

Protocolo SNMPv2

Prof. Mauro Tapajós



Problemas a Serem Resolvidos

- Pouca funcionalidade comparativamente ao CMIP/CMIS (protocolo/serviço de gerenciamento da arquitetura OSI)
- Segurança rudimentar (baseada em comunidades)
- Impossibilidade de comunicação entre gerentes
- Limitações no protocolo e suas mensagens.
- A característica atômica das mensagens, ou seja, o fato de uma mensagem obter total sucesso ou total fracasso nas operações (não há a possibilidade de sucesso parcial num *request*)
- Impossibilidade de configuração remota dos agentes

Histórico

- Seria natural a evolução do protocolo para corrigir falhas, limitações e, principalmente, falta de segurança
- No começo de 1992, o IETF anunciou a chamada por propostas para uma nova versão
- A primeira resposta veio nas RFC 1351 a 1353 e foi chamada SNMP seguro (incompatível com a versão 1)
- A versão 2 foi oficialmente apresentada em 1993 e hoje é chamada de SNMPv2 “clássica” ou SNMPv2p (baseada em *parties*)

Premissas SNMPv2p

- Manter as características de segurança oferecidas pelo SNMP seguro com a opção de outro algoritmo de criptografia alternativo ao DES
- Permitir o gerenciamento hierárquico por meio de comunicação entre gerentes
- Aceitar vários serviços de transporte
- Apresentar um mecanismo de coleta de informações mais eficiente
- Introduzir um novo modelo administrativo

Community-based SNMPv2

- A abrangente proposta SNMPv2 “Clássica” (SNMPv2p) baseada em *parties* -> fracasso
- Em 1995 foi feita mais uma revisão do protocolo em resposta à baixa aceitação da versão SNMPv2p
- Tal revisão se deu principalmente no que diz respeito ao contexto baseado em *parties*, à configuração dos agentes, à dificuldade de descoberta automática da rede e à implementação do modelo administrativo e de segurança
- O único consenso então obtido foi aceitar as novidades no protocolo, porém permanecendo o contexto de operação sob a antiga forma de comunidades

SNMPv2c

- Assim, foi apresentada mais uma versão de SNMP, agora chamada de SNMPv2c (baseado em comunidades)
- Nesta, o modelo administrativo apresentado na versão “clássica” e baseado em *parties*, foi completamente descartado
- SNMPv2c: assimilou somente as novas mensagens, correções e SMIv2, esperando ainda:
 - ⇒ Segurança
 - ⇒ Configuração Remota
 - ⇒ Infra-estrutura Administrativa
- SNMPv2c é a atual versão oficial do protocolo!!!

SNMPv2c

- Com a não aceitação do modelo de segurança proposto em SNMPv2p, a mensagem SNMPv2c possui os mesmos delimitadores e cabeçalhos da versão SNMPv1 (com exceção do campo versão que agora tem valor 1 indicando a versão 2)
- Foi mantido o mesmo esquema de comunidades da versão 1
- Apesar do fracasso, pode-se considerar que há aspectos positivos apresentados na versão SNMPv2p:
 - a criação de extensões da linguagem (que facilitam a declaração de novos objetos)
 - a melhoria da performance do protocolo na troca de informações com um melhor tratamento de erros
 - A útil experiência prática foi obtida nas implementações e testes com SNMPv2p

Alterações na SMI – SMIv2

- A SMIv2 é um superconjunto da primeira SMI (versão 1) - RFCs 1442, 1443, 1444
- SMIv2 é um evolução totalmente compatível com a SMIv1 (a exceção é o novo tipo de dado Counter64)
- Novos (e mais adequados) tipos de dados: *Counter32*, *Counter64*, BIT STRING, ..
- Uma nova macro para convenções de texto (TEXTUAL CONVENTIONS) que descreve melhor e com detalhes um tipo de dado específico definido pelo usuário

