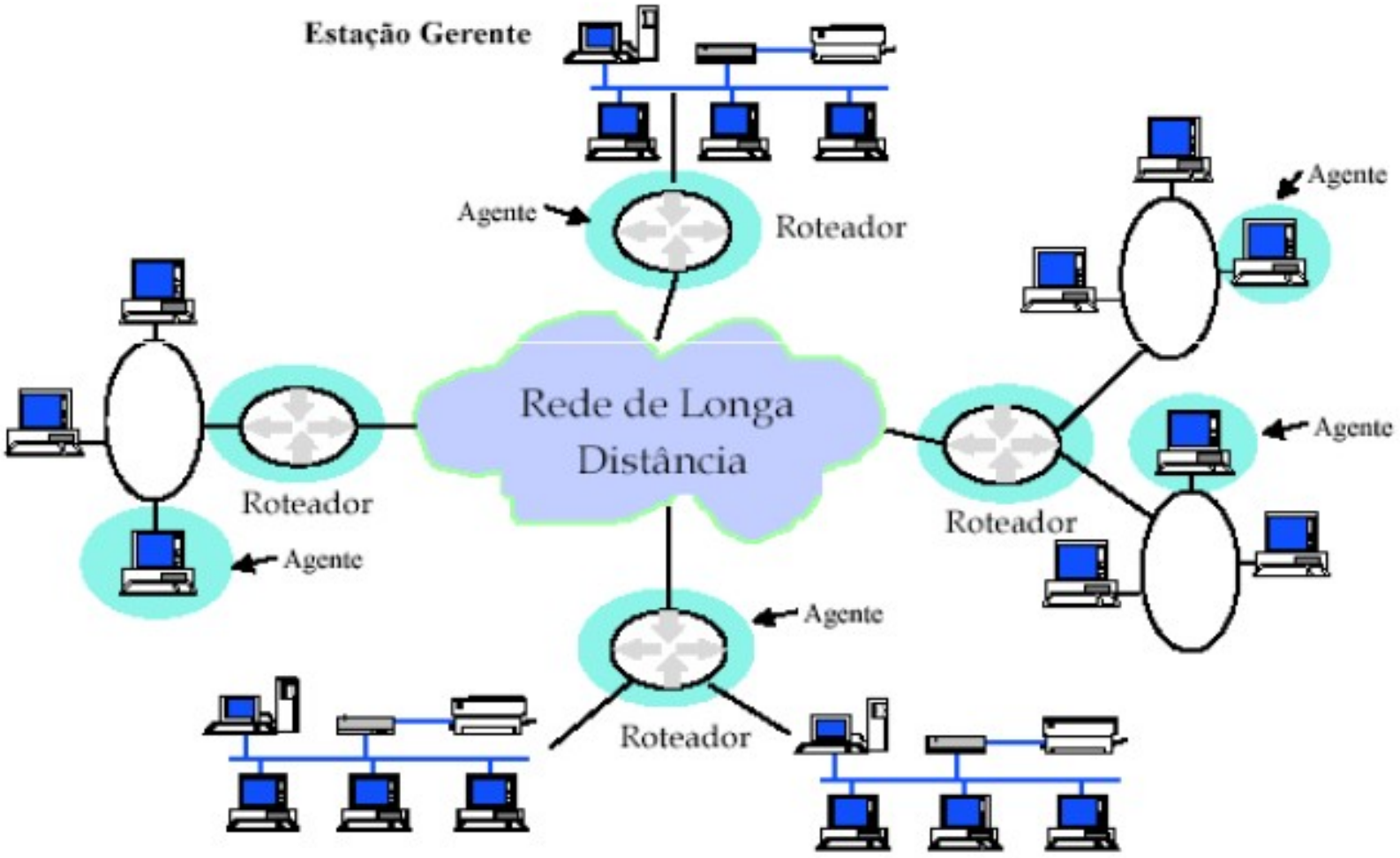
Arquitetura SNMP e seus componentes



Arquitetura SNMP e seus componentes

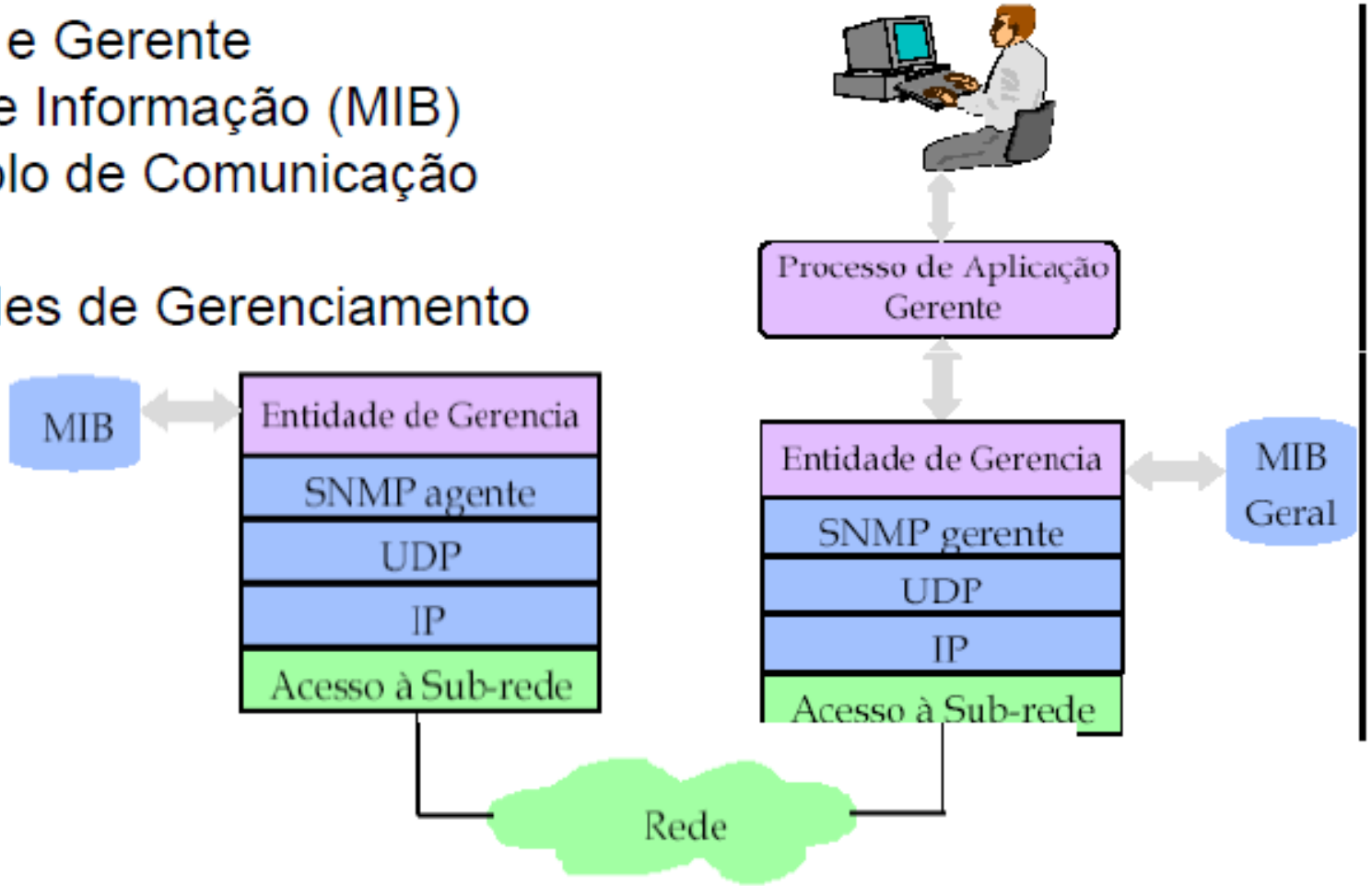


Componentes de Gerenciamento SNMP



Componentes de Gerenciamento

- Agente e Gerente
- Base de Informação (MIB)
- Protocolo de Comunicação (SNMP)
- Entidades de Gerenciamento



Visão Geral do SNMP

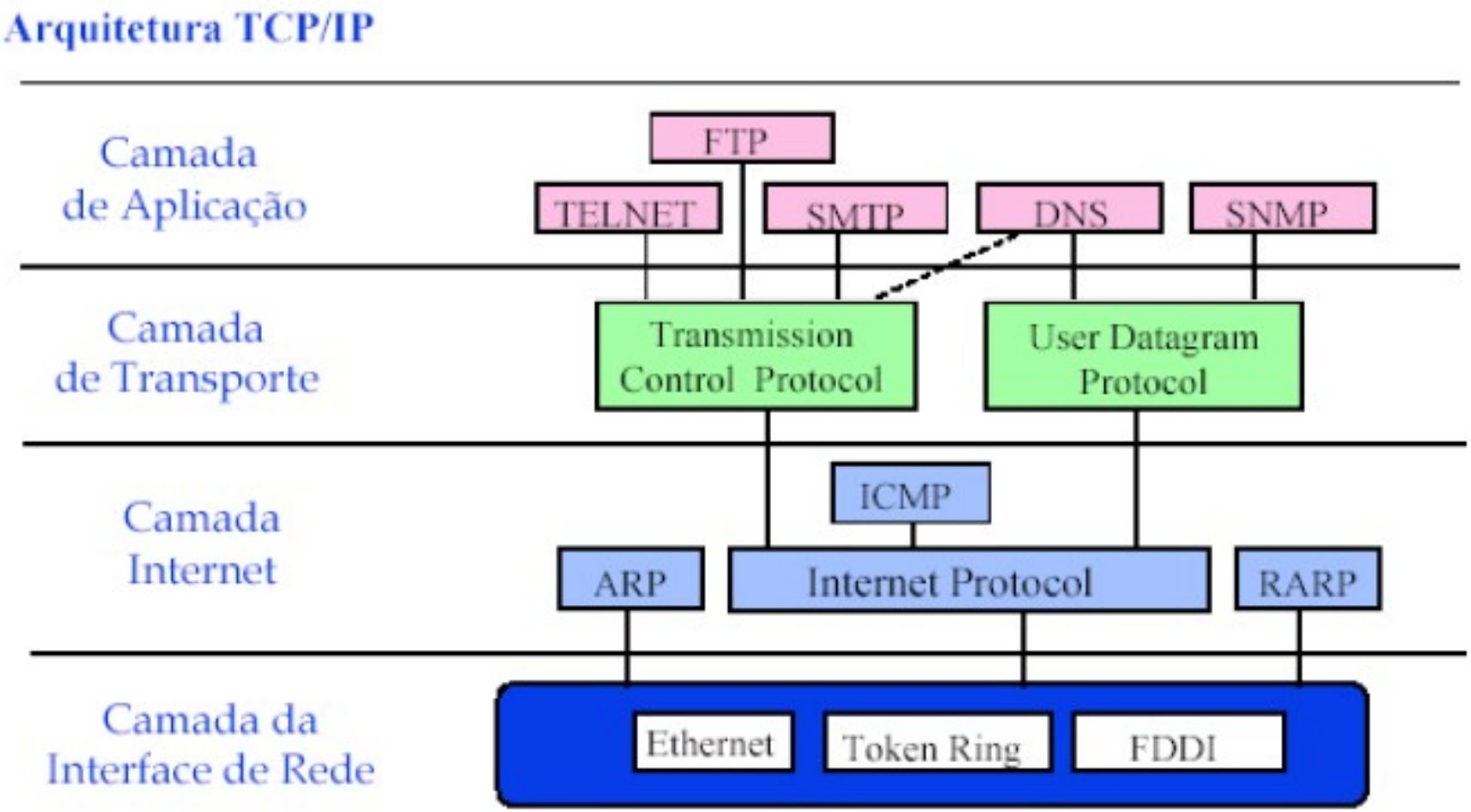
- **Management information base (MIB):**
  – base de dados distribuída com dados de gerenciamento de rede

- **Structure of Management Information (SMI):**
  – linguagem de definição para objetos da MIB

- **protocolo SNMP**
  – transporta informações e comandos sobre objetos entre o gerenciador e o elemento gerenciado

- **segurança, capacidades administrativas**
  – característica nova do SNMPv3

Gerenciamento SNMP



SMI – Linguagem e Estrutura de Definição de Dados

Propósito: A estrutura de informações de gerenciamento SMI define na realidade a estrutura da MIB. Desta forma, SMI estabelece os tipos de dados que poderão ser utilizados na MIB e, além disso, define como os recursos dentro da MIB são representados e nomeados. Todo isto para manter a simplicidade e extensibilidade dentro da MIB.

