Índice

Arquiteturas

Arquitetura TMN

- Aplicação exclusiva para redes de telecomunicações.
- Tem como base o modelo funcional da gestão OSI.
- Rede de transmissão de dados dedicada exclusivamente à gestão.
- Sistema centralizado, mas com mais hierarquização do que a arquitetura OSI.

Arquitetura OSI

- Cinco áreas funcionais (FCAPS): configuração, gestão de falhas/anomalias, controlo de performance, gestão da segurança e contabilização.
- A gestão é também uma aplicação, necessitando da pilha completa de protocolos OSI (muito pesada).
- Bases de dados dos objetos a gerir (informação de gestão) conceptualiza os recursos nos vários níveis da pilha.
- Sistema centralizado (problemas de escabilidade).
- Protocolo/Serviço de interface: CMIP/CMIS.

INMF

- Objetos/Protocolos simples.
- Consome poucos recursos nos equipamentos de rede a gerir.
- Arquitetura simples e centralizada.
- Bases de dados de objetos de gestão (MIB) inspiradas no formalismo de abstração da OSI.
- Gestão está no nível das aplicações (i.e., é um serviço aplicacional).
- Preocupações crescentes ao nível da segurança.

INMF: Componentes de Normalização

- Normalização atual em vários componentes:

  > Structure of Management Information (SMI)

  > Management Information Bases(MIBs)

  > Simple Network Management Protocol (SNMP)

  > User-based Security Model (USM)

  > View Access Control Model (VACM)

- Modelo de comunicação assimétrico e assíncrono.
- Sistema com Poolling intensivo.
- Tipo de objetos simples com identificação através de OIDs, como na OSI.

INMF: Objetos de Gestão (SMI & MIBs)

- Os tipos possíveis para os objetos são definidos em ASN.1 na SMI (vai na 2ª versão).
- Os tipos de objetos permitidos são relativamente simples e a sua manipulação/organização é limitada e complexa.
- Os objetos são conceptualizações de recursos ou parâmetros de funcionamento dos equipamentos.
- Identificação universal e hierárquica com OIDs e o agrupamento dos objetos é feito em MIB Groups.
- Políticas de acesso definidas por MIB Views.

Simple Network Manag.Protocol(SNMP)

- Protocolo de transporte da informação/dados de gestão, simples e assíncrono.
- Utiliza preferencialmente o protocolo UDP e está pensado para utilização de mecanismos de polling intensivo.
- Apenas quatro comandos/primitivas para os gestores:
  - snmp-get, snmp-getnext, snmp-getbulk e snmp-set.
- Três comandos/primitivas para os agentes:
  - snmp-response, snmp-trap e snmp-inform.
- Pouca evolução dos PDUs ao longo do tempo.
- Apenas duas versões:
  > SNMPv1 (RFC 1157) – para o INMFv1;
  > SNMPv2 (RFC 1905) – para o INMFv2 e INMFv3.
- Primitivas/Operações SNMP:

  > get-req* (SNMPv1 & v2),

  > get-next-req* (SNMPv1 & v2),

  > get-bulk-req* (SNMPv2),

  > set-req* (SNMPv1 & v2),

  > get-response (SNMPv1 & v2),

  > trap (SNMPv1) & notification (SNMPv2),

  > inform (SNMPv2).

SNMP: Segurança & Controlo de Acesso

- Área de maior evolução atual, sobretudo a partir do SNMPv2.
- Sistema complexo definido em duas normas:
  - User-based Security Model (USM), (mecanismos para encriptação e sumarização)
  - View Access Control Model (VACM). (definição do controlo de acesso aos objetos)
- Utilização de mecanismos de segurança relativamente recentes.
- Indefinição no conceito de chave ou sistema de gestão de chaves.

- Utilização real ainda é baseada em community strings
- V1 / v2c:

  >“Community” String enviada em todas as mensagens protocolares

  >Atua na verdade como sendo uma “password”

  >O agente só responde se a “password” está correta

  PRÓS e CONTRAS:

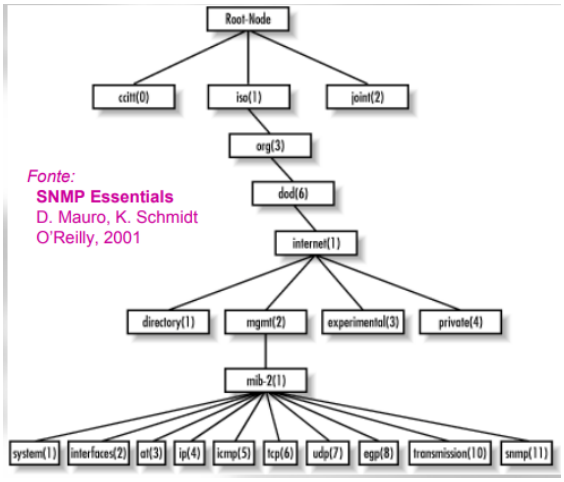
  >Simples, muito simples, de implementar….