Alguns Tipos de Dados – SMIv2

Data Type	Description
INTEGER	Integers in the range of -2^{31} to $2^{31} - 1$,
UInteger32	Integers in the range of 0 to $2^{32} - 1$,
Counter32	A nonnegative integer that may be incremented modulo 2^{32} .
Counter64	A nonnegative integer that may be incremented modulo 2^{64} .
Gauge32	A nonnegative integer that may increase or decrease, but shall not exceed a maximum value. The maximum value can not be greater than $2^{32} - 1$.
TimeTicks	A nonnegative integer that represents the time, modulo 2^{32} , in hundredths of a second.
OCTET STRING	Octet strings for arbitrary binary or textual data; may be limited to 255 octets.
IpAddress	A 32-bit internet address.
Opaque	An arbitrary bit field.
BIT STRING	An enumeration of named bits.
OBJECT IDENTIFIER	Administratively assigned name to object or other standardized element. Value is a sequence of up to 128 nonnegative integers.

Nova macro OBJECT-TYPE

Exemplo:

`snmpAlarmInterval OBJECT-TYPE`

`SYNTAX Integer32`

`UNITS "seconds"`

`MAX-ACCESS read-create`

`STATUS current`

`DESCRIPTION`

`"The interval in seconds over which the data is sampled and compared..."`

`::= { snmpAlarmEntry 3 }`

- Cláusula MAX-ACCESS: *not-accessible*, *accessible-for-notify*, *read-only*, *read-write* e *read-create*
- Cláusula STATUS: *current*, *obsolete* e *deprecated*

Melhorias na manipulação de tabelas

- O tipo *RowStatus* – substitui a antiga coluna *EntryStatus* com os seguintes possíveis valores:
 - 1.**active** – linha operacional
 - 2.**notInService** – linha desabilitada
 - 3.**notReady** – linha ainda não completa
 - 4.**createAndGo** – criar a linha e disponibilizá-la
 - 5.**createAndWait** – criar a linha mas esperar por outros valores
 - 6.**destroy** – deletar todos os objetos da linha
- Às vezes não é possível se preencher uma linha da tabela num único set, por isso se deve mandar “esperar” pelo resto dos dados (*createAndWait*)

Alterações na Pilha de Transporte

- A pilha de transporte necessária para rodar SNMP sempre foi UDP
- Esta característica ficou mais flexível em SNMPv2, onde outras pilhas de protocolos podem ser usadas
- Entre as possibilidades de configuração para SNMPv2 estão:
 - Protocolos de transporte OSI
 - Appletalk (protocolo usado por máquinas *macintosh*)
 - SPX usado pelas redes *Novell*

