

Protocolo SNMPv1

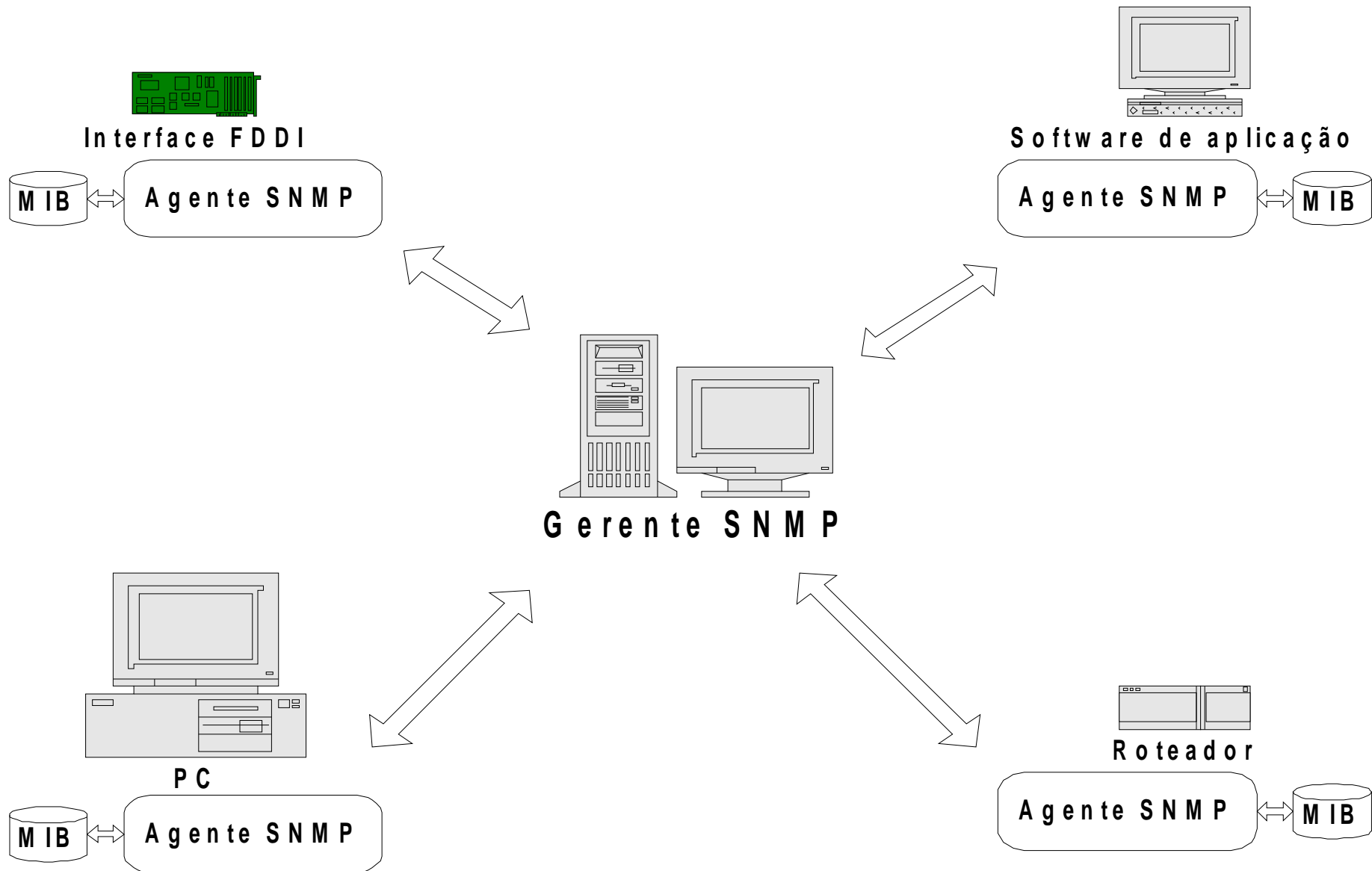
Prof. Mauro Tapajós



Introdução

- *Simple Network Management Protocol*
- SNMP é o protocolo de gerência mais usado por fabricantes e operadores de redes de comunicação
- Define como funciona a arquitetura de gerenciamento de redes TCP/IP
- Simples para ser implementado em todo tipo de equipamentos
- Flexível o bastante para aceitar futuras modificações