Tipos de Dados Básicos

INTEGER
Integer32
Unsigned32
OCTET STRING
OBJECT IDENTIFIER
IPaddress
Counter32
Counter64
Gauge32
Time Ticks
Opaque

SMI – Tipos de Dados Básicos

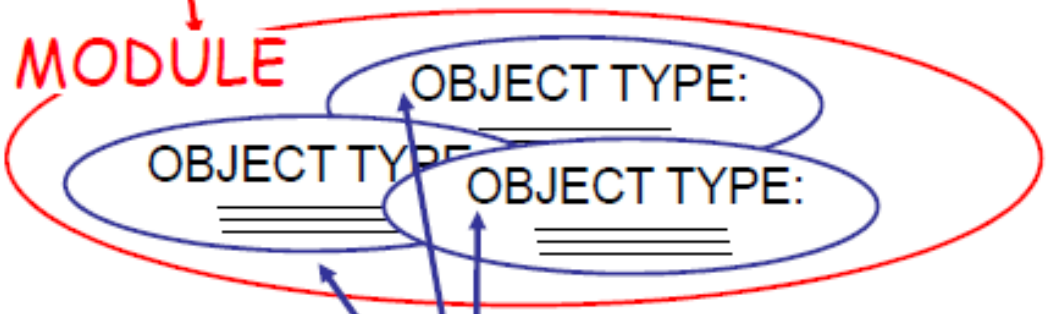
INTEGER	Inteiros [-2^{31} a $2^{31} - 1$]
Integer32	Inteiros [0 a $2^{31} - 1$]
OCTET STRING	String de octeto para dados binários
OBJECT IDENTIFIER	Elemento padronizado
IPaddress	Um endereço IP
Counter32	Um interiro não negativo até 2^{32}
Counter64	Um interiro não negativo até 2^{64}
Time Ticks	Representa tempo, em centésimo de segundo
Opaque	Um campo de bit arbitrário

SNMP - MIB

Um módulo MIB é especificado pela SMI como:

MODULE-IDENTITY

(100 MIBs padronizadas, mais proprietárias)



objetos especificados via construção OBJECT-TYPE da SMI

Exemplo de Objeto e Módulo

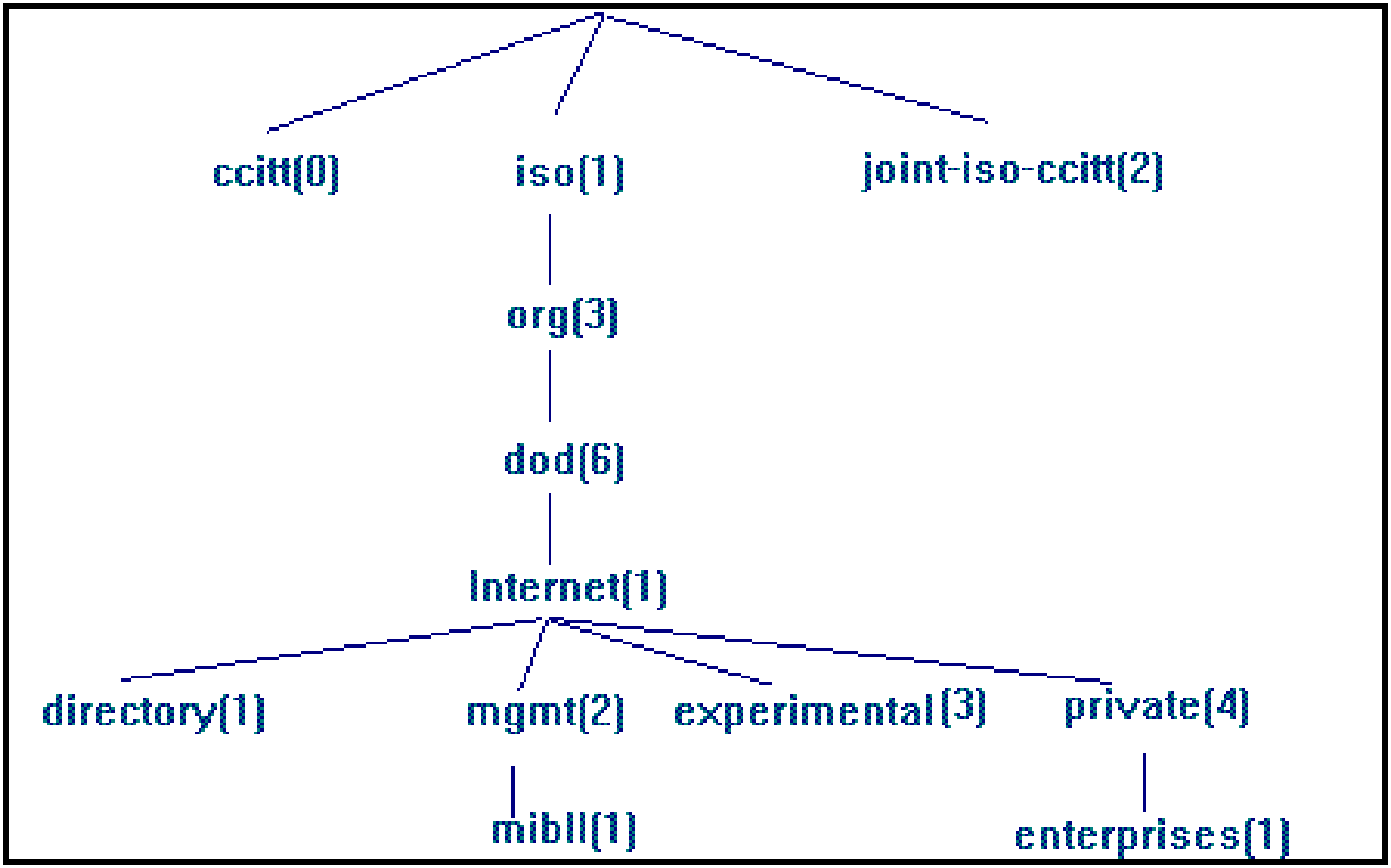
OBJECT-TYPE: ipInDelivers

```
ipInDelivers OBJECT TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS   current
  DESCRIPTION
     "The total number of input
      datagrams successfully
      delivered to IP user-
      protocols (including ICMP)"
::= { ip   9}
```