  >Não é cifrado! à Segurança muito fraca à exige ”canal” seguro

  >Não há o conceito de “utilizador” mas sim de comunidade de gestores (dilui-se responsabilidades)
- V3:
  - Não usa “community strings”…
  - Autenticação dos utilizadores (agente e monitor verificam identidade)
  - “Segredo partilhado” para cada utilizador…
  - Mensagens são enviadas com hash (sumário) da mensagem criado com o segredo partilhado… o hash pode ser validado no destino
  - Conteúdo da mensagem (payload) pode opcionalmente ser cifrado com um segundo ”segredo partilhado”
  - Propriedades: autenticação, integridade, confidencialidade

SNMP: Management Information Bases

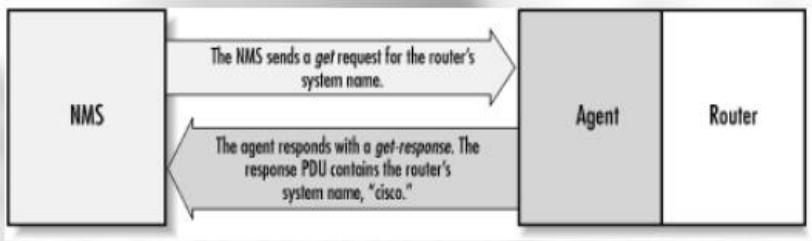
MIB-2



SNMP: Primitivas/Operações SNMP

> get-req* (SNMPv1 & v2),

```
$ snmpget -v 2c router-lab public .1.3.6.1.2.1.1.5.0
system.sysName.0 = "cisco"
```



> get-next-req* (SNMPv1 & v2),

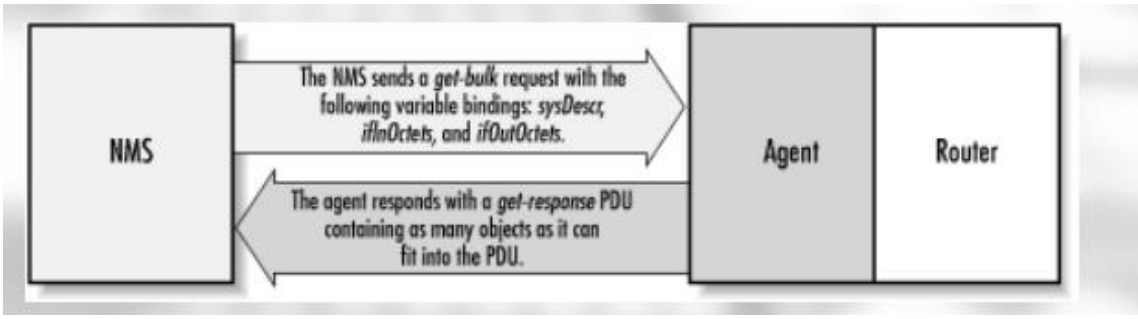
```
$ snmpwalk -v 2c router-lab public system
system.sysDescr.0 = "Cisco Internetwork Operating [...]"
system.sysObjectID.0 = OID: enterprises.9.1.19
system.sysUpTime.0 = Timeticks:(27210723)3 days, 3:35:07.23
system.sysContact.0 = ""
system.sysName.0 = "cisco"
system.sysLocation.0 = "labcom-di-uminho-pt"
system.sysServices.0 = 6
```

Nota: O comando **snmpwalk** do Net-SNMP é implementado à custa de várias operações **get-next-request**...

> get-bulk-req* (SNMPv2),

```
   $ snmpbulkget -v 2c -B 1 3 router-lab public
sysUpTime ifInOctets ifOutOctets
   system.sysUpTime.0 = Timeticks: (27210723) 3 days, 3:35:07.23
   interfaces.ifTable.ifEntry.ifInOctets.1 = 70840
   interfaces.ifTable.ifEntry.ifOutOctets.1 = 70840
   interfaces.ifTable.ifEntry.ifInOctets.2 = 143548020
   interfaces.ifTable.ifEntry.ifOutOctets.2 = 111725152
   interfaces.ifTable.ifEntry.ifInOctets.3 = 0
   interfaces.ifTable.ifEntry.ifOutOctets.3 = 0
```

Nota: A opção **–B** serve para indicar os valores de **<non-repeaters>** e **<max-repetitions>** da operação **get-bulk-request**...

