



INTRODUÇÃO À GERÊNCIA DE REDES DE COMPUTADORES

GERÊNCIA DE REDES

- ⌘ Gerência $\bar{\text{O}}$ Métodos para planejar, configurar, controlar, monitorar, corrigir falhas e administrar redes de computadores
- ⌘ Modelo Gerente-Agente
 - w nós gerenciáveis – 1 ou mais nós gerenciáveis
 - w estrutura de informação de gerenciamento – SMI (regras de descrição dos objetos)
 - w base de informações de gerenciamento – MIB (conjunto de informações de gerenciamento)
 - w operações de gerenciamento – primitivas para manipulação via usuários.

PROTOCOLO DE GERENCIAMENTO DA INTERNET (SNMP)

- ⌘ Protocolo assíncrono de requisição e resposta (*request/response*)
- ⌘ Único requisito de transporte do SNMP é um serviço de transporte sem conexão
- ⌘ Permite a uma NMS centralizada consultar agentes para obter e modificar informações nas MIBs

3

PROTOCOLO DE GERENCIAMENTO DA INTERNET (SNMP)

- ⌘ SNMP é o padrão para protocolo de gerência mais popular.
- ⌘ Foi o padrão adotado por vários fabricantes e operadoras.
- ⌘ Define como funciona a arquitetura de gerenciamento de redes TCP/IP.
- ⌘ É simples para ser implementado em todo tipo de equipamentos e flexível o bastante para aceitar futuras modificações.
- ⌘ É o protocolo de gerenciamento da arquitetura TCP/IP. Define como funciona a arquitetura de gerenciamento Internet.

4

PROTOCOLO DE GERENCIAMENTO DA INTERNET (SNMP)

- Está intimamente ligado à forma como as informações de gerenciamento estão organizadas. Apresenta uma SMI própria.

5

SMI SNMP (RFC 1155)

sysUptime OBJECT-TYPE

SYNTAX Time-Ticks

ACCESS read-only

STATUS mandatory

DESCRIPTION

.The time (in hundredths of a second) since the network management portion of the system was last re-initialized..

::= { system 3 }

**Exemplo
Declaração de objeto**

6

SMI SNMP (RFC 1155)

- ☞ As variáveis e seus valores estão diretamente relacionadas com a realidade do equipamento.
- ☞ Um objeto é definido segundo a macro ASN.1 OBJECT-TYPE.
- ☞ As operações sobre objetos da MIB são restritas a leituras e escritas.

7

TIPOS DE DADOS SNMP

INTEGER (signed 32-bit integer)

OCTET STRING

OBJECT IDENTIFIER (OID)

NULL (valor Null)

IpAddress (OCTET STRING de tamanho 4)

Counter (unsigned 32-bit integer)

Gauge (unsigned 32-bit integer)

TimeTicks (unsigned 32-bit integer)

Opaque (tipos não usados em SNMPv1)

Outros: *DateAndTime, DisplayString, MacAddress, PhysAddress, TimeInterval, TimeStamp, TruthValue, VariablePointer*. todos são textual conventions usados como tipos de dados

8

SNMP / GERÊNCIA DE REDES

☞ Gerenciamento de rede:

“Inclui a disponibilização, a integração e a coordenação de elementos de hardware, software e humanos para monitorar, testar, consultar, configurar, analisar, avaliar e controlar os recursos da rede, e de elementos, para satisfazer às exigências operacionais, de desempenho e de qualidade de serviço em tempo real a um custo razoável.”

[Saydam, 1996]

9

SNMP / GERÊNCIA DE REDES

☞ A *International Organization for Standardization* (ISO) definiu as principais áreas de gerenciamento de rede.

☞ A divisão proposta engloba as seguintes áreas:

- Gerência de falhas;
- Gerência de contabilização;
- Gerência de configuração;
- Gerência de segurança;
- Gerência de desempenho.

10

PROTOCOLOS DE GERENCIAMENTO DE REDES

SNMP:

O SNMP é um protocolo de gerência utilizado para obter informações de servidores SNMP - agentes espalhados em uma rede baseada na pilha de protocolos TCP/IP.

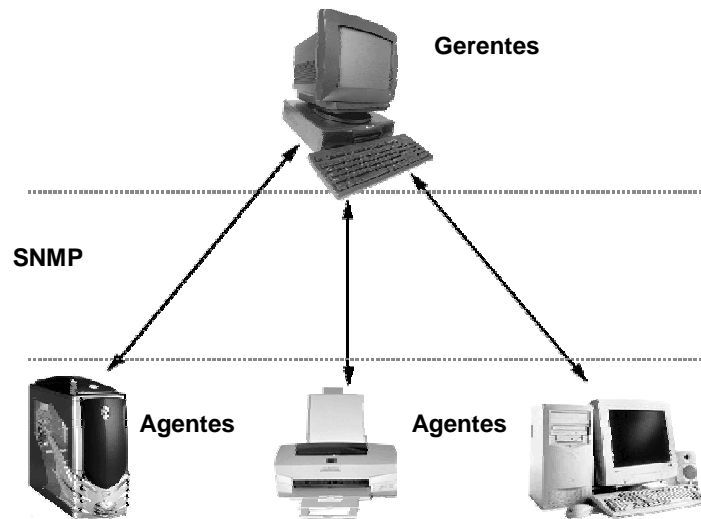
11

SNMP

No SNMP os dados são obtidos através de requisições de um gerente a um ou mais agentes utilizando os serviços do protocolo de transporte UDP, para enviar e receber suas mensagens através da rede.

12

SNMP / MODELO DE GERÊNCIA SNMP



13

SNMP / TCP/IP

O TCP/IP é o acrônimo para “Transmission Control Protocol / Internet Protocol”, e serve para caracterizar a família de protocolos utilizada nas comunicações de computadores.

14

SNMP / TCP/IP

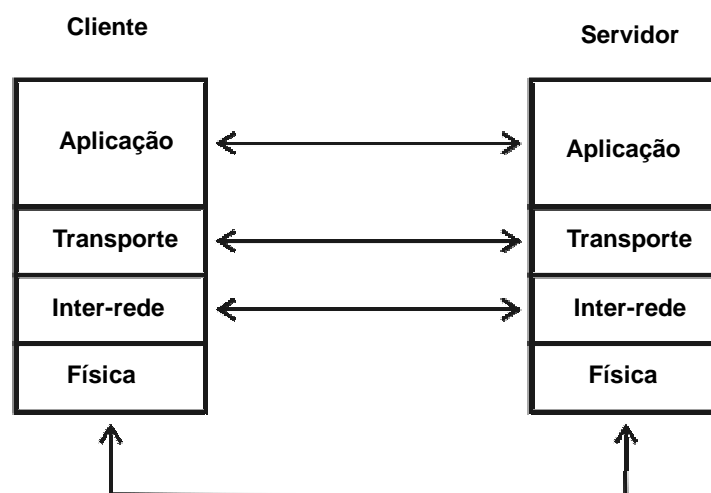
É apresentado através de um modelo de 4 camadas que descreve o caminho que a informação percorre por uma rede.

São elas:

- Camada de aplicativo;
- Camada de transporte;
- Camada de Inter-rede;
- Camada Física;