MODULE-IDENTITY: ipMIB

```
ipMIB MODULE-IDENTITY
  LAST-UPDATED "941101000Z"
  ORGANIZATION "IETF SNMPv2
          Working Group"
  CONTACT-INFO
    " Keith McCloghrie
          "
    ......
  DESCRIPTION
    "The MIB module for managing IP
    and ICMP implementations, but
    excluding the management of
    IP routes."
  REVISION "019331000Z"
 ..........
::= {mib-2 48}
```

Árvore de Registro de Tipos de Objetos

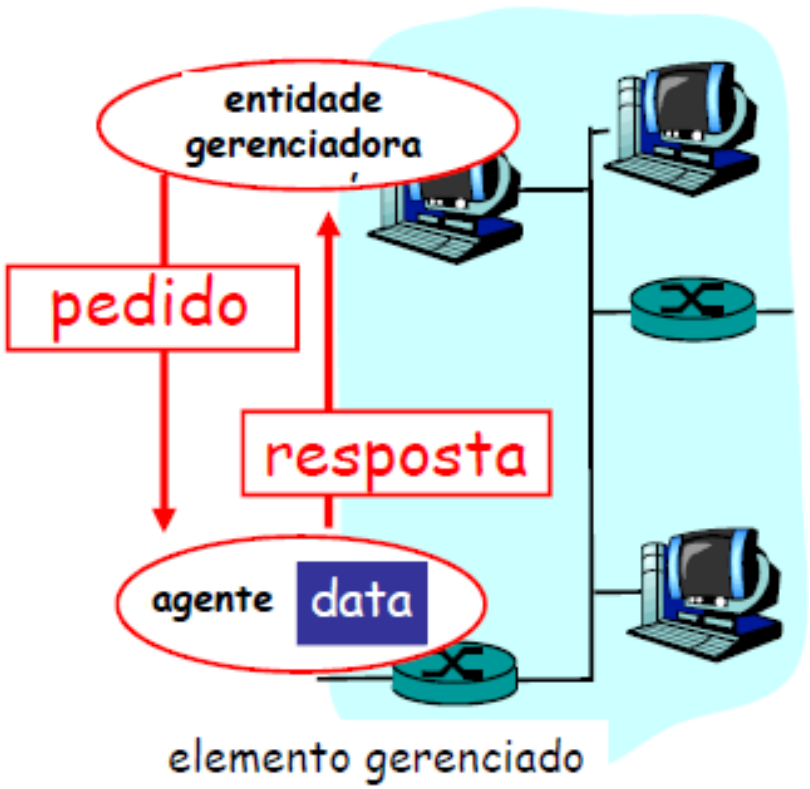


Exemplo MIB - Módulo UDP

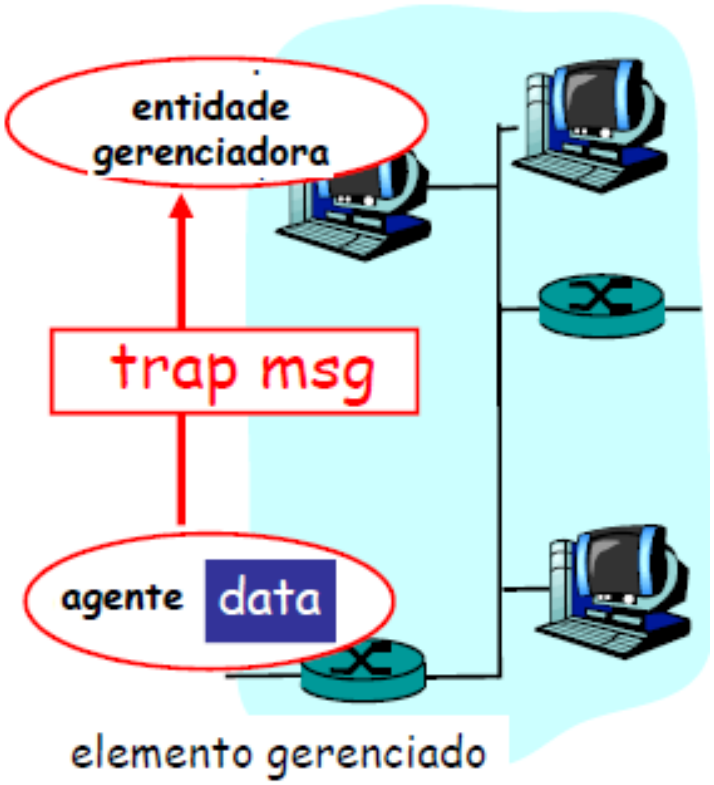
Object ID	Nome	Tipo	Comentários
1.3.6.1.2.1.7.1	UDPInDatagrams	Counter32	número total de datagramas entregues neste nó
1.3.6.1.2.1.7.2	UDPNoPorts	Counter32	número de datagramas com app destino inexistente
1.3.6.1.2.1.7.3	UDInErrors	Counter32	número de datagramas não entregues por outras razões
1.3.6.1.2.1.7.4	UDPOutDatagrams	Counter32	número de datagramas enviados
1.3.6.1.2.1.7.5	udpTable	SEQUENCE	uma linha para cada porta emn uso por uma aplicação, fornece o número da porta e o endereço IP