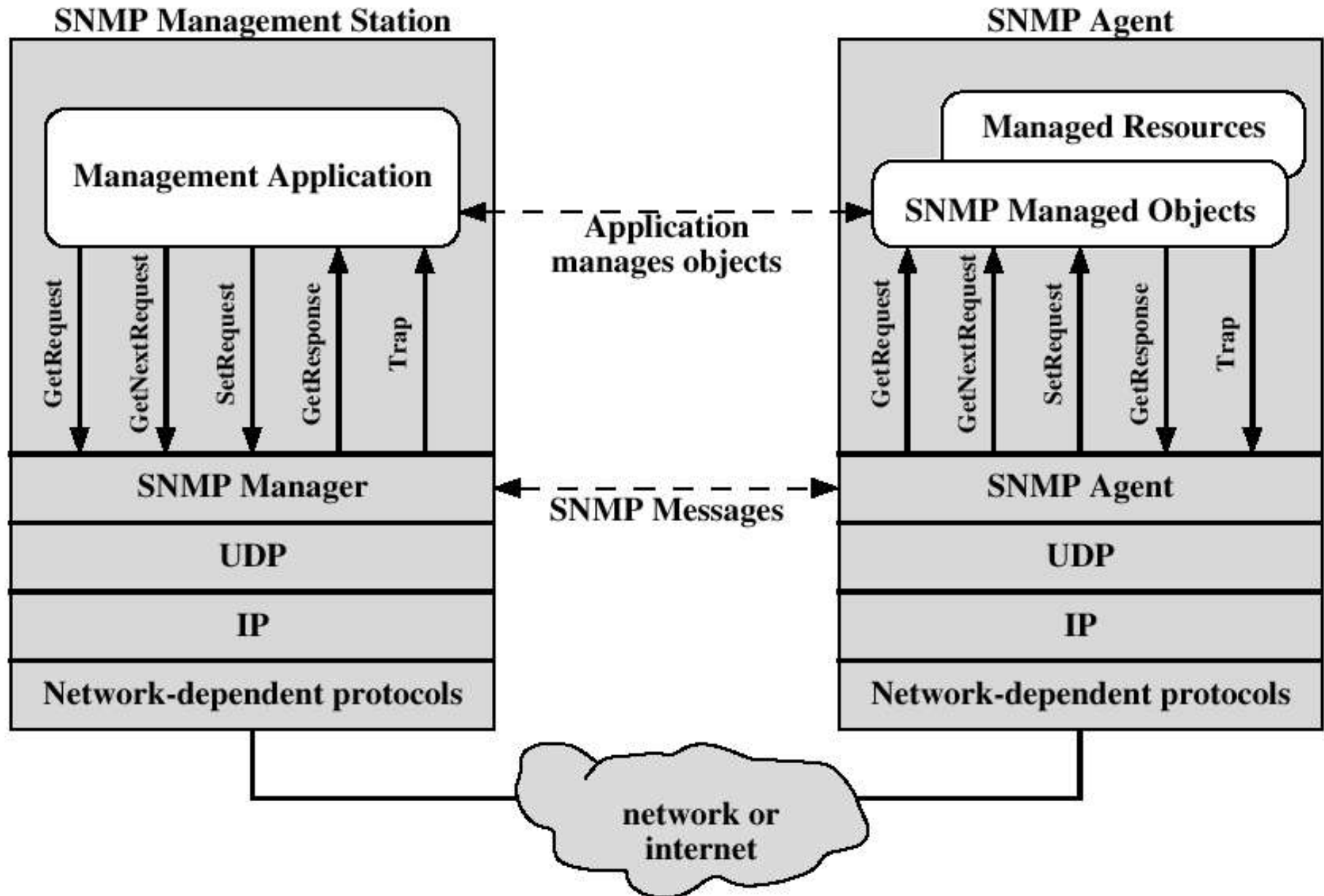
Arquitetura SNMP



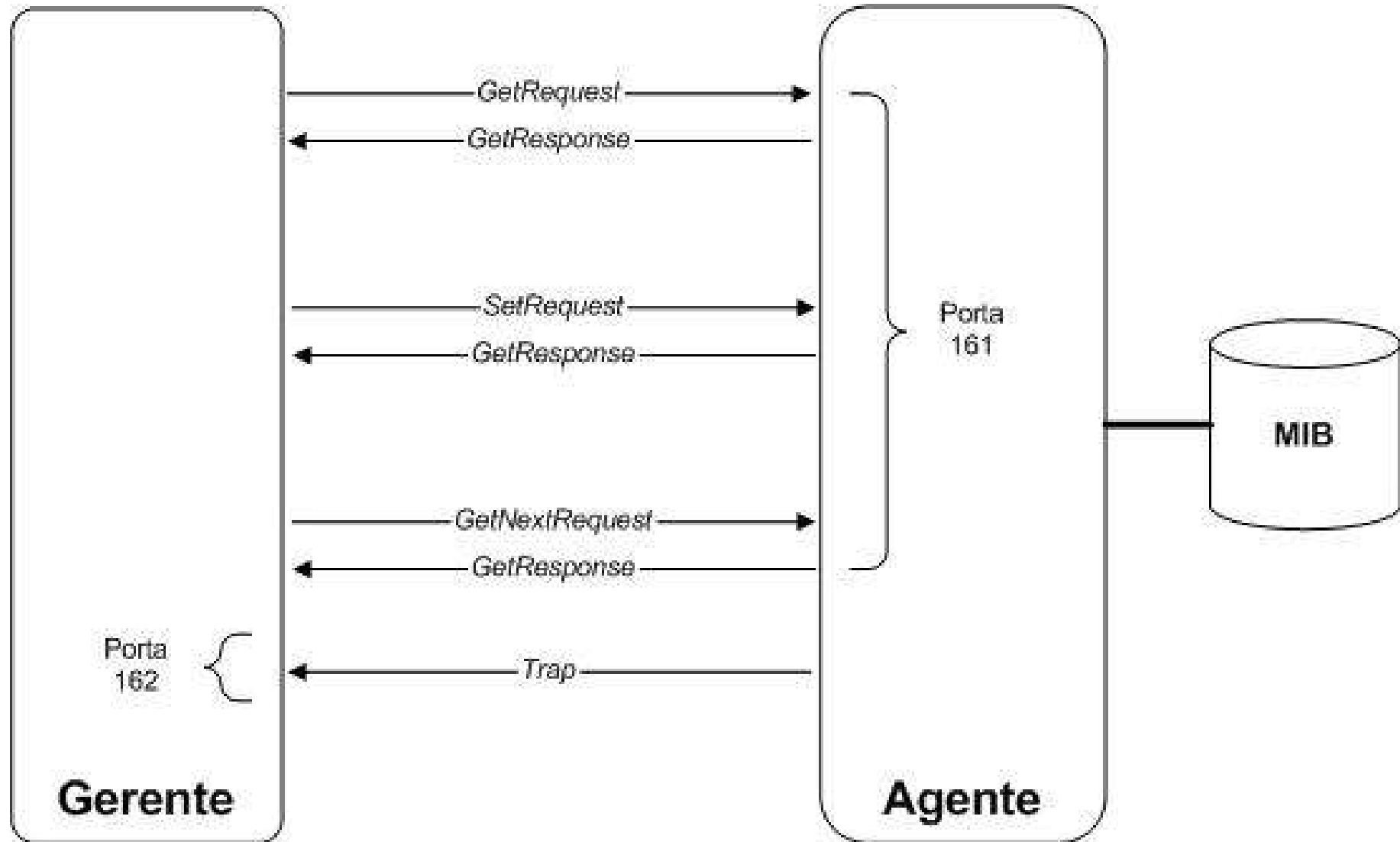
Características do Protocolo

- Não necessita de um serviço de transporte confiável (normalmente roda sobre UDP)
- Utiliza as portas 161 (agente) e 162 (gerente, para as *traps*)
- Baseado em arquitetura cliente-servidor (pedidos-respostas)