15

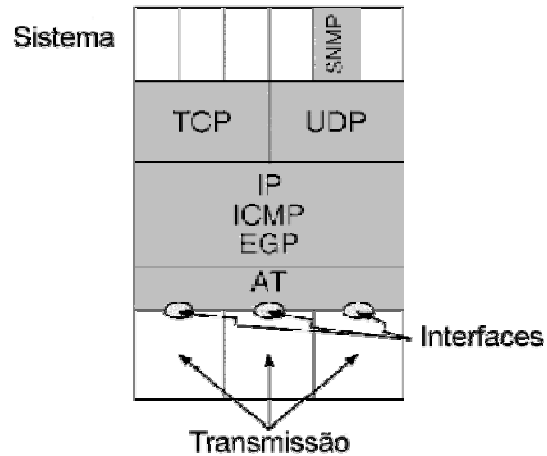
SNMP / PILHA TCP/IP



16

SNMP / LOCALIZAÇÃO SNMP

Localização do protocolo SNMP na pilha TCP/IP



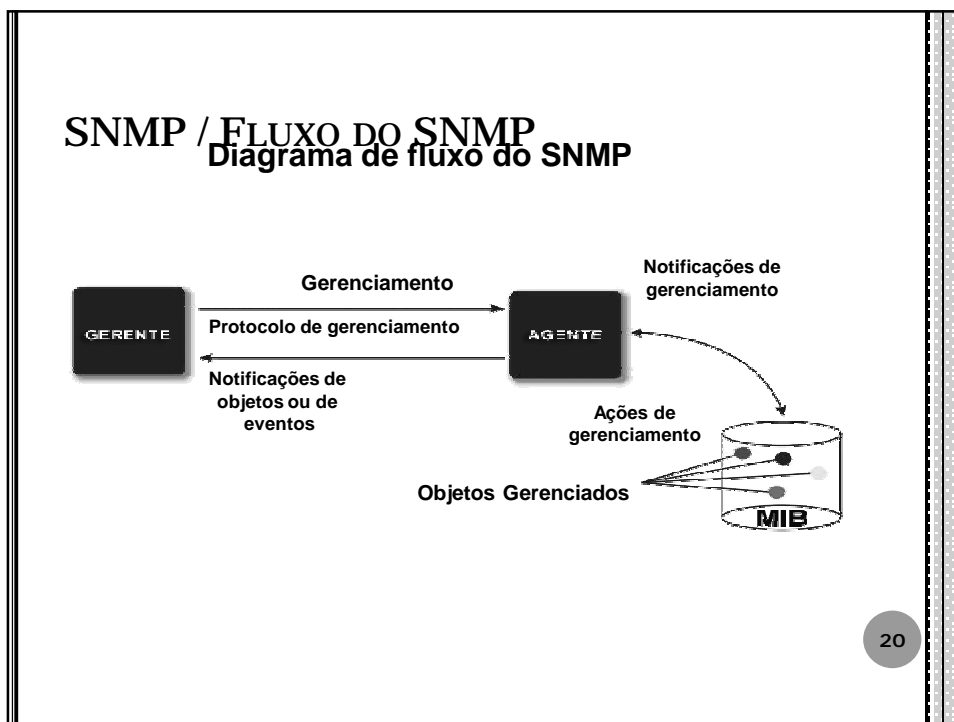
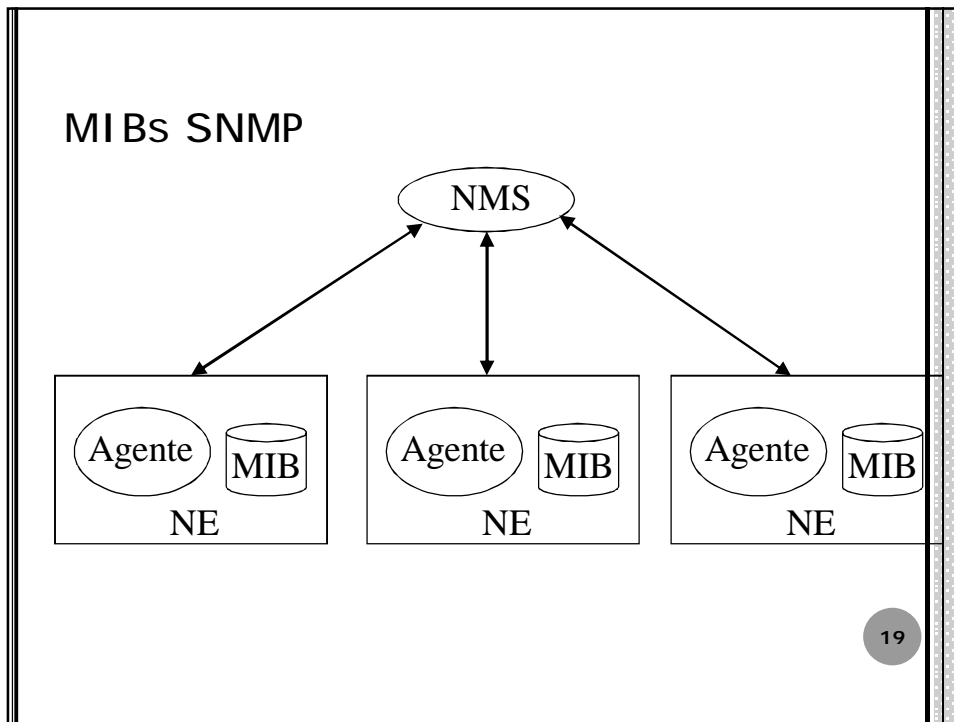
17

SNMP / CARACTERÍSTICAS SNMP

O modelo de gerenciamento SNMP para redes TCP/IP, é composto pelos seguintes elementos:

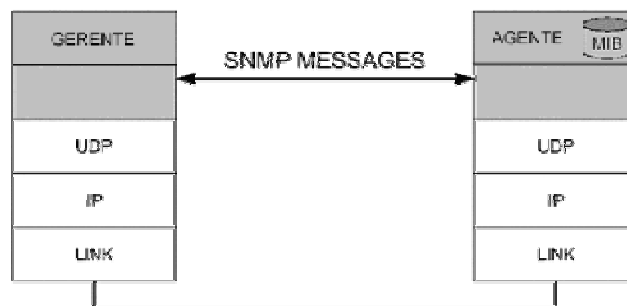
- ☛ Estação de gerenciamento (NMS);
- ☛ Agente de Gerenciamento;
- ☛ Base de Informações (MIB);
- ☛ SNMPv1, SNMPv2, SNMPv3;

18



SNMP / GERENTE X AGENTE

Relacionamento entre gerente e agente baseado no modelo TCP/IP



21

SNMP / OPERAÇÕES DO SNMP

☞ Get;

- Utilizada para ler o valor de uma variável; o gerente solicita que o agente obtenha o valor da variável;

☞ Set;

- Utilizada para alterar o valor da variável; o gerente solicita que o agente faça uma alteração no valor de uma variável;

☞ Trap;

- Utilizada para comunicar um evento; o agente comunica ao gerente o acontecimento de um evento previamente determinado.

22

COMUNICAÇÃO SNMP

Gerente

Agente

23

PROTOCOLO DE GERENCIAMENTO DA INTERNET (SNMP)

- ☛ **Consiste de 3 tipos de operações:**
 - **GET:** a NMS recupera uma informação específica do agente
 - **SET:** a NMS altera uma informação específica no agente
 - **TRAP:** um agente reporta um evento para a NMS

24

SNMP / MIB

Definição:

MIB – Management Information Base

“Conjunto de objetos gerenciados, que abrange as informações necessárias para a gerencia da rede.”

25

SNMP / OBJETOS GERENCIADOS

“Visão abstrata de um recurso real do sistema.”

Assim são os objetos gerenciados:
todos os recursos que devem ser gerenciados.

Por exemplo: Consumo de banda, Status de operação, colisões de pacotes...

26

SNMP / TIPOS DE MIB

Basicamente são definidos três tipos de MIBs:

☞ MIB II:

Estão os objetos usados para obter informações específicas dos dispositivos de rede.

☞ MIB experimental:

É aquela em que seus objetos ainda estão sendo pesquisados pela IAB (*Internet Architecture Board*)

☞ MIB privada:

É aquela que contém objetos definidos por outras organizações.

27

MIB

☞ MIB-I: SNMP foi desenvolvido primariamente para gerenciar redes TCP/IP, assim a primeira MIB padronizada continha informações específicas a TCP/IP como:

- número de interfaces de rede com seus endereços IP
- contadores de datagramas UDP
- tabela de conexões TCP ativas
- Entre outros...

28

MIB

- ☞ MIB-II: Esta MIB, total ou parcialmente, é normalmente implementada em produtos comerciais. Contém objetos relacionados com características normalmente encontradas nos equipamentos ligados em redes
- ☞ A MIB-II é a MIB implementada por padrão em todos os agentes com suporte a SNMP.

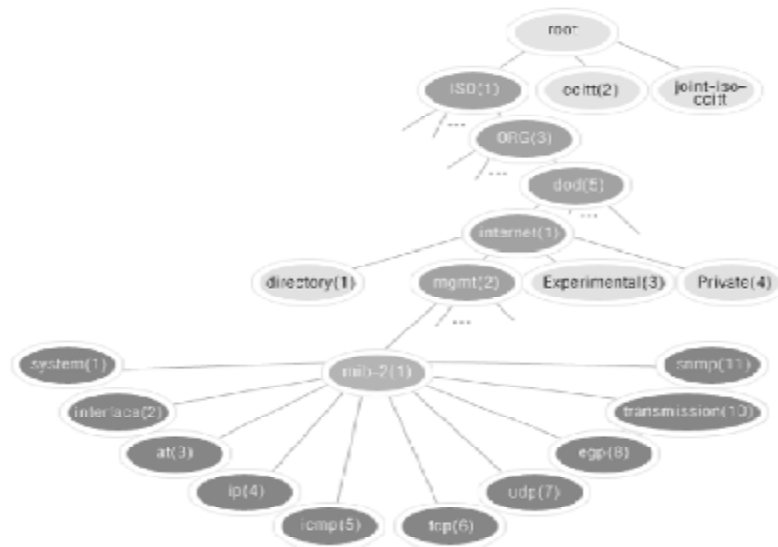
29

SNMP / SMI E ASN.1

- ☞ As regras de construção das estruturas da MIB são descritas através da SMI – *Structure of Management Information*.
- ☞ Cada objeto da MIB é especificado de acordo com a ASN.1 – *Abstract Syntax Notation One* e contém: *Nome, identificador, sintaxe, definição e acesso*.

30

ESTRUTURA LÓGICA DA MIB



ESTRUTURA LÓGICA DA MIB

- ☛ A partir da raiz, temos 3 ramos:
 - ITU-T (CCITT) [0]
 - ISO [1]
 - Joint ITU-T e ISO [2]
- ☛ O ramo iso por sua vez se ramifica em:
 - Standard [0]
 - Registration Authority [1]
 - Member-body [2]
 - Identified-Organization [3]

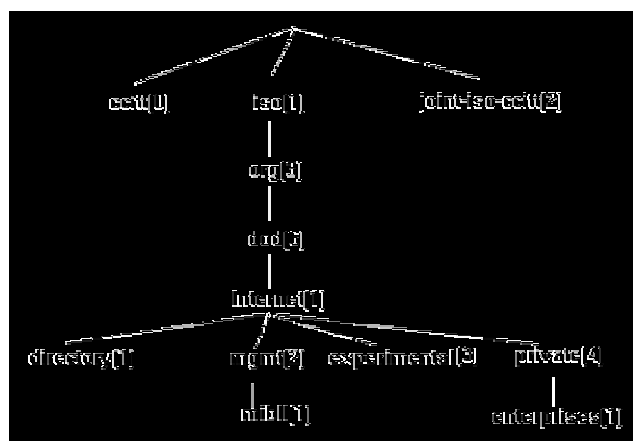
ESTRUTURA LÓGICA DA MIB

- ☛ Dentro de [3], temos o Department of Defense (DoD) [6] e abaixo o IAB (Internet Architecture Board) [1], assim iso.identified-organization.DoD.IAB == 1.3.6.1
- ☛ Este normalmente é o prefixo para todos os objetos de interesse na área de gerenciamento.