Protocolo SNMP

Duas formas de transportar informações da MIB: comandos e eventos



Modo comando/resposta



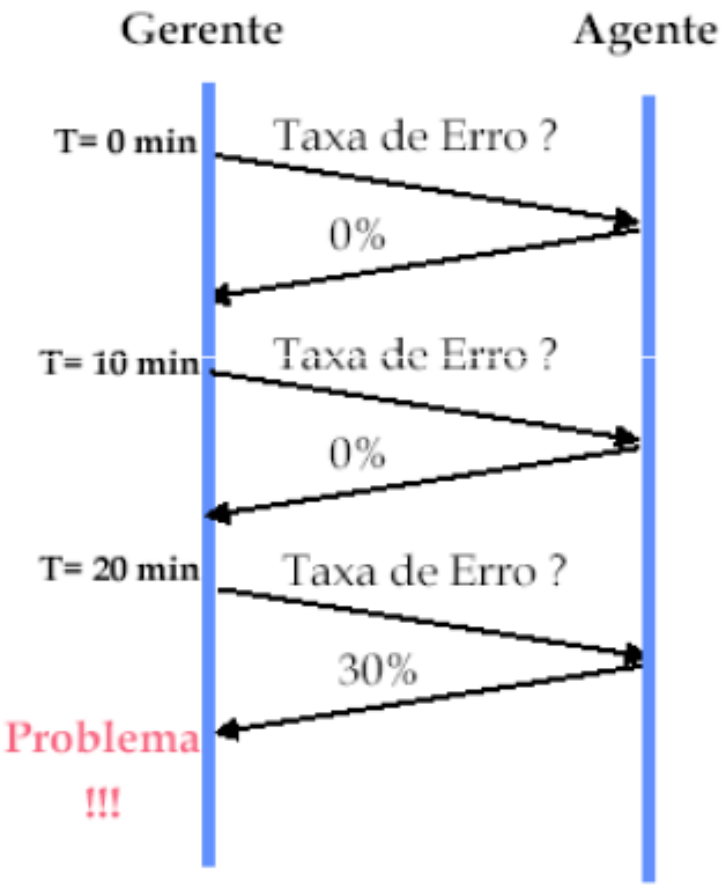
Modo evento

SNMP – Tipos de Mensagens

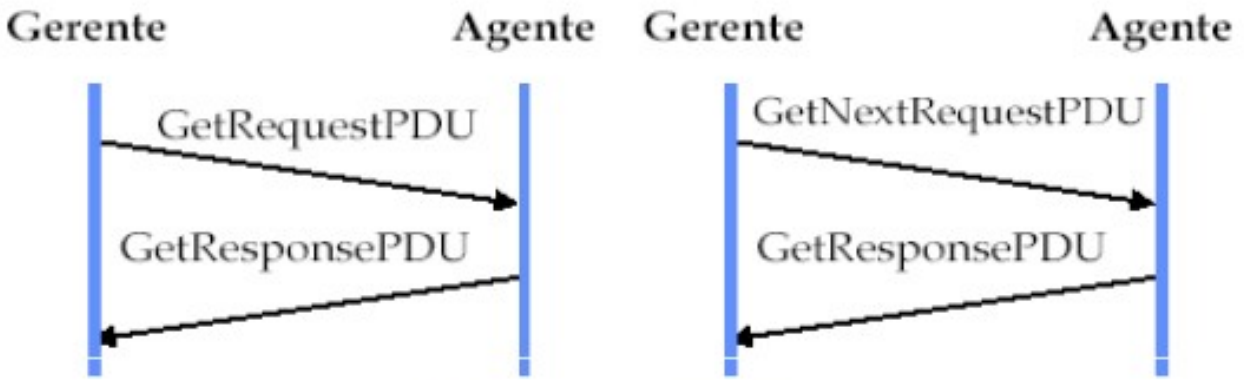
Tipo de Mensagem	Função
GetRequest GetNextRequest GetBulkRequest	gerente-para-agente: "me envie dados" (instância,próximo na lista, bloco)
InformRequest	gerente-para-gerente: eis o valor da MIB
SetRequest	gerente-to-agente: define valor da MIB
Response	agente-para-gerente: valor, resposta ao pedido
Trap	gerente-para-agente: informa gerenciador de evento excepcional

- SNMPv1
- Protocolo do tipo Request-Response, não-orientado à conexão
  - Utilização do UDP
- Operações atômicas
  - Se o gerente enviar um pedido de leitura dos valores de uma lista de objetos e um dos objetos não puder ser lido, não será devolvido o valor de nenhum dos objetos
- Gerenciamento Centralizado
  - Define apenas operações entre gerente e agentes
  - Não são definidas primitivas de comunicação entre gerentes

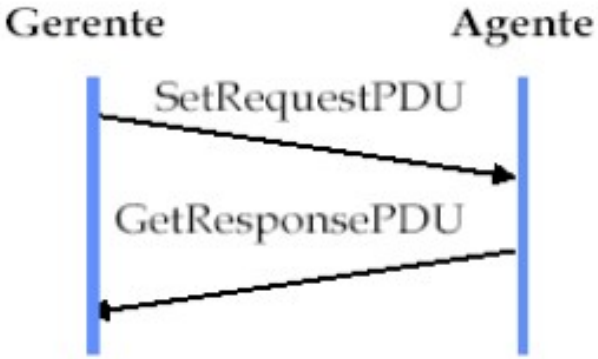
- Mecanismo básico: Polling
  - Para obter informações de recursos gerenciados, o gerente deve periodicamente requisitar o valor dos objetos da MIB e, baseado nestes valores, decidir se o dispositivo está operando normalmente