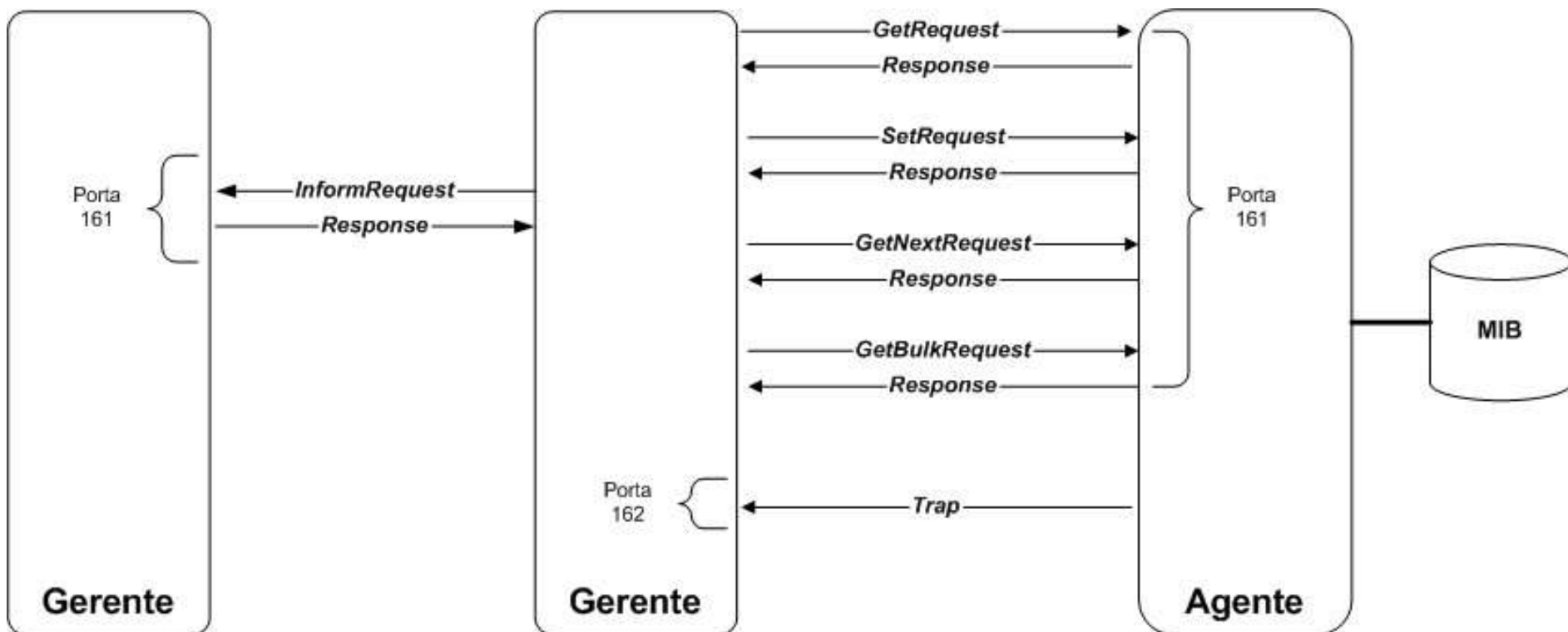
Alterações nas Operações

- ***getRequest / getNextRequest*** : não existe mais a perda da resposta toda em caso de problema numa das variáveis. Se num pedido de *get/getNext* ocorrer algum dos seguintes erros, o valor da variável será preenchido com o código correspondente e o status de erro da mensagem (*errorstatus*) será sucesso (zero)
 - *noSuchObject*
 - *noSuchInstance*
 - *endOfMibView*
- ***response***: é o novo nome da operação *getResponse*

Alterações nas Operações

- ***setRequest***: a operação de *set* é executada em duas fases: na primeira, as variáveis são testadas e na segunda elas são propriamente alteradas. Esta operação permanece atômica (tudo ou nada)
- A lista de códigos de erros aumentou definindo outras situações