33

SNMP / ESTRUTURA LÓGICA DA MIB

Arvore hierárquica definida pela ISO



34

SNMP / ESTRUTURA LÓGICA DA MIB

- ☞ Os inteiros indicam a seqüência de nodos ao longo de um caminho iniciando no topo da árvore.
- ☞ A árvore é estática, significando que os nodos são determinados quando a MIB é designada.
- ☞ Em acréscimo, para prover identificação única de tipos de objetos, a estrutura da árvore mostra grupos de objetos abaixo de uma única sub-árvore. Um nome próprio (correspondente ao identificador do objeto) é também associado ao tipo de objeto.

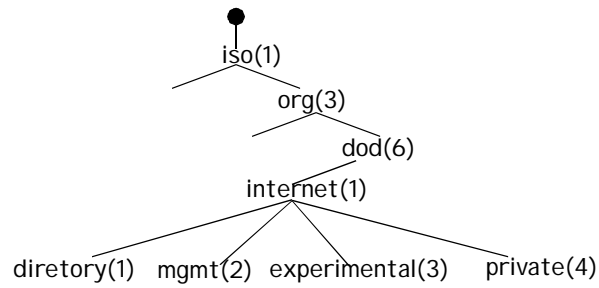
35

SNMP / ESTRUTURA LÓGICA DA MIB

- ☞ A sintaxe define a estrutura de dados abstrata. Um subconjunto da ASN.1 é utilizada para definição de tipos de dados e suas propriedades.
- ☞ A codificação de objetos segue as regras básicas de codificações com ASN.1.
- ☞ Dados gerenciados são indexados pelas folhas, localizadas na base da árvore

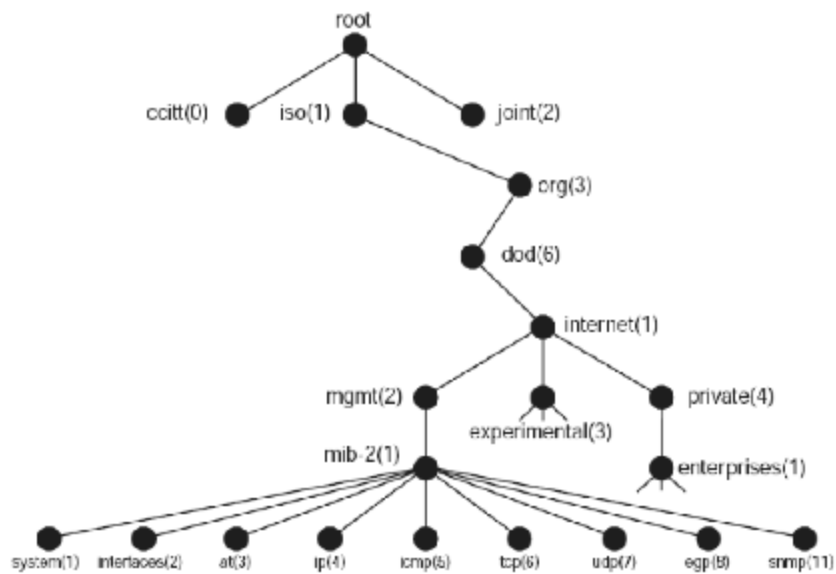
36

ESTRUTURA EM ÁRVORE DA MIB SNMP



37

ÁRVORE DA MIB II



ÁRVORE DA MIB II

☛ Abaixo do ramo Internet, tem-se:

- directory (1): uso futuro com serviços de diretórios OSI
- mgmt (2): objetos definidos por documentos do IAB
- experimental (3): objetos para testes e pesquisas
- private (4): objetos definidos por grupos ou organizações, como fabricantes por exemplo

☛ Logo abaixo do ramo mgmt (2) tem-se a MIB-II, mib-2(1).

39

GRUPOS DA MIB

Group	Objects for	#
System	Basic system information	7
Interfaces	Network attachments	23
AT	Address translation	3
IP	Internet protocol	42
ICMP	Internet control message protocol	26
TCP	Transmission control protocol	19
UDP	User datagram protocol	7
EGP	Exterior gateway protocol	18
SNMP	SNMP applications entities	39
Legend: # = Number of objects in the group		

GRUPOS DA MIB

- ☛ **system**: informações gerais do agente/equipamento (descrição, up time, pessoa de contato)
- ☛ **interfaces**: descrição das interfaces do equipamento, endereços físicos e contadores
- ☛ **at** (address translation): mapeamento de endereços físicos/rede
- ☛ **ip**: tabelas de endereçamentos e contadores
- ☛ **icmp**: contadores ICMP
- ☛ **tcp**: tabela de conexões TCP e contadores
- ☛ **udp**: tabela UDP e contadores
- ☛ **egp**: tabela de vizinhos EGP e contadores
- ☛ **snmp**: registros estatísticos das mensagens SNMP

41

GRUPO DA MIB

- ☛ Houve extensões da MIB a partir do número 13
 - MIBs privadas – cada fabricante possui seu próprio número
 - Novo grupo – **transmission** - onde ficam abaixo somente grupos de objetos relacionados com tecnologias de transmissão (tecnologias de rede).

42

DECLARAÇÕES DAS MIBS

- ☛ **MODULE-IDENTITY**
- ☛ **OBJECT-IDENTITY**
- ☛ **OBJECT-TYPE**
- ☛ **NOTIFICATION-TYPE**
- ☛ **TEXTUAL-CONVENTION**
- ☛ **OBJECT-GROUP**
- ☛ **MODULE-COMPLIANCE**
- ☛ **AGENT-CAPABILITIES**

43

DECLARAÇÕES DAS MIBS

- ☛ A SMIv2 criou várias MACROS para melhorar as declarações de módulos de MIB.
- ☛ **MODULE-IDENTITY** . define, através de uma seção de identificação comum, um módulo de MIB
- ☛ **OBJECT-TYPE** . sintaxe e semântica de um objeto gerenciado
- ☛ **OBJECT-IDENTITY** . texto adicional sobre um objeto gerenciado

44

DECLARAÇÕES DAS MIBS

- ☛ NOTIFICATION-TYPE . sintaxe de um notificação SNMPv2 (trap). Substituiu a macro TRAP-TYPE, usada em SNMPv1
- ☛ TEXTUAL-CONVENTION . sintaxe refinada de um tipo de dado (melhora a compreensão de um tipo de dado específico).
- ☛ OBJECT-GROUP . define um conjunto de objetos relacionados
- ☛ MODULE-COMPLIANCE . lista os módulos de MIB obrigatórios ou opcionais
- ☛ AGENT-CAPABILITIES . detalha uma implementação particular
- ☛ Várias das MIB's criadas com a SMIv1, foram relançadas sob a SMIv2, inclusive vários dos grupos de objetos da MIB-II

45

UM MIB BROWSER

46

UM MIB BROWSER

- ☛ Um MIB browser é uma aplicação que permite a obtenção (e alteração) de variáveis numa MIB de um agente SNMP. Este utilitário oferece uma interface adequada de visualização de objetos e seus valores e executa as operações SNMP necessárias para obter informações e alterá-las nos agentes.
- ☛ Uma opção livre é:
 - mbrowser - <http://www.kill-9.org/mbrowse/>
 - O Webgetif (<http://www.webgetif.com/>) é uma versão grátis com interface WEB.
- ☛ Existem várias opções comerciais que disponibilizam versões TRIAL.

47

ATIVIDADES PRÁTICAS

- ☛ **Uso do comando snmptranslate para mostrar detalhes do grupo system da MIB-II**
- ☛ **Uso de um MIB Browser**
- ☛ **Validação de uma MIB**

48

SNMP / FERRAMENTAS DE GERENCIA SNMP

- ☞ **Comerciais: HpOpenview, What's up...**
- ☞ **Dominio público: MRTG, Cacti...**
- ☞ **Ambiente de desenvolvimento: PHP;**

49

SNMP / USO DO SNMPv1

- ☞ **Gerenciamento de dispositivos embarcados;**
- ☞ **O mundo é IP;**
- ☞ **Suporte em CLP's, terminais de comunicação;**

50

SNMP / VANTAGENS

- ⌘ O agente SNMP é pequeno e simples;
- ⌘ Flexibilidade: Construção de MIB's definida pelo usuário.
- ⌘ Uso de um protocolo bem definido;
- ⌘ Disponibilidade de ferramentas da área de redes.