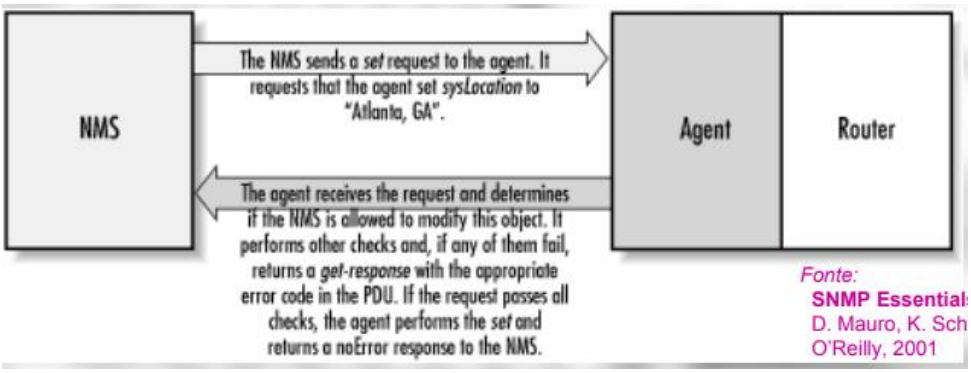


> set-req* (SNMPv1 & v2),

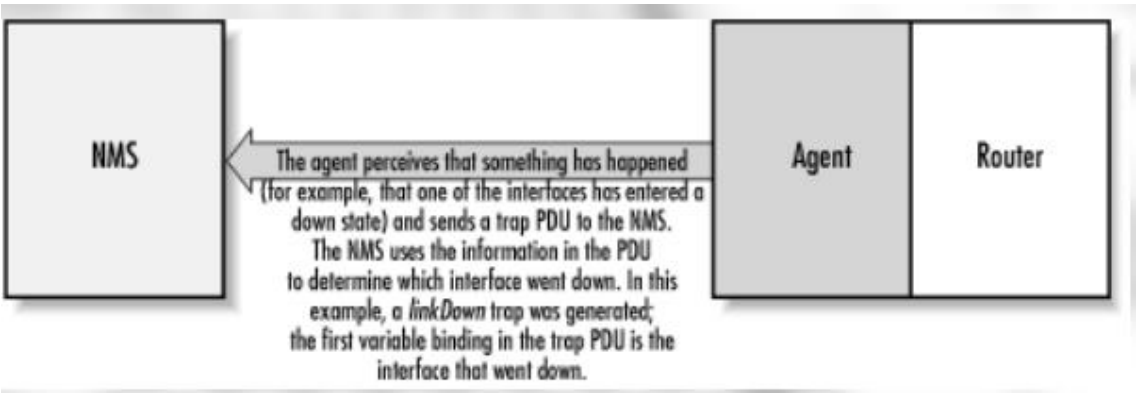
- **set-request()***

```
$ snmpget -v 2c router-ext public system.sysLocation.0
system.sysLocation.0 = "labcom-di-uminho-pt"

$ snmpset -v 2c router-ext labcom system.sysLocation.0
s "Buraco Negro"
system.sysLocation.0 = "Buraco Negro"

$ snmpget -v 2c router-ext public system.sysLocation.0
system.sysLocation.0 = "Buraco Negro"
```

Nota: A opção **s** serve para indicar que o novo valor é do tipo **Octet String** (no caso uma **DisplayString**)...



> trap (SNMPv1) & notification (SNMPv2),



Operações não solicitadas pelo gestor, informando de situações que não devem esperar para serem detetadas por polling

Exemplos:

- Um dos interfaces da máquina onde está a correr o agente mudou de estado operacional.
- Uma chamada para um modem não teve sucesso.
- Um integrado de memória ficou defeituoso.
- A temperatura de funcionamento atingiu níveis anormais.

FCAPS: Fault Management

FCAPS is the ISO Telecommunications Management Network model and framework for network management

- Diagnostic Testing
- Fault Detection/Isolation/Network Monitoring
- Fault Correction/Network Recovery
- Alarm Generation/Filtration/Handling/Correlation
- Logging & Statistics

FCAPS: Configuration Management

- Resource Management (Initialization & Provisioning)
- Network & Services Discovering
- Configuration Policies Management & Automation
- User/Clients Management (Registration & Support)
- Logging & Statistics

FCAPS: Accouting Management

- Resource Management (Costs Definition & Resource Usage)
- Users/Clients Quotas Monitoring, Reporting & Billing
- Auditing
- Logging & Statistics

FCAPS: Performance Management

- Resource Utilization & Peformance Monitoring (For network devices, systems and services)
- Users/Clients Utilization & Satisfaction
- Data Analysis & Capacity Planning
- Logging & Statistics

FCAPS: Security Management

- Threat Management (Definition & Monitoring)
- Users/Clients Access Management & Certification (Definition, Monitoring & Reporting)
- Security Garantees (Privacy, Authentication, etc)
- Auditing
- Logging, Data Analysis & Statistics

USM (User-based Security Model)

Ameaças principais

- Modificação da informação (em trânsito pela rede, modificando valores)
- Disfarce (masquerade) – perigo de alguém assumir a identidade do outro, fazendo operações não autorizadas

Ameaças secundárias

- Espionagem (disclosure) - observação indevida de conteúdos
- Modificação da "stream" de mensagens tirando partido do facto do protocolo ser connection-less (reordenar, atrasar, repetir mensagens, descartar mensagens)