A comunicação SNMP

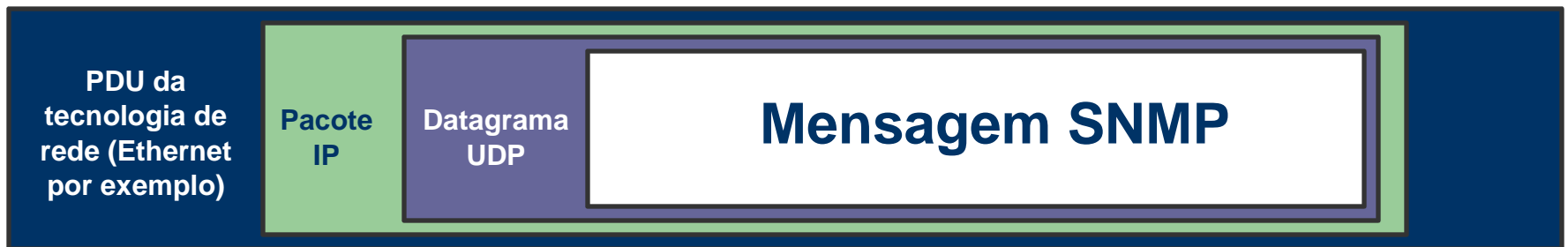


Operações SNMPv1

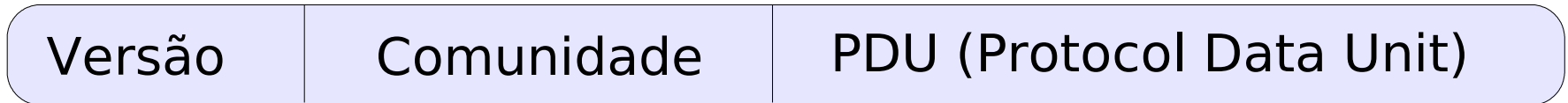


Encapsulamento do Protocolo SNMP

- Normalmente SNMP utiliza serviço UDP sobre IP
- Existe a possibilidade de se usar outras pilhas de transporte (SPX, etc)



Mensagem SNMPv1



Número inteiro indicando a versão do SNMP sendo usada (0 para SNMPv1)

Este campo carrega o nome de comunidade que o originador da mensagem está usando.

Dados efetivos de gerência a serem analisados e processados

Formato do PDU SNMPv1

PDU Type	Request ID	Error Status	Error Index	Object 1 Value 1	Object 2 Value 2	Object x Value x
----------	------------	--------------	-------------	------------------	------------------	------------------

Variable Bindings

- **Tipo de PDU:** especifica a operação SNMPv1
- **Request ID:** número sequencial para associar pedidos com respostas
- **Error Status:** código de retorno da operação
- **Error Index:** em caso de erro, aponta para a variável onde houve o problema
- **Lista de pares objeto-valor:** objetos e valores a serem utilizados na operação

O que pode dar errado numa operação SNMPv1?

- O campo *error status* indica o que ocorreu
- O campo *error index* aponta para a primeira variável que apresentou falha
- Valores de *error status*:
 - 0 - *noError* (não houve erro)
 - 1 - *tooBig* (a resposta é muito grande para a implementação)
 - 2 - *noSuchName* (uma das variáveis não consta na MIB do agente requisitado)
 - 3 - *badValue* (erro no pedido de *set*)
 - 4 - *readOnly* (uma mensagem *Set* tenta alterar uma variável *read-only*)
 - 5 - *genError* (erro desconhecido)

Coleta dos Dados de uma MIB - *GetRequest*

- As mensagens de *Get* coletam dados da MIB do agente sem alterá-los
- Para isso, envia uma mensagem especificando os objetos desejados por meio de uma lista de pares (ID das variáveis - Valor da variável)
- A mensagem de *Get* segue com os campos de valor das variáveis todo nulo
- Operação atômica: se ocorre um erro em algum dos objetos pedidos, toda a mensagem é perdida

Coleta dos Dados de uma MIB com *GetNextRequest*

- As mensagens de *Getnext* coletam dados da MIB do agente de maneira semelhante à mensagem de *Get*
- A diferença é que *Getnext* pede um objeto que é o objeto imediatamente seguinte numericamente ao que está na mensagem
- Isso permite que uma MIB seja acrescentada com novos objetos e o gerente possa pedir estes novos objetos, mesmo sem conhecê-los

Alterações nos Dados de uma MIB - *SetRequest*

- As mensagens de *Set* alteram os dados na MIB do agente
- Para isso, envia uma mensagem especificando os objetos desejados por meio de uma lista de pares (ID das variáveis - Valor desejado da variável)
- A mensagem de *Set* segue com os campos de valor das variáveis contendo os valores a serem inseridos
- Sua resposta em caso de sucesso conterá os mesmos pares ID-Valor

Comunidades SNMP

- Nomes textuais que especificam o nível de acesso às facilidades oferecidas pelos agentes
- Mecanismo trivial de autenticação (Falha de segurança no protocolo)
- Casa um agente com um conjunto de entidades de aplicação
- Inibe a utilização da operação de *set* (falta de segurança)