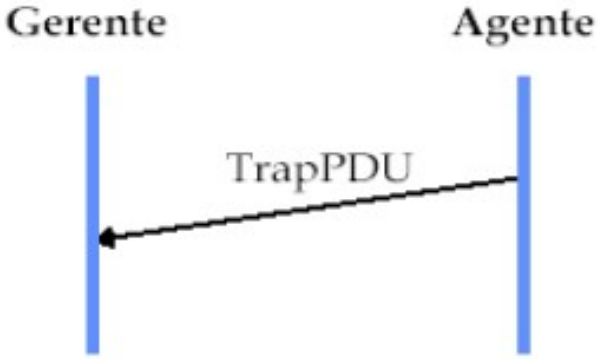
SNMP – Operações de Protocolo



(a) Leitura de Dados



(b) Atualização de Dados



(c) Notificação

Arquitetura SNMP e seus componentes

SNMP Message Field Definitions, General Message Format and Message Sections

(Page 3 of 3)

General PDU Format

The fields in each PDU depend on the PDU type, but can again be divided into the following general substructure:

- **PDU Control Fields:** A set of fields that describe the PDU and communicate information from one SNMP entity to another.
- **PDU Variable Bindings:** A set of descriptions of the MIB objects in the PDU. Each object is described as a "binding" of a name to a value.

Each PDU will follow this general structure, which is shown in Figure 276, differing only in the number of control fields, the number of variable bindings, and how they are used. In theory, each PDU could have a different message format using a distinct set of control fields, but in practice, most PDUs in a particular SNMP version use the same control fields (though there are exceptions.)

SNMP – Formato Geral de Mensagem

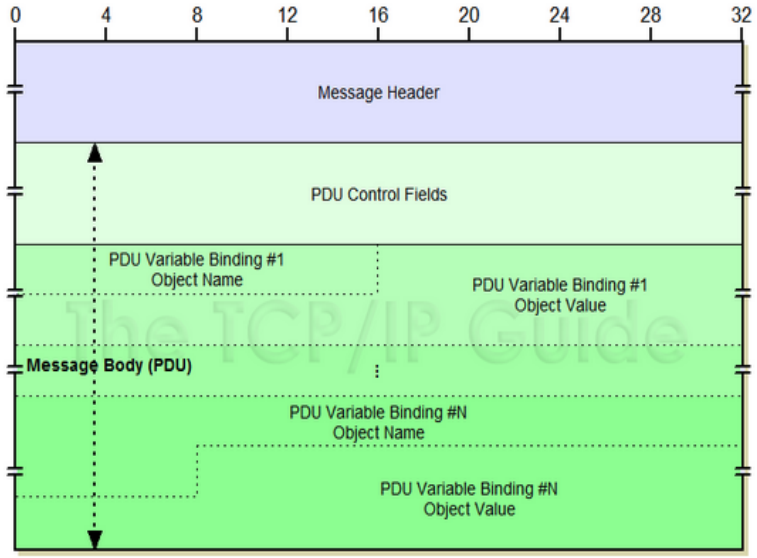


Figure 276: SNMP General Message Format

Table 211: SNMP Variable Binding Format

Subfield Name	Syntax	Size (bytes)	Description
Object Name	Sequence of *Integer*	Variable	***Object Name:*** The numeric object identifier of the MIB object, specified as a sequence of integers. For example, the object sysLocation has the object identifier 1.3.6.1.2.1.1.6, so it would be specified as "1 3 6 1 2 1 1 6" using ASN.1
Object Value	Variable	Variable	***Object Value:*** In any type of "get" request, this subfield is a "placeholder"; it is structured using the appropriate syntax for the object but has no value (since the "get" request is asking for that value!) In a "set" request (SetRequest-PDU) or in a reply message carrying requested data (GetResponse-PDU or Response-PDU), the value of the object is placed here.

Key Concept: The general format of SNMP messages consists of a *message header* and a *message body*. The body of the message is also called the *protocol data unit* or *PDU*, and contains a set of PDU *control fields* and a number of *variable bindings*. Each variable binding describes one MIB object and consists of the object's name and value.

Each variable binding describes one MIB object. The binding consists of a pair of subfields, one specifying the name of the object in standard SNMP object identifier notation, and one its value, formatted to match the object's SMI syntax. For example, if the object is of type *Integer*, the value field would be 4 bytes wide and contain a numeric integer value. Table 211 describes the subfield format for each PDU variable binding.

Créditos:
http://www.tcpipguide.com/free/t_SNMPMessageFieldDefinitionsGeneralMessageFormatand-3.htm

Arquitetura SNMP e seus componentes

SNMP Version 1 (SNMPv1) Message Format

(Page 1 of 3)

The SNMP general message format was, of course, first used to define the format of messages in the original SNMP Protocol, SNMP version 1 (SNMPv1). This first version of SNMP is probably best known for its relative simplicity, compared to the versions that followed it. This is reflected in its message format, which is quite straight-forward.

The general message format in SNMPv1 is a "wrapper" consisting of a small header and an encapsulated PDU. Not very many header fields were needed in SNMPv1 because the community-based security method in SNMPv1 is very rudimentary. Thus, the short overall format for SNMPv1 messages shown in Table 212 and Figure 277.

Table 212: SNMP Version 1 (SNMPv1) General Message Format

Field Name	Syntax	Size (bytes)	Description
Version	*Integer*	4	***Version Number:*** Describes the SNMP version number of this message; used for ensuring compatibility between versions. For SNMPv1, this value is actually ***0***, not 1.
Community	*Octet String*	Variable	***Community String:*** Identifies the SNMP community in which the sender and recipient of this message are located. This is used to implement the simple SNMP community-based security mechanism
PDU	—	Variable	***Protocol Data Unit:*** The PDU being communicated as the body of the message.