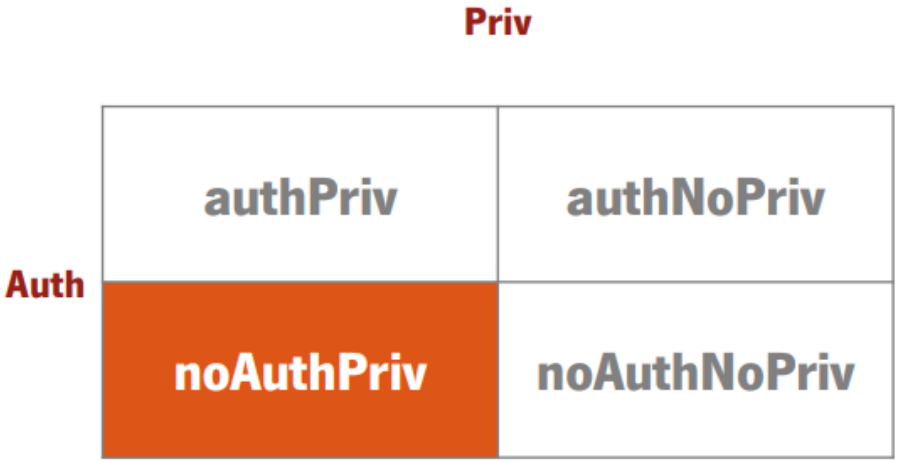
Ameaças não consideradas

- Negação de serviço
- Análise de tráfego – padrão de tráfego é previsível…

Garantias de Segurança (Requisitos)

- Integridade: garantia que as mensagens não foram alteradas na rede
- Autenticação: verificação da identidade do agente e do gestor
- Confidencialidade: não permitir espionagem
- Deteção de mensagens for a do tempo certo!
- Se o stress da rede for inconsistente com a segurança privilegia-se a primeira! (princípio base do USM)
- Não deve haver dependência de outros serviços, ex: NTP
- O mecanismo de segurança não pode alterar a filosofia de gestão SNMP…

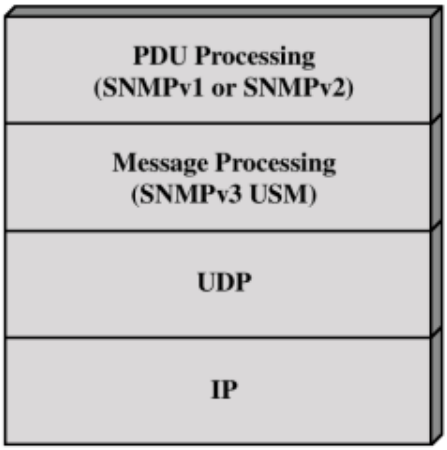
Confidencialidade vs Autenticação

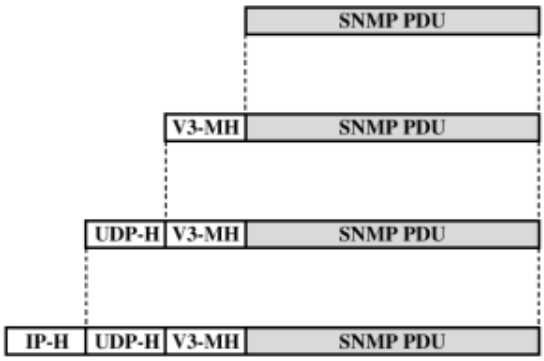


noAuthPriv não faz sentido

SNMP Messages (v3)

SNMPv3 define um formato de message que integra segurança e que possa ser usado em conjugação com v1/v2





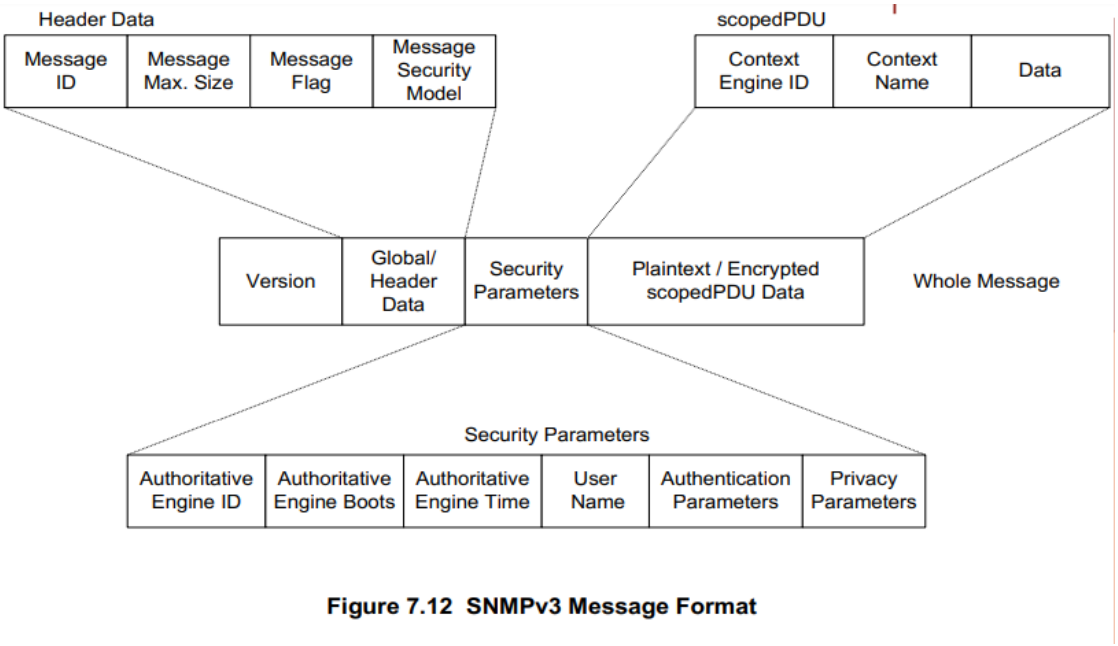
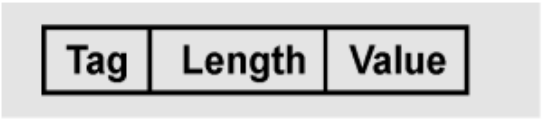