51

SNMP / DESVANTAGENS

- ⌘ Não é adequado para redes muito grandes;
- ⌘ Traps SNMP não são reconhecidos;
- ⌘ O padrão SNMP básico provê somente autenticação trivial;
- ⌘ Não suporta comunicação manager-to-manager;

52

GERENCIAMENTO TCP/IP

- ☉ **SNMP - *Simple Network Management Protocol***
 - RFC1155 *Structure and Identification of Management Information for TCP/IP-based internets*
 - RFC 1156 - *Management Information Base Network Management of TCP/IP-based internets*
 - RFC 1157 - *A Simple Network Management Protocol*
 - RFC 1213 - *Management Information Base Network Management of TCP/IP-based internets: MIB-II*
- ☉ **RMON - *Remote Network Monitoring***
 - RFC1271 e depois RFC 1757

53

GERENCIAMENTO TCP/IP

- ☉ **SNMPv2**
 - RFC1442 *Structure of Management Information for Version 2 of SNMP*
 - RFC1448 *Protocol Operations for Version 2 of SNMP*
- ☉ **SNMPv3**
 - 1998
 - Principal característica: Segurança

54

PROTOCOLO SNMP

- ☛ **Simple Network Management Protocolo**

- ☛ **Estrutura de Informação de Gerência (SMI) - ASN.1 (Abstract Syntax Notation One) / Macro OBJECT-TYPE**

- ☛ **Protocolo - ASN.1 / BER (Basic Encoding Rules) via UDP/IP**

55

ASN.1

- ☛ **Linguagem formal para definição de sintaxe abstrata (ISO)**
- ☛ **SNMP usa um subconjunto de tipos ASN.1, bem como a macro OBJECT-TYPE para a especificação da MIB**
 - Integer
 - Octet String
 - Display String
 - Object Identifier
 - Sequence
 - Sequence of

56

BER

- ⌘ Regras que geram a sintaxe de transferência
- ⌘ Tipos codificados em três campos: rótulo, tipo e valor

ex.:

```

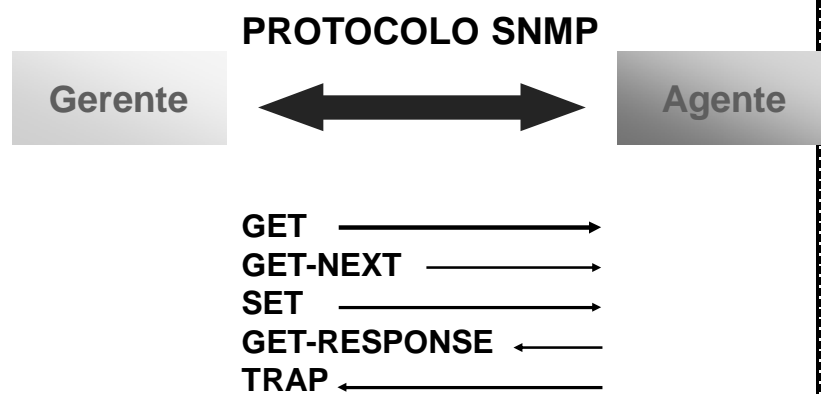
ex ::= sequence {
    nome OCTET STRING,
    idade INTEGER
}
    
```

dados: { adao, 45 }

dados codificados: 30 07 02 04 04 A D A O 02 01 45

57

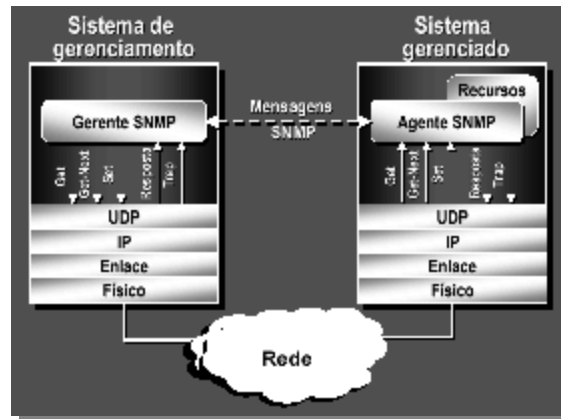
PROTOCOLO



58

O protocolo SNMP é transportado pelo protocolo UDP

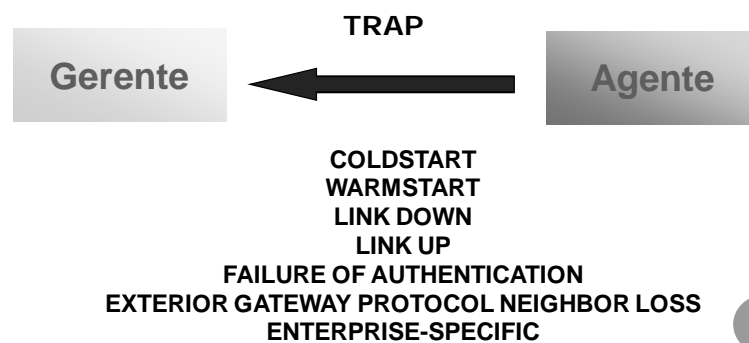
SNMP OVER UDP



59

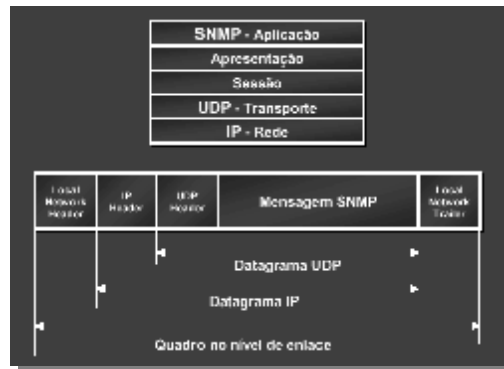
TRAPS

☛ Mensagens não solicitadas geradas por um agente SNMP



60

PDU's SNMP



61

GET, GET-NEXT, SET, GET-RESPONSE



62

ERROS

☞ Erros retornados por agentes SNMP

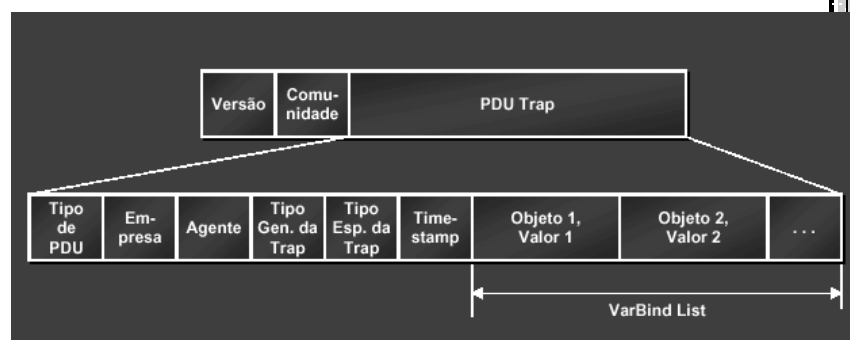
- 0 (noError)
- 1 (tooBig)
- 2 (noSuchName)
- 3 (badValue)
- 4 (readOnly)
- 5 (genError)

☞ Índice do Erro

- Indica a qual variável se refere o erro

63

TRAPS



64

OPERAÇÃO SNMP

⌘ ID do objeto + ID da instância

⌘ Objetos folha (.0)

- ex.: GET sysDescr.0
- ⌘ GET 1.3.6.1.1.1.1.0

⌘ Objetos como campo de uma tabelas (.chave)

- ex.: GET ipRouteNextHop.143.54.1.0
- ⌘ GET 1.3.6.1.1.1.5.7.143.54.1.0

65

OBJETOS RELEVANTES AO GERENCIAMENTO DE FALHAS

Grupo SYSTEM

sysDescr	descrição do sistema
sysLocation	localização física do sistema
sysContact	pessoa responsável pelo sist.
sysName	nome do sistema

66

OBJETOS RELEVANTES AO GERENCIAMENTO DE FALHAS

GRUPO INTERFACES

Dados sobre cada interface específico do dispositivo

ifTable tabela com informações sobre todos as interfaces
ifEntry linha com informações sobre uma interface
ifNumber número de interfaces

67

OBJETOS RELEVANTES AO GERENCIAMENTO DE CONFIGURAÇÃO

Grupo INTERFACES

ifDescr nome do interface
ifType tipo do interface
ifMTU máximo tamanho de datagrama
ifSpeed velocidade do interface (BPS)
ifAdminStatus up/down/test

68

OBJETOS RELEVANTES AO GERENCIAMENTO DE PERFORMANCE

ifInDiscards	taxa de entradas descartadas
ifOutDiscards	taxa de transmissões descartadas
ifInErrors	taxa de erros de entrada
ifOutErrors	taxa de erros em transmissões
ifInOctets	taxa de bytes recebidos
ifInUcastPkts	taxa de pacotes unidirecionados recebidos
ifOutUcastPkts enviados	taxa de pacotes unidirecionados
ifInNUcastPkts recebidos	taxa de pacotes multirecionados
ifOutNUcastPkts enviados	taxa de pacotes multirecionados
ifInUnknownProtos	taxa de pacotes recebidos com protocolo desconhecido
ifOutQLen	total de pacotes na fila de saída

69

EXEMPLO

```

maquina% snmpi -a routcv
snmpi> bulk system
snmpi: 3 rows retrieved in 0.643473 seconds during 8 iterations
snmpi: threads: at most 2 active, total of 6 created, and 5 did
nothing
snmpi: messages: 9 requests sent, along with 0 retries
snmpi:      9 responses rcvd, along with 0 duplicates
snmpi: timeouts: min=0.082 fin=0.082 max=2.000 seconds
partial results only...
row  system
1.0  "ROUTCV DEC WANrouter 500 V1.0.7"
2.0  1.3.6.1.4.1.36.2.15.4.2.1.1
3.0  3 days, 14 hours, 2 minutes, 42 seconds (30976200 timeticks)