MIB View

- Um subconjunto de objetos de uma MIB de um dispositivo
- Determina um grupo de objetos relacionados por grau de acesso
- Controle de acesso SNMP define que uma determinada *community* está restrita às variáveis de uma determinada MIB view
- Um perfil de comunidade = MIB *view* + modo de acesso da *view* para aquela comunidade

Controle de acesso

- Cada agente controla sua MIB e o seu uso por estações gerentes, segundo o nome de comunidade apresentado na mensagem SNMPv1 (esquema de autenticação trivial)
- Aspectos de segurança necessários
 - Serviço de autenticação
 - Política de acesso
 - Serviço proxy

Controle de acesso

- Modos de acesso (mínimo) a um objeto em SNMPv1 (cláusula ACCESS na macro OBJECT-TYPE):
 - *Read-only*
 - *Read-write*
 - *Write-only*
 - *Not-accessible* (usado, por exemplo, para evitar que se peça uma tabela inteira)

Comunidades: Para que?

- Controlar quais dispositivos são controlados por qual gerente
- Prevenir que gerentes desautorizados escrevam na MIB de um determinado agente
- Restringir a informação vista por determinado gerente
- Definir um modelo administrativo para a arquitetura

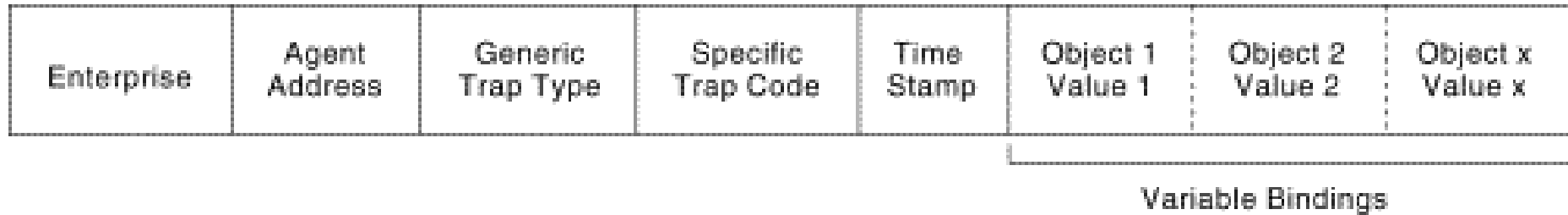
Mensagem *Trap*

- Reporta ao gerente eventos, problemas e condições anormais
- *Traps* específicas são definidas pelo valor do *enterprise OBJID* e o número *specific-trap*
- Manipulação e diagnóstico de várias *traps* que chegam (correlação): inteligência que realiza um primeiro diagnóstico da situação
- Política de monitoramento: se existem muitos dispositivos na rede fica inviável se fazer pollings. Uma solução é se fazer pollings menos frequentes e responder somente se uma *trap* foi recebida alertando algo

Tipos de *Traps*

- 0 – *coldStart***: reinicialização do dispositivo, configuração do agente não se altera
- 1 – *warmStart*** : reinicialização do dispositivo com possível alteração da configuração do agente
- 2 – *linkDown***: falha num dos links do agente
- 3 – *linkUp***: volta de um link do agente
- 4 – *authenticationFailure***: falha na autenticação de uma mensagem recebida
- 5 – *egpNeighborLoss***: perda da comunicação com um vizinho EGP
- 6 – *enterpriseSpecific***: traps específicas

Formato do PDU para a *trap* SNMPv1



- **Enterprise:** OBJID que indica o tipo de dispositivo que enviou a trap
- **Tipo da trap genérica:** um dos 6 tipos de *trap* genérica
- **Código específico da trap:** identifica uma *trap* específica de uma determinada “Enterprise”
- **Timestamp:** Tempo passado entre o momento em que foi gerada a trap e a última reinicialização
- **Lista de pares objeto-valor:** específica da *trap*

Declaração de uma *Trap específica* (MACRO ASN.1 TRAP-TYPE)

frDLCIStatusChange TRAP-TYPE

ENTERPRISE frame-relay

**VARIABLES {frCircuitIndex,
frCircuitDlci, frCircuitState}**

DESCRIPTION

**"This trap indicates that the indicated
Virtual Circuit has changed state. It
has either been created or
invalidated, or has toggled between
the active and inactive states."**