Figure 7.12 SNMPv3 Message Format

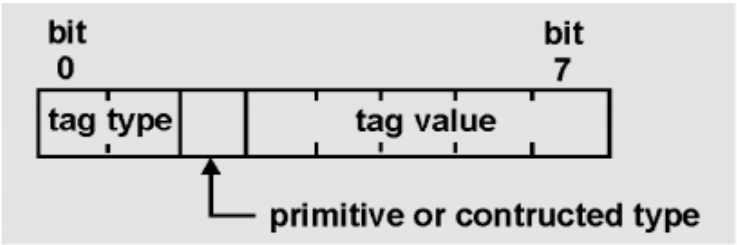
BER – Basic Encoding Rules

Os tipos definidos em ASN.1 precisam ser enviados entre sistemas heterogéneos =Necessidade de um mecanismo de encoding universal

Uso de tripletos: Etiqueta, Tamanho, valor



Etiquetas de 8bits:



Os "compiladores" ASN.1 pegam nas definições e geram:

- Definições de estruturas de dados (na linguagem de programação)
- Funções para codificar e descodificar as estruturas de dados

WIKI:SNMP

O Protocolo Simples de Gerenciamento de Rede (SNMP) é um protocolo padrão da Internet para coletar e organizar informações sobre dispositivos gerenciados em redes IP e para modificar essas informações para alterar o comportamento do dispositivo. Dispositivos que normalmente suportam SNMP incluem modems a cabo, roteadores, switches, servidores, estações de trabalho, impressoras e muito mais. **O SNMP é amplamente usado no gerenciamento de rede para monitoramento de rede. O SNMP expõe dados de gerenciamento na forma de variáveis nos sistemas gerenciados organizados em uma base de informações de gerenciamento (MIB) que descreve o status e a configuração do sistema. Essas variáveis podem então ser consultadas remotamente (e, em algumas circunstâncias, manipuladas) pelo gerenciamento de aplicativos.**

Um agente é um módulo de software de gerenciamento de rede que reside em um dispositivo gerenciado. Um agente tem conhecimento local de informações de gerenciamento e traduz essas informações para ou de um formulário específico do SNMP.

Uma estação de gerenciamento de rede executa aplicativos que monitoram e controlam dispositivos gerenciados. O SNMP opera na camada de aplicativos do conjunto de protocolos da Internet. Todas as mensagens SNMP são transportadas através do User Datagram Protocol (UDP). O agente SNMP recebe solicitações na porta UDP 161. O gerente pode enviar solicitações de qualquer porta de origem disponível para a porta 161 do agente. A resposta do agente é enviada de volta para a porta de origem no gerenciador. O gerente recebe notificações (Traps e InformRequests) na porta 162. O agente pode gerar notificações de qualquer porta disponível.

Problemas de implementação

A maioria das implementações SNMP, independentemente da versão do protocolo, usa o mesmo código de programa para unidades de dados de protocolo de decodificação (PDU). Assim, muitos fornecedores tiveram que emitir patches para suas implementações SNMP. Entre outros problemas foram encontrados com a decodificação de mensagens de interceção SNMP recebidas pela estação de gerenciamento SNMP ou solicitações recebidas pelo agente SNMP no dispositivo de rede.

Os poderosos recursos de gravação do SNMP, que permitem a configuração de dispositivos de rede, não estão sendo totalmente utilizados por muitos fornecedores, em parte devido à falta de segurança nas versões SNMP antes do SNMPv3 e em parte porque muitos dispositivos simplesmente não podem ser configurados MIB objeto muda. Os requisitos da operação do Conjunto SNMP não são fáceis de implementar corretamente, e muitos fornecedores optaram por omitir o suporte para o Conjunto - provavelmente para reduzir seu custo de desenvolvimento e reduzir o tamanho do código, entre outras razões.

A estrutura de árvore aparentemente simples do SNMP e a indexação linear podem nem sempre ser compreendidas suficientemente bem dentro das estruturas de dados internas que são elementos do design básico de uma plataforma. Consequentemente, o processamento de consultas SNMP em determinados conjuntos de dados pode resultar em maior utilização da CPU do que o necessário. Um exemplo disso seria tabelas de roteamento grandes, como BGP ou IGP.