```

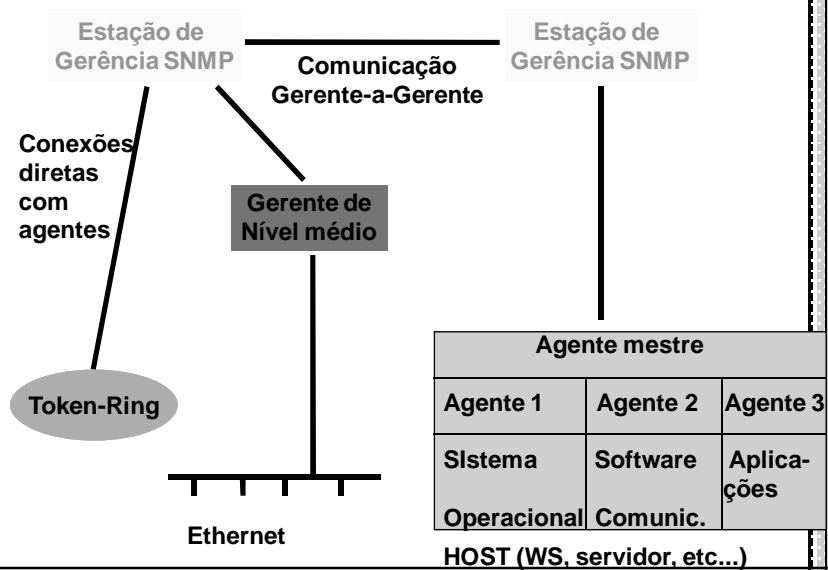
70

SNMPv2

- ⌘ Gerenciar recursos arbitrários e não apenas recursos de rede (aplicações, sistemas e comunicação gerente-a-gerente)
- ⌘ Continua simples e rápido
- ⌘ Incorpora segurança
- ⌘ Funciona sobre TCP/IP, OSI e outros protocolos
- ⌘ Interopera com plataformas SNMP
- ⌘ Gerenciamento hierárquico

71

SNMPv2



OPERAÇÕES SNMPv2

- ☛ GetRequest
- ☛ GetNextRequest
- ☛ SetRequest
- ☛ Response
- ☛ Trap
- ☛ GetBulkRequest
- ☛ InformRequest

73

SNMPv2

- ☛ SNMPv2 : utilização do protocolo *Manager-to-Manager*
- ☛ Alarmes e eventos
- ☛ *Party* : segurança
- ☛ Segurança
 - mecanismo de autenticação
 - privacidade (criptografia)
 - controle de acesso (por tipo de acesso)

74

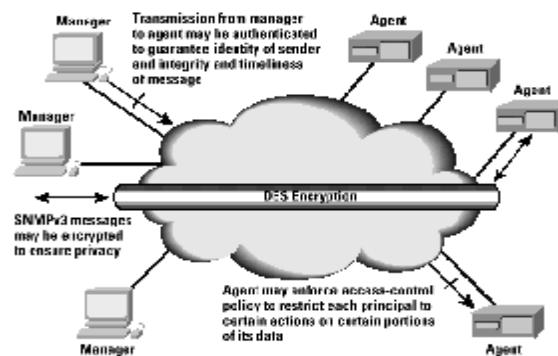
SNMPv3

- ⌘ Apelo do SNMP é a sua simplicidade
- ⌘ Conjunto de *Proposed Standards* em Janeiro de 1998

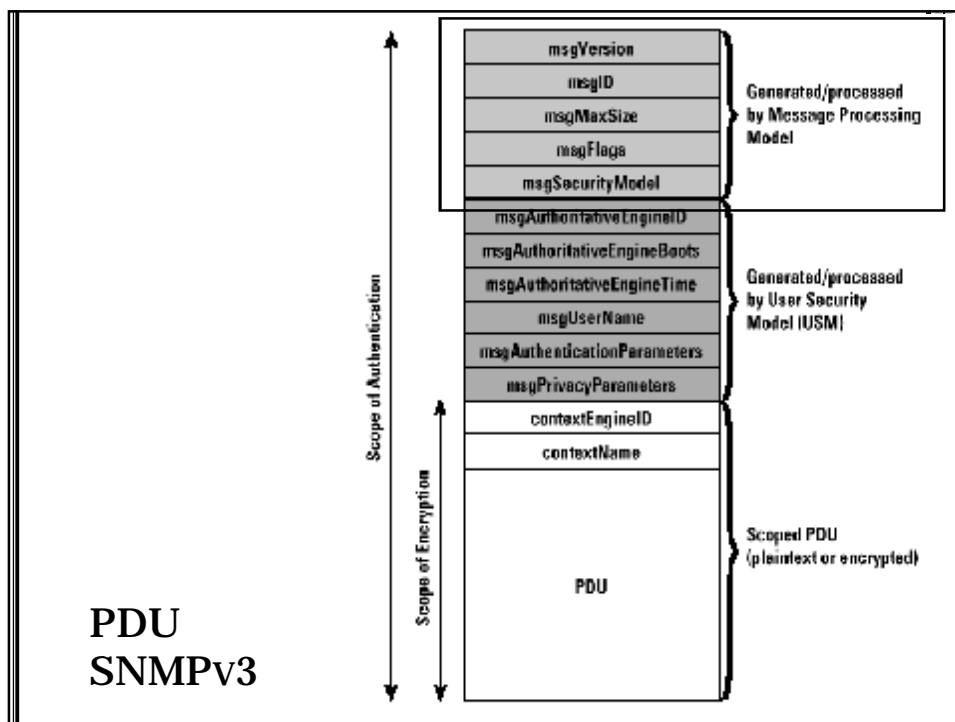
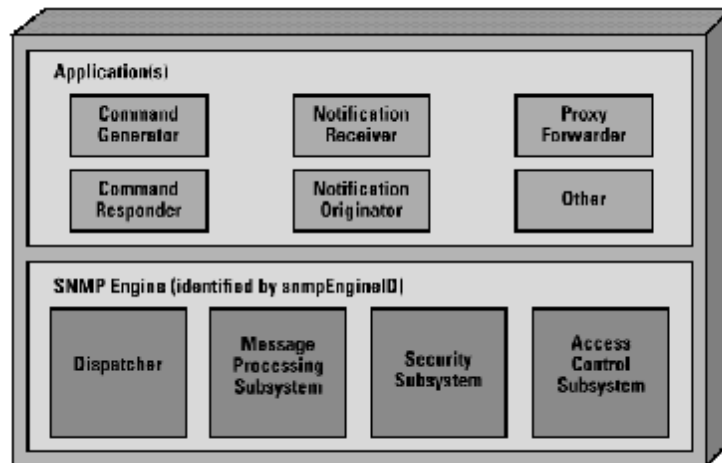
RFC	Title
2271	An Architecture for Describing SNMP Management Frameworks
2272	Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)
2273	SNMPv3 Applications
2274	User-Based Security Model for SNMPv3
2275	View-Based Access Control Model (VACM) for SNMP

SEGURANÇA NO SNMPv3

- ⌘ Principais características do SNMPv3
 - Autenticação Digital
 - Criptografia de Dados



ARQUITETURA SNMPV3



PDU
SNMPv3

USER-BASED SECURITY MODEL (USM)

☛ Definido na RFC 2274

- Autenticação: provê integridade de dados e autenticação da origem
 - ☛ MD5 ou SHA-1
- *Timeliness*: protege contra atrasos e/ou *replay*
- Privacidade: provê criptografia de dados
 - ☛ CBC (Chiper Block Chaining)
- Formato da Mensagem: define o formato dos parâmetros de segurança da PDU
- *Discovery*: obtenção de informações sobre outras *SNMP engines*
- *Key Management*: define os procedimentos para geração de chaves.

79

USM – ENTIDADE AUTORITATIVA

☛ Transmissores ou Receptores são definidos como entidades autoritativas, de acordo com as seguintes regras:

- Quando uma mensagem SNMP contém um *payload* que espera por uma resposta, o receptor desta mensagem é autoritativo
 - ☛ Get, Get-next, GetBulk, Set e Inform
- Quando uma mensagem SNMP contém um *payload* que não espera por uma resposta, a entidade origem da PDU é autoritativa
 - ☛ Trap, Respose e Report

80

USM - ...