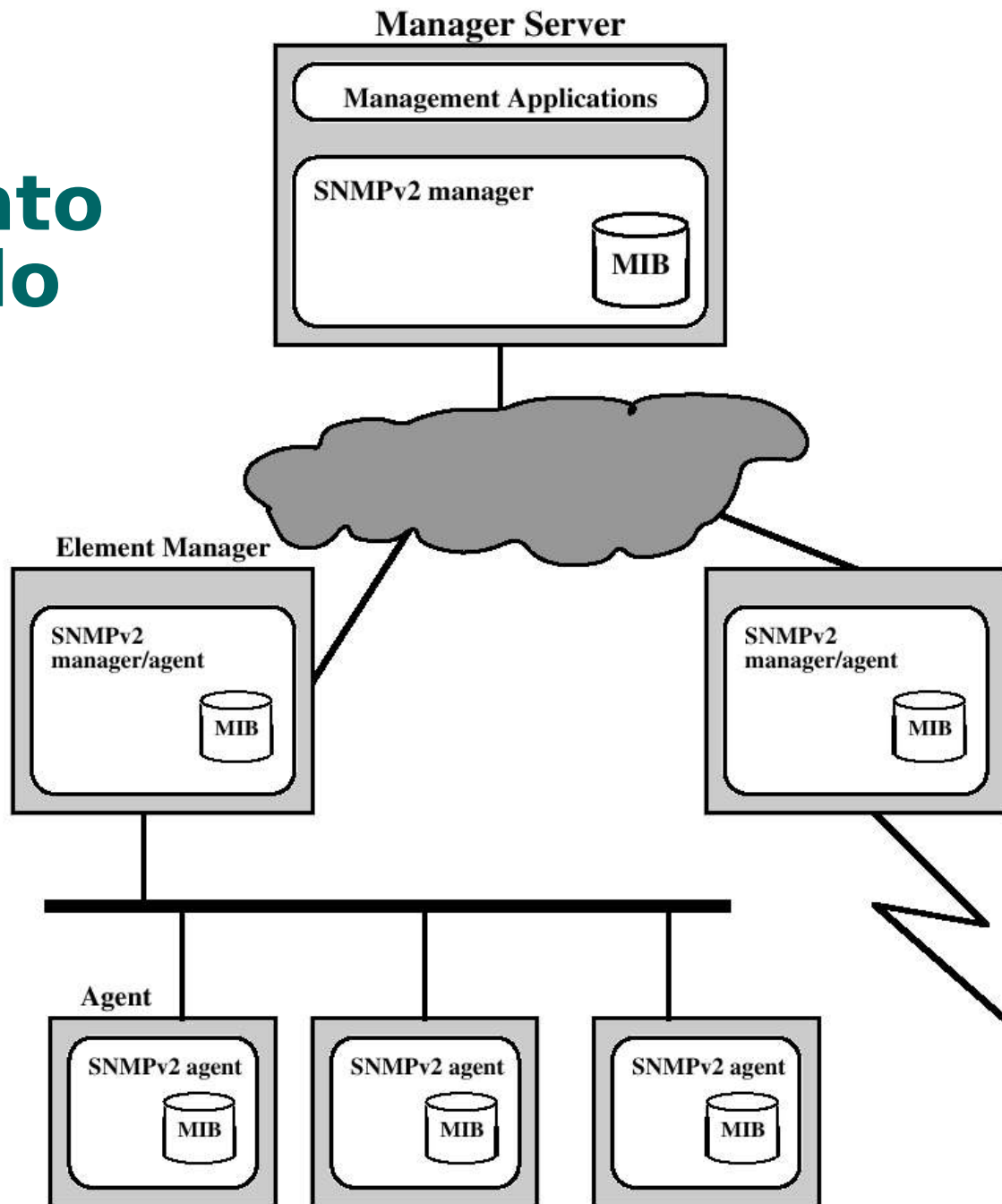
Operações SNMPv2



Nova Operação: *InformRequest*

- A nova mensagem *InformRequest* permite a um gerente enviar uma notificação à outro gerente quando for necessário
- Operação análoga à da mensagem *trap*, porém com confirmação explícita (uma mensagem *Response*)
- Um gerente envia um *InformRequest* a outro em resposta a eventos complexos como a ultrapassagem de um limite máximo de erros ou muitas tentativas de *get*
- Esta nova possibilidade permite a hierarquização da estrutura gerencial do SNMPv2

Gerenciamento Hierarquizado



Nova Operação: *getBulkRequest*

- Em SNMPv1 quando se desejava grandes quantidades de dados de uma MIB, a limitação do tamanho das respostas era função da implementação no agente distante
- Se era pedido um volume muito grande de informações para um único *get*, a resposta era vazia com o campo *error status* com valor *tooBig*
- Se era pedido um volume muito pequeno de informações para um único *get*, a coleta de dados não é eficiente

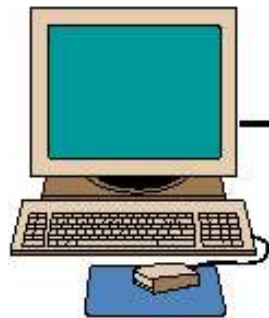
Maior Eficiência com *getBulkRequest*

- Esta nova operação otimiza a recuperação de um volume considerável de variáveis, pedindo eficientemente o máximo de dados que um agente pode enviar por meio de uma mensagem *response*
- O pedido de um *getBulkRequest* pode ser tanto de variáveis individuais ou de linhas de uma tabela
- A diferença entre as operações *getBulkRequest* e *getNextRequest* está na capacidade da primeira enviar linhas inteiras de uma tabela, além de “navegar” na MIB de forma sequencial exatamente como *get-next*