Alguns valores de SNMP (especialmente valores tabulares) exigem conhecimento específico de esquemas de indexação de tabelas e esses valores de índice não são necessariamente consistentes entre as plataformas. Isso pode causar problemas de correlação ao buscar informações de vários dispositivos que podem não empregar o mesmo esquema de indexação de tabela (por exemplo, buscando métricas de utilização de disco, em que um identificador de disco específico é diferente nas plataformas).

Resolução de exercícios

Teste com criação de MIB

A arquitetura de gestão SNMP tem evoluído no sentido de minorar algumas das suas limitações iniciais. Alguns dos avanços centram-se na criação de mecanismos especiais que permitem atividades de monitorização e configuração com um nível funcional mais elevado/abstrato do que é possível utilizando apenas MIBs tradicionais como a MIB-II. É o caso dos conceitos de *Alarme* e de *Cálculo Remoto de Expressões*. Dê um exemplo de como a utilização destes dois conceitos avançados se podem complementar na implementação dum serviço de monitorização tecnologicamente mais avançado que os serviços tradicionais de monitorização SNMP.

Quando se complementa o Alarme e o Cálculo Remoto de Expressões para serviços de monitorização e configuração nas MIBs é permitido ultrapassar limites anteriores impostos pelo SNMP. Estes serviços comparam estatísticas de valores atuais com antigos limites impostos, quando estes são ultrapassados é criado um evento. Assim sendo é permitido a interligação de vários componentes distintos de um ou mais sistemas informáticos (SI), permitindo uma fácil administração dos mesmos. Monitorização de diferentes plataformas via SNMP (Simple Network Management Protocol) ou através de scripts desenvolvidos especificamente para cada situação, a monitorização de bases de dados, tais como SQL Server, MySQL e Oracle e monitorização de Websites etc.

Considere a criação duma MIB que inclua um conjunto de objectos que permita a monitorização do seguinte conjunto de variáveis, passíveis de serem obtidas num agente SNMP implementado numa On-Board-Unit (OBU) dum veículo automóvel: velocidade própria, distância para o veículo da frente e velocidade do veículo da frente. Tenha em consideração que os valores destas variáveis (ou amostras) serão armazenadas na instrumentação do agente a um intervalo mínimo desejável (por exemplo, 10 Hz) que pode ser parametrizado individualmente para cada uma delas. Também deve ser parametrizado o número máximo total de valores recolhidos na tabela de amostras da instrumentação/MIB. Tendo em conta que as variáveis em causa podem ser obtidas por tecnologias diversas disponíveis no automóvel (sensores laser, sensores ultra-sons, sensores de alta-frequência, sistema GPS, comunicações sem fios entre os automóveis, etc.), a cada amostra deve estar associado o tipo de tecnologia com que foi obtido. Além disso, como, na realidade, as amostras serão introduzidas na tabela a intervalos não regulares (devido à tal diversidade das implementações dos vários tipos de tecnologias que permitem a recolha de tais amostras), a cada amostra também deve ser associado o tempo de cálculo. Portanto, além do seu valor, a cada amostra devem estar associados o tipo de variável, o tipo de tecnologia usada para a obter e uma tag temporal que identifica univocamente o tempo em que o seu valor foi calculado. A OBU dum automóvel é uma espécie de caixa preta que as aplicações veiculares implementadas nos dispositivos do utilizador podem monitorizar através de SNMP sobre uma rede IP interna do próprio automóvel. Esta rede local interliga os dispositivos do utilizador e a OBU.

a) Apresente a especificação duma MIB para ser implementada pela instrumentação do agente SNMP dentro da OBU e que permita a monitorização nos moldes descritos anteriormente. Utilize, se possível, a sintaxe definida pela SMIv2.

```
-- The OBU Group
 OBUNumber OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Numero de automoveis em automovelTable."
::= { OBU 1 }

-- The OBUTable table
OBUTable OBJECT-TYPE
 SYNTAX SEQUENCE OF OBUEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "This entity's table of OBU."
 ::= { OBU 2 }

  OBUEntry OBJECT-TYPE
  SYNTAX OBUEntry
  ACCESS not-accessible

  STATUS mandatory
  DESCRIPTION "An entry (row) for information about data scanned by one. The key for the table
  is the OBU's index number." INDEX { OBUIndex }
  ::= { OBUTable 1 }

  OBUEntry::= SEQUENCE {
  OBUVpropria INTEGER,
  OBUDfrente INTEGER,
  OBUVfrente INTEGER,
  OBUIndex INTEGER,
   OBUTipo DisplayString,
   OBUTimestamp timeticks }
```

//Valores

OBUIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION "The index of the automovel in this group of OBU."

::= { OBUEntry 1 }

OBUSensor OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-write

STATUS mandatory

DESCRIPTION "Um texto que contém a identificação do tipo: (sensores laser, sensores ultra-sons, sensores de alta-frequência, sistema GPS, comunicações sem fios entre os automóveis, etc.)"