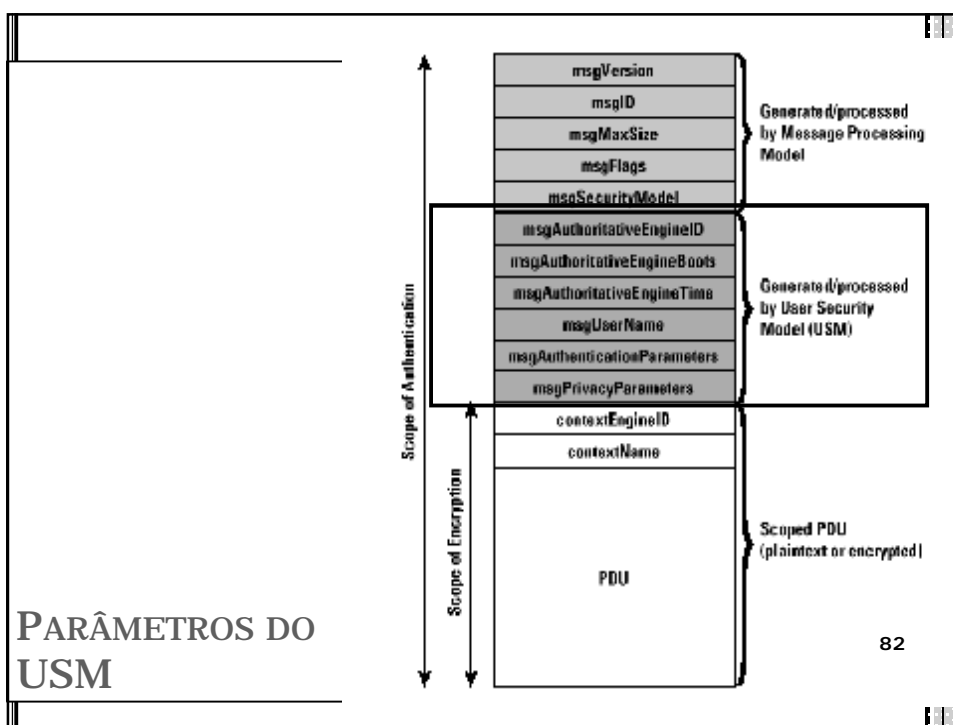
☛ A designação de autoridade serve a dois propósitos:

- O *timeliness* da mensagem é determinado com respeito ao *clock* mantido pela *engine* autoritativa. Assim a entidade não autoritativa pode sincronizar seu *clock* com a entidade autoritativa
- O processo de localização de chaves habilita o armazenamento das chaves em uma única *engine*

☛ Métodos de Criptografia

- Chave compartilhada
- 2 chaves
 - ☛ *Authkey*
 - ☛ *PrivKey*

81



CRIPTOGRAFIA DE DADOS

☞ Usa o CBC

- Chave *privKey* de 16 bytes
 - ☞ Utiliza-se os 8 primeiros bytes para o DES, pois ele necessita de 56 bits
- Vetor de Inicialização de 64 bits
 - ☞ Os 8 bytes restantes da *privkeys* são usados para o pre-IV
 - ☞ Para garantir que dois IV diferentes são utilizados dois “textos” diferentes, codificados com a mesma chave, é produzido um valor “salteado” de 8 bytes.
 - ☞ Executa-se um XOR entre o valor salteado e o pre-IV para gerar o novo IV

83

VIEW ACCESS CONTROL MODEL

☞ Este modelo tem duas características importantes

- Determina se o acesso a um objeto gerenciado de uma MIB Local é permitido
- Faz uso da MIB que define a política de controle de acesso para um agente

☞ Elementos

- *Groups*
- *Security level*
- *Context*
- *MIB views*
- *Access policy*

84

ELEMENTOS

☞ *Groups*

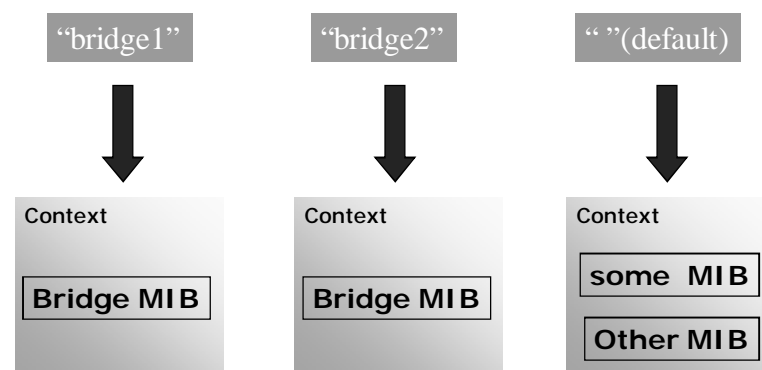
- Zero ou mais tuplas
 <*securityModel, securityName*>

☞ *Contexts*

- É o nome de um subconjunto de instâncias de objeto da MIB Local
- Conceito relacionado a controle de acesso
- Uma instância de objeto ou objeto pode estar associado a mais de um contexto
- Para identificar uma instância individualmente deve-se usar o *contextName* e o *contextEngineID*

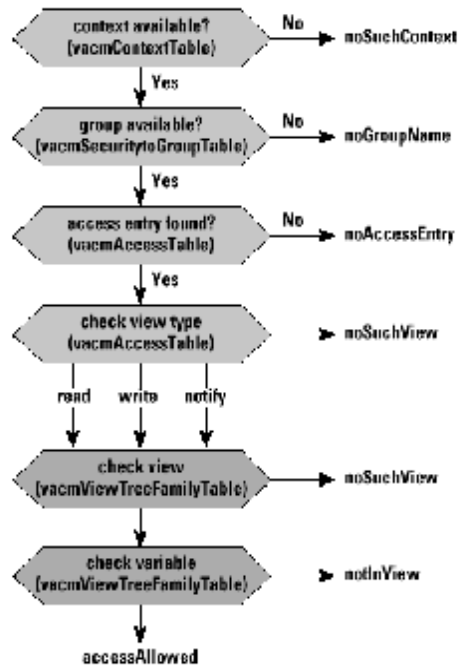
85

CONTEXTO - EXEMPLO

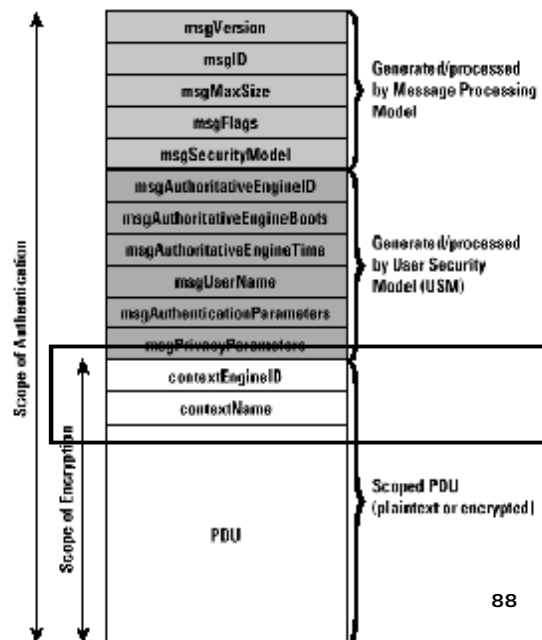


86

FLUXO



PARÂMETROS DC VACM



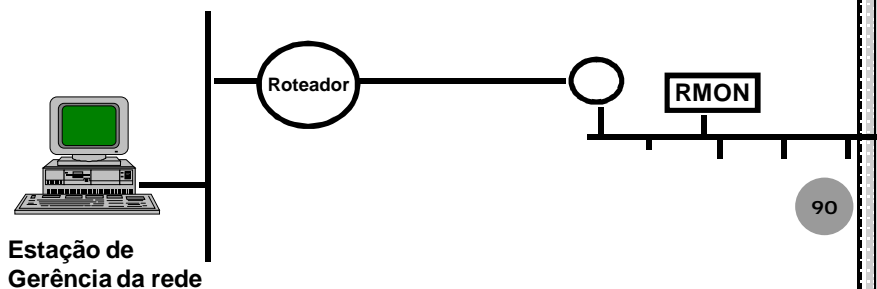
RMON - REMOTE MONITORING MIB

- ⌘ Agente procurador
- ⌘ MIB definida inicialmente para informações Ethernet e FDDI, que permite coleta e algum nível de tratamento local de dados, por um dispositivo diretamente conectado a uma rede local
- ⌘ Pode ser implementada num HUB, num analisador de rede ou mesmo numa estação da rede (mesmo num PC)
- ⌘ Pode operar *off-line*, coletando dados para posterior envio ao gerente

89

COMENTÁRIOS SOBRE RMON

- ⌘ Analisadores de rede
- ⌘ Ambiente TCP/IP
- ⌘ Usa SNMP para reportar dados
- ⌘ Extensões proprietárias



90

METAS DO RMON

☉ Operação *off-line*

- coleta dados e acumula estatísticas
- recuperação posterior
- notificação em caso de problemas

☉ Monitoração pró-ativa

- programas de diagnóstico
- log da performance da rede
- notificar em caso de exceção e prover dados para diagnóstico

91

METAS DO RMON

☉ Detecção e relato de problemas

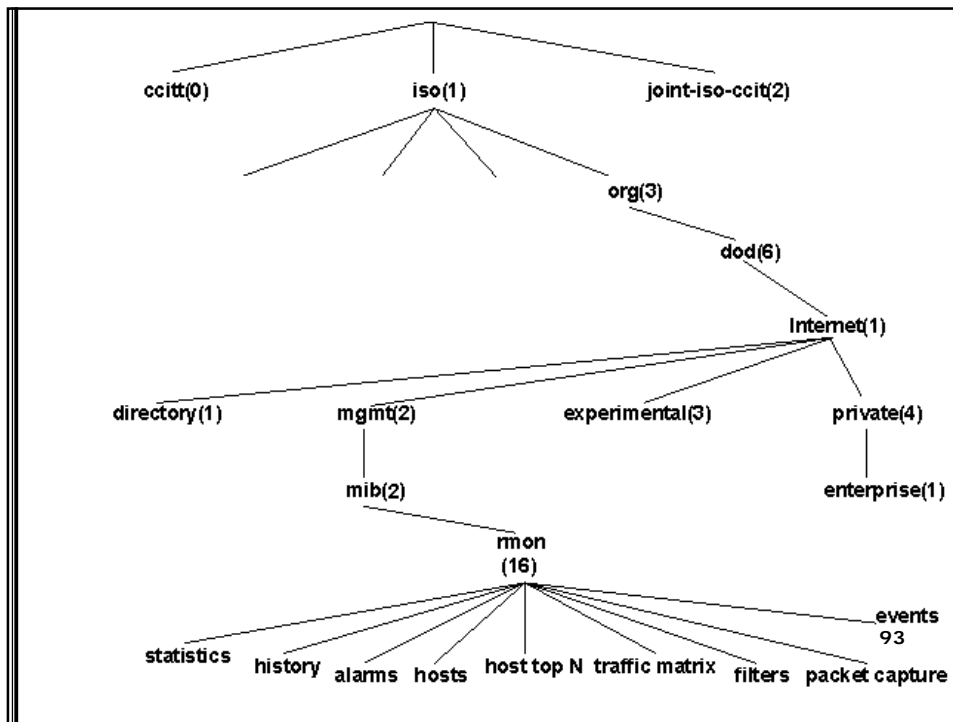
- reconhecer condições de erro (log ou notificação)