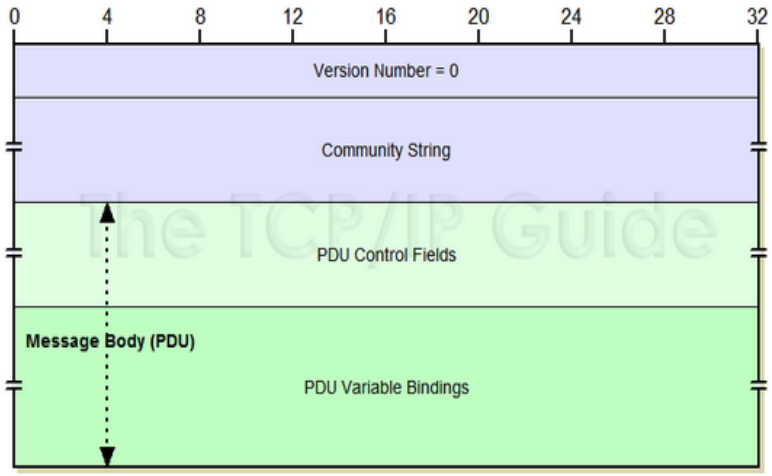


Figure 277: SNMP Version 1 (SNMPv1) General Message Format

SNMP – Cabeçalho e PDU

Créditos:
http://www.tcpipguide.com/free/t_SNMPVersion1SNMPv1MessageFormat.htm

Arquitetura SNMP e seus componentes

SNMP Version 1 (SNMPv1) Message Format

(Page 2 of 3)

SNMPv1 PDU Formats

All of the PDUs in SNMPv1 have the same format, with one exception: *Trap-PDU*. The exact semantics of each field in the PDU depends on the particular message. For example, the *ErrorStatus* field only has meaning in a reply and not a request, and object values are used differently in requests and replies as well.

SNMPv1 Common PDU Format

Table 213 and Figure 278 show the common format for most of the SNMPv1 PDUs: *GetRequest-PDU*, *GetNextRequest-PDU*, *SetRequest-PDU* and *GetResponse-PDU*:

Table 213: SNMP Version 1 (SNMPv1) Common PDU Format

Field Name	Syntax	Size (bytes)	Description
PDU Type	*Integer (Enumerated)*	4	***PDU Type:*** An integer value that indicates the PDU type: *PDU Type Value* / PDU Type 0 – *GetRequest-PDU* 1 – *GetNextRequest-PDU* 2 – *GetResponse-PDU* 3 – *SetRequest-PDU*
Request ID	*Integer*	4	***Request Identifier:*** A number used to match requests with replies. It is generated by the device that sends a request and copied into this field in a *GetResponse-PDU* by the responding SNMP entity.
Error Status	*Integer (Enumerated)*	4	***Error Status:*** An integer value that is used in a *GetResponse-PDU* to tell the requesting SNMP entity the result of its request. A value of zero indicates that no error occurred; the other values indicate what sort of error happened: *Error Status Value* / Error Code / Description 0 / *noError* / No error occurred. This code is also used in all request PDUs, since they have no error status to report. 1 / *tooBig* / The size of the *GetResponse-PDU* would be too large to transport. 2 / *noSuchName* / The name of a requested object was not found. 3 / *badValue* / A value in the request didn't match the structure that the recipient of the request had for the object. For example, an object in the request was specified with an incorrect length or type. 4 / *readOnly* / An attempt was made to set a variable that has an *Access* value indicating that it is read-only. 5 / *genErr* / An error other than one of the preceding four specific types occurred.
Error Index	*Integer*	4	***Error Index:*** When *Error Status* is non-zero, this field contains a pointer that specifies which object generated the error. Always zero in a request.
Variable Bindings	Variable	Variable	***Variable Bindings:*** A set of name-value pairs identifying the MIB objects in the PDU, and in the case of a *SetRequest-PDU* or *GetResponse-PDU*, containing their values. See the general message format topic for more on these bindings.