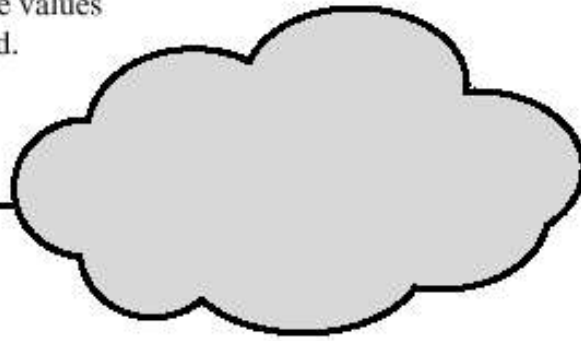
Nova Operação: *getBulkRequest*

**GetBulkRequest(non-repeaters = 2,
max-repetitions = 6, X, Y, TA, TB, TC)**

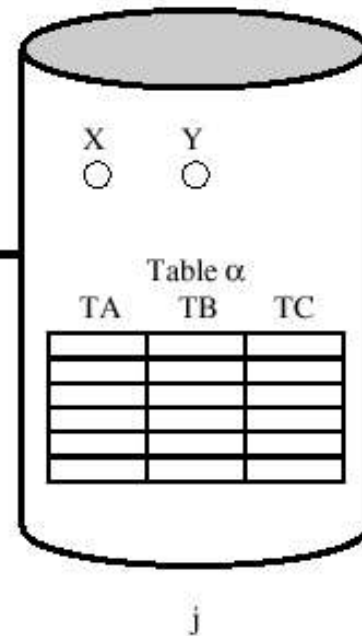
Manager issues request with six variable names; for the first two variables (non-repeaters = 2), a single value is requested; for the remaining variables, six successive values (max-repetitions = 6) are requested.



Management Workstation



Agent returns single value for X, Y, and six rows of table α .



Response [X, Y, TA(1), TB(1), TC(1),
TA(2), TB(2), TC(2),
TA(3), TB(3), TC(3),
TA(4), TB(4), TC(4),
TA(5), TB(5), TC(5),
TA(6), TB(6), TC(6)]

Formato do PDU *getBulkRequest*

PDU Type	Request ID	Non-repeaters	Max-repetitions	Object 1 Value 1	Object 2 Value 2	Object x Value x
				Variable Bindings		

- *Non-repeaters* – especifica o número de objetos escalares máximo a ser enviado no *response*
- *Max-repetitions* – define o máximo número de vezes que deve se seguir pelas variáveis de várias estâncias
- Note que o formato do PDU é o mesmo para as outras operações, simplesmente são usados os campos *error status* e *error index*

snmpV2-traps

- A definição de uma *trap* pode ser feita pela nova macro NOTIFICATION-TYPE e apresenta basicamente seu nome e sua lista de variáveis
- Todos os campos específicos da *trap* versão 1 estão agora dentro da lista de variáveis (no lugar do antigo campo *timestamp* agora é colocado o objeto *sysUptime*)
- Houve uma uniformização do formato da PDU, de forma a evitar o processamento extra para lidar com dois diferentes tipos de PDU como na versão 1

Mensagens SNMPv2

PDU type	request-id	0	0	variable-bindings
----------	------------	---	---	-------------------

(a) GetRequest-PDU, GetNextRequest-PDU, SetRequest-PDU, SNMPv2-Trap-PDU, InformRequest-PDU

PDU type	request-id	error-status	error-index	variable-bindings
----------	------------	--------------	-------------	-------------------

(b) Response-PDU

PDU type	request-id	non-repeaters	max-repetitions	variable-bindings
----------	------------	---------------	-----------------	-------------------

(c) GetBulkRequest-PDU

name1	value1	name2	value2	• • •	namen	valuen
-------	--------	-------	--------	-------	-------	--------

(d) variable-bindings

Situação Atual do Protocolo SNMP

- SNMP v1 e v2c são largamente usadas por fabricantes e operadores
- Utiliza esquema de segurança trivial baseada em *community-strings*
- Inibição do uso das operação de controle (*sets*)
- A aceitação de SNMPv3 ainda é uma incógnita