☉ Dados de valor adicionado

- determinar quem transmite mais ou gera mais erros

☉ Múltiplos gerentes

92



RMON - GRUPOS

- ☺ Estadísticas
- ☺ Histórico
- ☺ Alarme
- ☺ Host
- ☺ HostTopN
- ☺ Matriz
- ☺ Filtro
- ☺ Captura de Pacotes
- ☺ Eventos

RMON - GRUPO ESTATÍSTICAS

- ⌘ **Pacotes, octetos, broadcasts, multicasts, colisões**
- ⌘ **Pacotes descartados pelo agente**
- ⌘ **Erros**
 - undersize
 - fragments
 - CRC/Alignment
 - Collision
 - Oversizes
- ⌘ **Processamento local, reduz tráfego gerente-agente e carga de processamento no gerente**

95

RMON - GRUPO ESTATÍSTICAS

- ⌘ **SNMP é usado para recuperar a tabela inteira**
- ⌘ **Ethernet e Token-ring e futuramente outros interfaces, tal como FDDI**

96

ESTATÍSTICAS

☿ Carga, erros, tamanho de pacotes

☿ etherStatsTable

- index *
- dataSource
- dropEvents
- octets
- pkts
- broadcastPkts
- multicastPkts
- CRCAlignErrors
- undersizePkts
- oversizePkts
- fragments
- jabber

97

ESTATÍSTICAS

☿ etherStatsTable (continuação)

- collisions
- pkts64Octets
- pkts65to127Octets
- pkts128to255Octets
- pkts256to511Octets
- pkts512to1023Octets
- pkts1024to1518Octets
- owner
- status

98

HISTÓRICO

☞ Armazena amostras das estatísticas

☞ `historyControlTable`

- `index *`
- `dataSource`
- `bucketsRequest`
- `bucketsGranted`
- `interval`
- `owner`
- `status`

99

HISTÓRICO

☞ `etherHistoryTable`

- | | |
|------------------------------|--------------------------------|
| – <code>index *</code> | – <code>CRCAAlignErrors</code> |
| – <code>sampleIndex *</code> | – <code>undersizePkts</code> |
| – <code>intervalStart</code> | – <code>oversizePkts</code> |
| – <code>dropEvents</code> | – <code>fragments</code> |
| – <code>octets</code> | – <code>jabbers</code> |
| – <code>pkts</code> | – <code>collisions</code> |
| – <code>broadcastPkts</code> | – <code>utilization</code> |
| – <code>multicastPkts</code> | |

100

ALARME

☛ Geração de alarmes a partir de limites estabelecidos

☛ alarmTable

- index *
- interval
- variable
- sampleType
(absoluto, delta)
- value
- startupAlarm
- risingThreshold
- fallingThreshold
- risingEventIndex
- fallingEventIndex
- owner
- status

101

HOST

☛ Estatísticas concernentes aos hosts das sub-redes monitoradas

☛ hostControlTable

- index *
- dataSource
- tableSize
- lastDeleteTime
- owner
- status

102

HOST

☞ hostTable

- | | |
|-----------------|--------------------|
| – address * | – inOctets |
| – creationOrder | – outOctets |
| – index * | – outErrors |
| – inPkts | – outBroadcastPkts |
| – outPkts | outMulticastPkts |

☞ hostTimeTable

- creationOrder *
- index *
-

103

HOSTTOPN

☞ Armazena tabelas de hosts ordenados segundo os objetos: inPkts, outPkts, inOctets, outOctets, outErrors, outBroadcast e outMulticast

☞ hostTopNControlTable

- | | |
|-------------------|---------------|
| – controllIndex * | – requestSize |
| – hostIndex | – grantedSize |
| – rateBase | – startTime |
| – timeRemaining | – owner |
| – duration | – status |

104

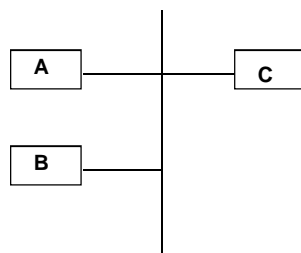
HOSTTOPN

- `hostTopNTable`
 - `hostTopNReport *`
 - `hostTopNIndex *`
 - `hostTopNAddress`
 - `hostTopNRate`

105

MATRIZ

- Registra dados sobre o tráfego entre pares de hosts



Gera dados sobre: A->B, B->A, B->C, C->B, A->C, C->A

106

MATRIZ

☉ matrixCotrolTable

- index *
- dataSource
- tableSize
- lastDeleteSize
- owner
- status

107

MATRIZ

☉ matrixSDTable

- sourceAddress *
- destAddress *
- index *
- SDpkts
- SDOctets
- SDErrors

108

MATRIZ

- ☉ matrixDSTable
 - destAddress *
 - sourceAddress *
 - index *
 -

109

FILTRO

- ☉ Monitoração de pacotes seleccionados através de testes de padrões
- ☉ Filtro de dados e de status

filtro de dados

```

dataoffset      = 0
pktData        = 00 00 00 00 00 A5 00 00 00 00 00 BB
pktDataMask    = FF FF FF FF FF FF FF FF FF FF FF FF
pktDataNotMask = 00 00 00 00 00 00 FF FF FF FF FF FF
  
```

filtro de status

```

pacote > 1.1518 (0), pacote < 64 (1), CRC ou alinhamento (2)
  
```

110

FILTRO

☉ filterTable

- index *
- channelIndex
- dataOffset
- pktData
- pktDataMask
- pktDataNotMask
- pktStatus
- pktStatusMask
- pktStatusNotMask
- owner
- status

111

FILTRO

☉ channelTable

- index *
- ifIndex
- acceptType
- dataControl
- turnOnEventIndex
- turnOffEventIndex
- eventIndex
- eventStatus
- description
- owner
- status

112

EXEMPLOS DE CONSULTA (FILTRO)

- ☞ filtra os pacotes que tenham como um destino o host A e não se originam no servidor
- ☞ filtra os pacotes IPX que possuem erros
- ☞ filtra os pacotes destinados ao servidor RARP

113

CAPTURA DE PACOTES

- ☞ Possibilita armazenar pacotes de um determinado canal
- ☞ `bufferControlTable`
 - `index *`
 - `channelIndex`
 - `fullStatus`
 - `fullAction`
 - `captureSliceSize`
 - `downloadSliceSize`
 - `downloadOffset`
 - `maxOctetsRequest`
 - `maxOctetsGranted`
 - `capturePkts`
 - `turnOnTime`
 - `owner`
 - `status`

114

CAPTURA DE PACOTES

☞ captureBufferTable

- controlIndex *
- index *
- packetId
- packetData
- packetLength
- packetTime
- packetStatus

115

EVENTOS

☞ Um evento é disparado por uma condição configurada em outro grupo. Tal evento pode gerar um TRAP e/ou um registro de LOG

☞ eventTable

- index *
- description
- type
- community
- lastTimeSent
- owner
- status

116

EVENTOS

☉ logTable

- eventIndex *
- index *
- time
- description

117

RMON 2

☉ Grupo de trabalho instituído pelo IETF em dezembro de 1994

☉ Novas e ampliadas funcionalidades

- Possibilidade de selecionar pacotes tanto por seu endereço Ethernet quanto pelo endereço TCP/IP
- Capacidade de filtro aumentada
- Habilidade para rastrear protocolos (com campos de comprimento variável)
- Possibilidade de efetuar decodificação nas 7 camadas

118

RMON 2

- ⌘ **Agente RMON2 implementa a MIB II, RMON e outros grupos que permitem a monitoração em todas as camadas**
- ⌘ **Tabelas de controle para controlar as operações do agente RMON**
- ⌘ **Tabelas com os resultados das monitorações**

119

RMON 2

- ⌘ **Camada de aplicação significa uma classe de protocolos não limitada aos níveis MAC e de rede, podendo incluir protocolos de transporte, sessão, apresentação e aplicação**
- ⌘ **Um diretório de protocolos contém um registro de protocolos que o agente é capaz de reconhecer e monitorar**
- ⌘ **Protocolos reconhecidos são os definidos em tempo de implementação**
- ⌘ **Extensibilidade limitada: decodificação do próximo nível baseada em tabelas**

120

RMON2 - GRUPOS

- ☿ Protocol directory
- ☿ Protocol distribution
- ☿ Address mapping
- ☿ Network layer host
- ☿ Network layer matrix
- ☿ Application layer host
- ☿ Application layer matrix
- ☿ User history
- ☿ Probe configuration

121

PROGRAMAÇÃO SNMP

- ☿ API CMU (Carnegie Mellow University)

```
struct snmp_session *snmp_open (session)
struct snmp_session *session;

struct snmp_session {
    char *community;
    int community_len;
    int retries;
    long timeout;
    char *peername;
    short remote_port;
    short local_port;
    int (*callback) ();
}
```