SNMP – Controle PDU e Payload

Créditos:
http://www.tcpipguide.com/free/t_SNMPVersion1SNMPv1MessageFormat-2.htm

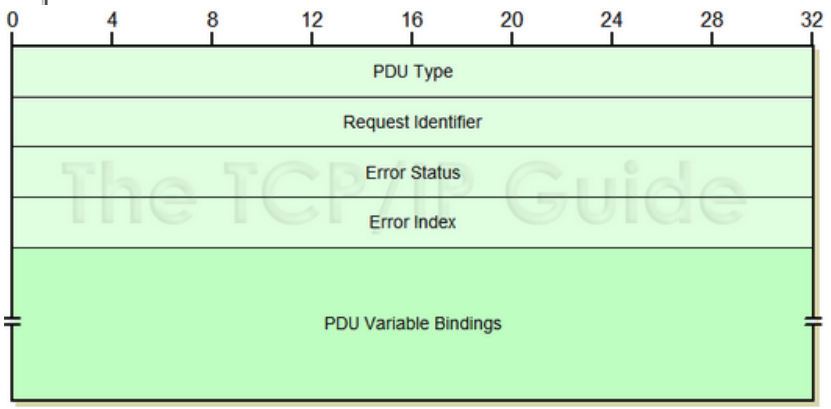


Figure 278: SNMP Version 1 (SNMPv1) Common PDU Format

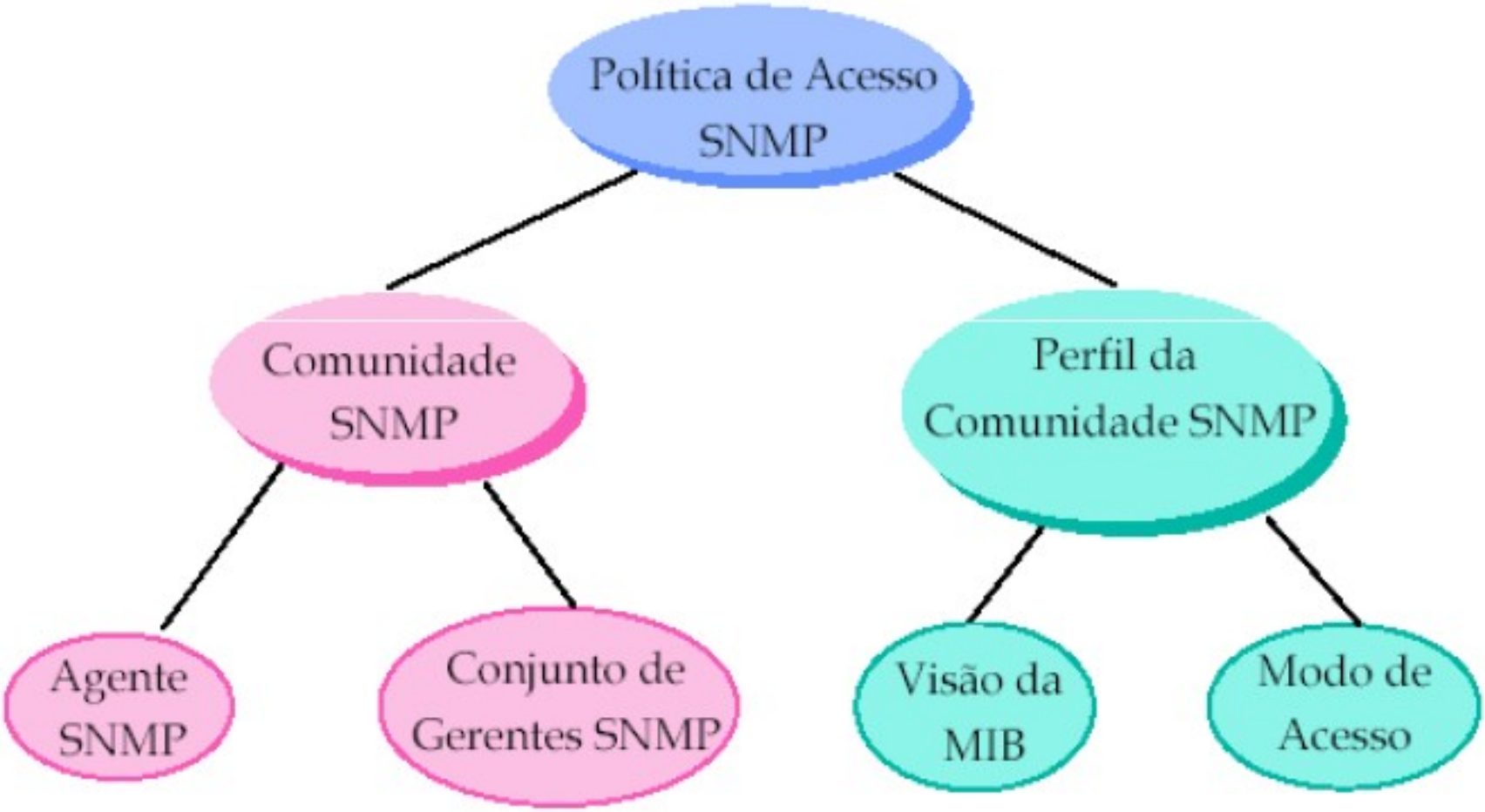
SNMP – Campos de Mensagem

Campo	Descrição
version	• versão do SNMP
community	• informação utilizada como senha para autenticar a mensagem SNMP.
request id	• identificação de pedido.
error status	• indica o tipo de erro ocorrido no tratatamento do pedido correspondente (e.g., 0 - sem erro, 1 - *request* PDU muito longa... (4) - objeto *read-only*).
error index	• informação adicional relativa ao erro ocorrido, contendo a indicação da variável da lista que causou do erro)
variablebindings	• lista de variáveis e seus valores.
enterprise	• tipo do objeto que origou o *trap*.
agent addr	• endereço IP do objeto que gerou o *trap*.
generic trap	• tipo de *trap*
specific trap	• código que especifica a natureza do *trap*.
time-stamp	• tempo decorrido entre a última reniciação da entidade que emitiu o *trap* e a generação de *trap*.

SNMP – Segurança

- A **política de acesso** SNMP é baseada na definição de:
  - **Comunidade SNMP:** define a relação existente entre um agente e um conjunto de gerentes. Cada comunidade deve possuir um nome, que deve ser fornecido em todas as operações de *get* and *set*.
  - **Perfil da Comunidade:** associa à comunidade:
    - **Visão da MIB:** corresponde a um subconjunto da MIB que pode ser acessada pela comunidade (dificilmente implementada)
    - **Modo de Acesso:** especifica o modo de acesso (*read-only* ou *read-write*) permitido.

SNMP – Segurança



SNMP – Política de Acesso

- roteador do fabricante XXX
  - Comunidades suportadas: 2
  - Comunidade 1: public
    - Permissão de acesso: qualquer gerente
    - MIB View: todas as variáveis
    - Modo de Acesso: RO (Read-Only)
  - Comunidade 1: private
    - Permissão de acesso: qualquer gerente
    - MIB View: todas as variáveis
    - Modo de Acesso: RW (Read-Write)

SNMP – Política de Acesso

- switch do fabricante YYY
  - Comunidades suportadas: 2
  - Comunidade 1: public
    - Permissão de acesso: qualquer gerente
    - MIB View: estatísticas das interfaces
    - Modo de Acesso: RO (Read-Only)
  - Comunidade 1: private
    - Permissão de acesso: g1 (ip: 10.0.2.4), g2 (ip: 10.1.4.9)
    - MIB View: todas as variáveis
    - Modo de Acesso: RW (Read-Write)

SNMP – Política de Acesso

- **Objeto Modelado – Porta de Rotador**

  - **Atributos:**

•ID_Porta	GET	0,1,2
•Tipo_Porta	GET	Ethernet/Serial
•Taxa_Transf_Max	GET	9600,19200,…
•Taxa_Trans_Efet	GET	0 – Taxa_Trafs_Max
•Taxa_Erros	GET	0 – Taxa_Trafs_Max
•Reset_Estat	SET	on/off
•Tempo_Ativa	GET	0 …999999 (min)
•Porta_Ativa	GET/SET	on/off

SNMPv1 – Limitações

- Inadequado para redes muito grandes devido ao problema de desempenho decorrente da utilização do mecanismo de *polling* (operações de *get* e *set*).
- Inadequado para transferir grande volume de dados.
- Baixa confiabilidade dos *traps*: inexistência de reconhecimento e operação sobre o UDP.
- Segurança a nível básico (mecanismo de autenticação).
- Inexistência de suporte à comunicação gerente-gerente.

Dúvidas?