::= { OBUEntry 2 }

ACCESS read-only

STATUS mandatory

DESCRIPTION "Data do momento"

::= { OBUEntry 3 }

OBUVpropria OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION "Velocidade do automóvel"

::= { OBUEntry 4 }

OBUDfrente OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION "Distancia para o veiculo da frente."

::= { OBUEntry 5 }

```
OBUVfrente OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION "Velocidade do automovel da frente."
::= { OBUEntry 6 }
```

b) Escreva um algoritmo duma simples aplicação veicular que gere um alarme ao condutor quando o automóvel se aproxima a menos duma distância de segurança DS ao veículo da frente e a sua velocidade e/ou aceleração sejam preocupantemente superiores às do veículo da frente (preocupante no sentido de se prever uma colisão se o condutor não alterar os parâmetros de condução). DS deve ser um parâmetro configurável pelo condutor ou pelo fabricante do automóvel (não se incomode com o mecanismo de interface que permite a sua configuração) e que já é passado para o algoritmo. Esta aplicação deve ter o papel dum gestor SNMP e deve utilizar primitivas SNMPv2c para obter os dados necessários à sua implementação dum agente SNMP numa OBU com a MIB definida na alínea anterior.

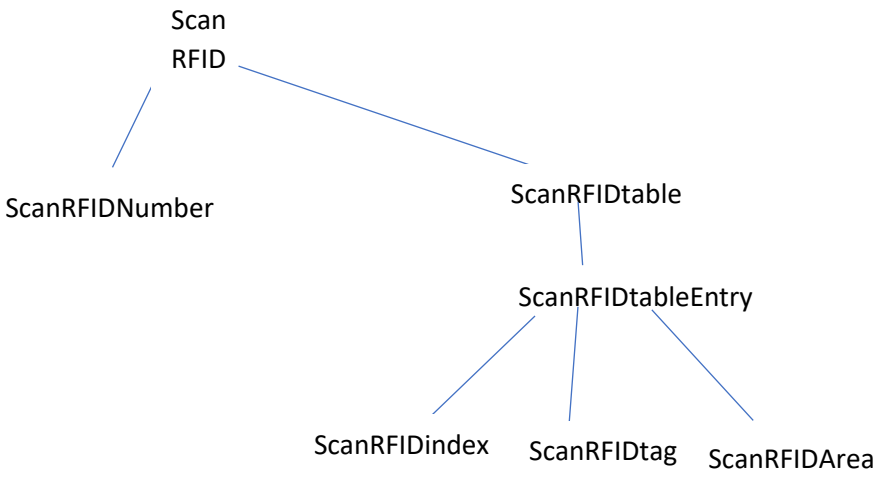
```
index = snmpget(OBU.1)  //Numero total de Carros
N = snmpget(OBU.IP,161,"public",OBUNumber.0) //Começa pelo inicial
oid = OBUDfrente //distancia do carro da frente
While(N <= índex){ //Percorrer todos os carros

  Inst = snmpgetnext(OBU.IP,161,"public",oid)
  If(inst.Dfrente < Ds){ Alarme } //Se a distancia atual for menor que a de segurança
  vel = 0; velFrente = 0; //Cria variáveis para guardar as velocidades
  oid = OBUVpropia //Velocidade propria
  Inst = snmpgetnext(OBU.IP,161,"public",oid)
  Vel = inst //Atribui o valor da velocidade a nova variavel
  oid = OBUVFrente //Velocidade do da frente
  Inst = snmpgetnext(OBU.IP,161,"public",oid)
  velFrente = inst //Atribui o valor da velocidade a nova variavel
  if(vel > velFrente){ Alarme }
  Oid = inst.oid;
  } //while
```

Teste Recurso

No anexo pode encontrar uma especificação dum grupo duma MIB experimental. Esta MIB pretende abstrair um conjunto de objectos que permita a gestão duma rede de leitores Radio Frequency Identification (RF-ID) utilizada para monitorizar a linha de produção duma fábrica de montagem de automóveis. A arquitetura considera que um agente SNMP está instalado em cada leitor RF-ID e implementa esta MIB. A cada leitor estão ligadas um conjunto de N antenas. Cada antena é identificada por um número/índice unívoco em cada leitor. Cada antena faz o "scan" duma determinada área de montagem e nessa área só pode estar um determinado carro a ser montado. Cada carro tem uma tag (etiqueta) RF-ID que identifica univocamente um carro. A tag é uma string de seis dígitos. Quando um carro está numa área de montagem a antena respetiva lê o valor da tag para a linha da tabela da MIB correspondente a essa antena. Quando não existe nenhum carro numa área a antena respetiva devolve o valor "000000" para indicar que não tem nenhuma tag válida ao alcance. A rede completa de leitores é monitorizada por uma aplicação de gestão SNMP: o software gestor interage com cada um dos agentes nos leitores. A aplicação gestora tem uma lista dos endereços IPs de todos os leitores. Tendo em conta a especificação da MIB responda às seguintes
questões:

a) **Que sequência de comandos (completos) do Net-SNMP usaria para obter a descrição das áreas sem carros dum leitor com um agente SNMP em 192.168.222.222:161 e community string igual a "public"? Esquematize um exemplo dos valores das instâncias dos objetos implementados no agente SNMP desse leitor e indique qual deveria ser o resultado dos comandos partindo do princípio que tinha quatro antenas ligadas, uma das quais sem nenhum carro em montagem.**