122

PROGRAMAÇÃO SNMP

```
void snmp_read (fd_set)
fd_set *fd_set;

int callback (operation, session, reqid, pdu)
int operation;
struct snmp_session *session;
int reqid;
struct snmp_pdu *pdu;
struct snmp_pdu {
    ipAddress address;
    int command;
    long reqid, errstat, errindex;
    OID *enterprise;
    int enterprise_len;
    ipAddress agent_addr;
    int trap_type, specific_trap;
    long timre;
    struct variable_list *variables;
}
```

123

PROGRAMAÇÃO SNMP

```
struct variable_list {
    struct variable_list *next_variable;
    OID *name;
    int name_len;
    char type;
    union {
        long *integer;
        char *string;
        OID *objId;
    } val;
    int val_len;
}

snmp_send (session, pdu)
snmp_create_pdu(command)
snmp_add_null (pdu, name, name_len)
snmp_fix_pdu (pdu, command)
snmp_synch_response (session, pdu, response)
```

124

PROGRAMAÇÃO SNMP

© SNMP-Capable GAWK (4BSD/ISODE SNMP)

```
BEGIN {
    pattern { actions }
    .
    .
    .
    END { }
```

125

PROGRAMAÇÃO SNMP

```
BEGIN {
    print sysDescr;

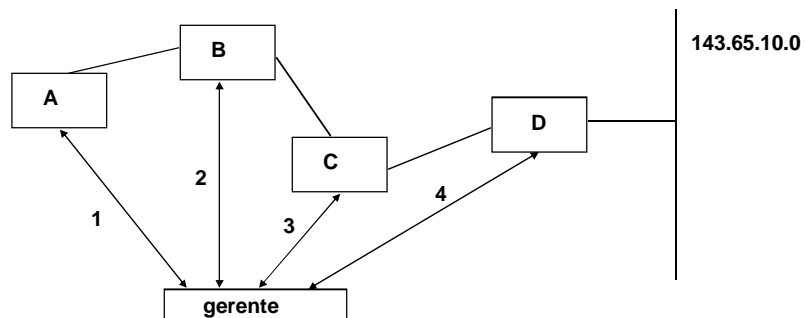
    for (i in ipRouteDest) {
        printf ipRoutingTable: to %s via %s\n",
            ipRouteDest, ipRouteNextHop;
    }

    for (i in ipRouteDest) {
        printf "Route: to %s via %s on %s\n",
            ipRouteDest, ipRouteNextHop,
            ifDescr[ipRouteIfIndex];
    }
}
```

126

- Variáveis especiais: AGENT, RETRIES, TIMEOUT, ERROR

PROGRAMAÇÃO SNMP



127

PROGRAMAÇÃO SNMP

```

BEGIN {
  printf "de %s para %s \n", AGENT, DEST;
  numhop = 2;
  while (numhop != 1) {
    hop = ipRouteNextHop [DEST];
    if (hop == 0) {
      printf "nao existe rota de %s para %s",
        AGENT, DEST;
      exit(1);
    }
    printf "maquina: %s", hop; numhop =
ipRouteMetric1[DEST];
    AGENT = hop;
  }
}
  
```

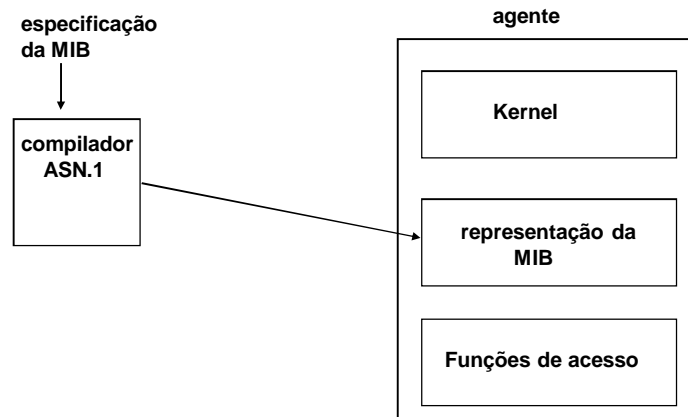
128

IMPLEMENTAÇÃO DE NOVOS AGENTES

- ☞ Mapeamento para a estrutura de informação (SMI)
- ☞ Método de acesso ao recurso
 - Demanda
 - *Polling*
 - Notificação
- ☞ Implementação

129

IMPLEMENTAÇÃO DE NOVOS AGENTES



130

FERRAMENTAS DE GERÊNCIA DE REDES

- ⌘ Inspeção dos objetos gerenciados
- ⌘ Setar filtros e limiares
- ⌘ Armazenamento / log
- ⌘ Apresentação
 - diagramas, cores
 - mensagens
- ⌘ Resposta automática
 - “trouble ticketing systems”
 - interpretação / explicação / sugestão



131

GERÊNCIA PRÓ-ATIVA

- ⌘ Antecipar problemas que provocarão determinado impacto na rede, principalmente em seu desempenho.
- ⌘ Capacidade de evitar a ocorrência desses problemas ou minimizar seu impacto
- ⌘ Elementos que contribuem para que a gerência pró-ativa de redes de computadores seja mais confiável
 - sistemas especialistas
 - monitores remotos
 - agentes procuradores
 - programas de simulação

132

DIFICULDADES

- ☞ Dificuldade de obtenção de informações relevantes
- ☞ Excesso de informações básicas (contadores e indicadores de *status*)
- ☞ Interpretar e correlacionar os dados?



AGREGANDO INTELIGÊNCIA À GERÊNCIA PRÓ-ATIVA

- ☞ Inferir a causa de um problema a partir de síndromes reconhecidas nos dados obtidos da rede
- ☞ Os dados podem ser obtidos por monitoração ou por captura remota (agentes SNMP ou agentes RMON)
- ☞ Dado o estado de um sistema, recomenda o que fazer a seguir com base no conhecimento acumulado a nível à esta situação



AGREGANDO INTELIGÊNCIA À GERÊNCIA PRÓ-ATIVA

- ☞ **Sistemas especialistas ou consultores inteligentes**
- ☞ **Capturar o conhecimento e a experiência de um ou mais especialistas**
- ☞ **Técnicas de representação do conhecimento**
 - Regras de produção
 - Se-Então

135

SISTEMAS ESPECIALISTAS APLICADOS À GERÊNCIA DE REDES

- ☞ **Orientados a diagnóstico**
- ☞ **Usam:**
 - Regras de produção para representar o conhecimento
 - Método de inferência de encadeamento para frente
 - Padrão de comparação
- ☞ **Monitoração**
- ☞ **Predição**
- ☞ **Controle**

136

CONSTRUÇÃO DOS MÓDULOS INTELIGENTES

- a) Que tipo de conhecimento é envolvido?
- b) Como pode o conhecimento ser representado?
- c) Quanto conhecimento é necessário?
- d) Qual é exatamente o conhecimento necessário?

137

EXEMPLO DE REGRA: NÍVEL DE BROADCAST

☛ Se a taxa de *broadcast* num intervalo de 1 hora é maior que 8% Então:

- verificar se o horário da ocorrência está dentro do horário útil (7:00-22:00 horas), do contrário, ignorar a ocorrência;
- identificar os *hosts* com os níveis mais altos de *broadcast* (*script broad_nivel.pl*);
- nos *hosts* identificados, analisar a configuração do software de comunicação, para saber quais são as razões pelas quais esses *hosts* estão transmitindo um número tão alto de pacotes *broadcasts*;

138

EXEMPLO DE REGRA: NÍVEL DE BROADCAST

- verificar máscara da rede. Uma máscara errada pode provocar tormenta de pacotes *broadcast*;
- verificar que na rede sendo monitorada não estejam estações com versões do UNIX incompatíveis

139

DESCOBERTA DE NOVOS PROBLEMAS

- ☞ Ler dados da rede sistematicamente e comparar com limiares determinados dinamicamente em função de parâmetros definidos pelo gerente:
 - janela de amostragem
 - tolerância
- ☞ Detecção de rajadas (problemas relacionados com causa já registrada anteriormente)
 - consulta à base de dados de problemas

140

DESCOBERTA DE NOVOS PROBLEMAS

- ☞ Determinar severidade do problema
componentes envolvidos
- ☞ Invocar rotina de diagnóstico
- ☞ Gerar registro de problema

141

APLICAÇÕES DE GERENCIAMENTO

- ☞ Tratamento inteligente dos dados
- ☞ Detecção de rajadas
- ☞ Gerência pró-ativa
- ☞ Determinação dinâmica de limiares
- ☞ Reconhecimento de padrões
- ☞ Análise de tendências
- ☞ Registro seletivo

Software de
terceiros

Desenvolvimento
próprio

142

ALGUMAS REFERÊNCIAS

- ☉ **BRISA - Gerenciamento de Redes - Uma abordagem de Sistemas Abertos - Makron Books - 1993**
- ☉ **Network Management - A practical perspective - Allan Leinwand & Karen Fang - Addison-Wesley 1993**
- ☉ **SNMP, SNMPv2 and CMIP - The practical Guide to Network-Management Standards - William Stallings - Addison-Wesley 1992**
- ☉ **Fault Management Tools for a Cooperative and Decentralized Network Operations Environment - Ewerton Madruga & Liane Tarouco - IEEE Journal on Selected Areas in Communications, August 1994**