::=1

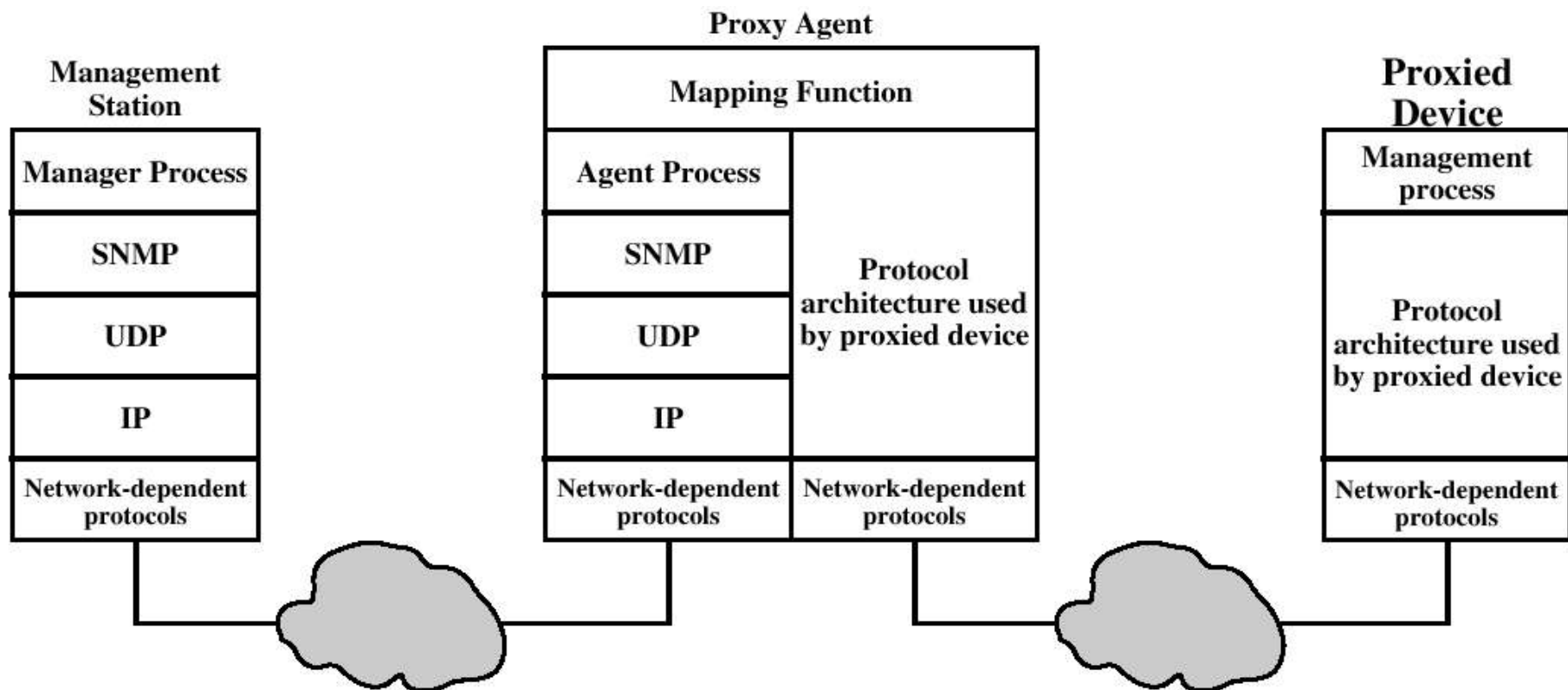
Tipos de Agentes SNMP

- Agentes SNMP basicamente podem ser construídos de duas formas:
- Agentes extensíveis
 - São desenvolvidos com arquitetura aberta com design modular permitindo adaptações para novos requisitos de dados de gerenciamento e extensões operacionais
- Agentes Monolíticos
 - Não são extensíveis. São construídos com otimizações para determinadas plataformas de hardware ou SO, visando melhor performance

Agentes Proxy

- É um agente que não está no mesmo sistema sendo gerenciado
- Acesso indireto a dispositivos:
 - sem suporte SNMP
 - sem suporte TCP/IP ou da rede sendo usada
- Segurança para um determinado ambiente de rede

Agentes Proxy



SNMPv1 - Limitações

- Não é adequado para gerenciar redes muito grandes
- Não é adequado para coletar grandes volumes de dados
- *Traps* não são confirmadas (não se sabe se chegaram ou não!)
- Autenticação trivial. Mais adequado a monitoramento (*gets*) do que controle (*sets*), falta de segurança
- Não permite comandos diretos no agente. Apenas alterações chaveadas por mudanças nos valores de objetos
- Não permite comunicação entre gerentes