Zona→DisplayString
LeitorRFID→OctetString(6)
Agente→ipAdress
nAreas→integer
tableRFID→Sequence of RFIDtableEntry



Nr de RFID
Snmpgetnext -v2c -c public 192.168.222.222:161 experimental.ScanRFID.ScanRFIDNumber
R: ScanRFIDNumber.0=Integer:4

Tags
Snmpbulkget -v2c -c public 192.168.222.222:161 -Cn0 -Cr4
experimental.ScanRFID.SCcanRFIDTable.ScanRFIDTableEntry.ScanRFIDTag
R:ScanRFIDTag.1=String:"123456"
ScanRFIDTag.2=String:"654321"
ScanRFIDTag.3=String:"234567"
ScanRFIDTag.4=String:"000000"

Snmpget -v2c -c public 192.168.222.222:161 -Cn0 -Cr4
experimental.ScanRFID.SCcanRFIDTable.ScanRFIDTableEntry.ScanRFIDArea.3
R:ScanRFIDTArea.4=String:"Tires"

Index	Tag	Area
1	123456	Motor
2	654321	Electric
3	234567	Doors
4	000000	Tires

b) Descreva as alterações que teria que fazer na MIB se desejasse mudar a arquitetura por forma a todas as áreas/antenas estarem ligadas a um único leitor RF-ID e também que fosse possível contabilizar o número de carros que passou em cada área e o tempo médio que cada carro demorou em cada área.

Um único leitor RF-ID possui um conjunto de X antenas. Como cada antena realiza um scan duma determinada área de montagem, e nessa área só pode se encontrar apenas 1 carro, então utilizava-se o time-stamp Timeticks numa coluna extra de forma a controlar quando o carro entrou na determinada área e outra coluna time-stamp timeticks para controlar quando o carro saía da área. Desta maneira é possível obter o tempo médio que cada carro demorou em cada área. Para obtenção do número de carros é criada outra coluna com o nome "contador" que assim que é obtido o valor de tempo de saída do carro essa linha é incrementada para essa determinada área, realizando se o mesmo processo para todas as outras áreas e obtendo-se por fim o número total de carros em cada área.

c) Escreva um algoritmo duma função duma aplicação SNMP gestora que permita, usando a MIB alterada da alínea b), contabilizar o número de carros que passou por uma determinada área e o tempo médio que cada carro demorou nessa área.

```
count = 0 //Contar o numero de carros em cada area

passagemDesignacao = 6 //Existem 7 designações "Motor, Electric ...."

tempo = 0 //Guardar o tempo médio final por area

String designação = "motor"; //String para comparar o oid = scanRFIDArea

index = snmpget(scanRFID.1) //Todos os sensores

N = snmpget(Leitores.IP,161,"public",scanRFIDNumber.0) //De modo a verificar as antenas

Oid = scanRFIDArea //Variavel para comparar

Oid2 = scanRFIDTempoInicial //Variavel para comparar

Oid3 = scanRFIDTempoFinal //Variavel para comparar


Oid4 = scanRFIDTag //Variavel para comparar

While( passagemDesignacao != 0){ //Enquanto não verificar todas as areas

Do (N<index) { //Enquanto não vir as antenas para a determinada area

Inst = snmpgetnext(leitores.IP,161,"public",OID)

Inst4 = snmpgetnext(leitores.IP,161,"public",OID4)

Oid = inst.oid

Oid4 = inst.oid4

Se (inst.area = designacao && inst4.tag != "000000") { //verificar se existe e pertença à
determinada área

Inc(count) //incrementa o contador para obter o numero de carros

Inst2 = snmpgetnext(leitores.IP,161,"public",OID2)

Oid2 = inst2.oid2

Inst3 = snmpgetnext(leitores.IP,161,"public",OID3)

Oid3 = inst3.oid3

Tempo = Tempo + (inst2.tempoInicial – inst3.tempoFinal) //Guarda-se o tempo que os carros
estiveram na montagem

} }//IF e Do
```

```
Result = ( Tempo / count ) //O tempo de todos a dividir pelo número deles dá o tempo médio de cada carro

Result2 = count

//Obteve-se o tempo médio para os carros na área designada por "Motor" e o número de carros. Agora repete-se o ciclo para as restantes áreas.

passagemDesignacao = passagemDesignacao – 1 //Decrementa-se o ciclo das designações das áreas de montagem

Switch(passagemDesignacao){ //Switch para mudar o nome da variavel

        Case 0:

                Designacao = "Electric"

                Break;

        Case 1:

                Designacao = "Doors"

                Break;

        Case 2:

                Designacao = "Tires"


                Break;

        Case 3:

                Designacao = "Setas"

                Break;

        Case 4:

                Designacao = "Interior"

                Break;

        Case 5:

                Designacao = "Paiting"

                Break;

}//switch

}//while
```

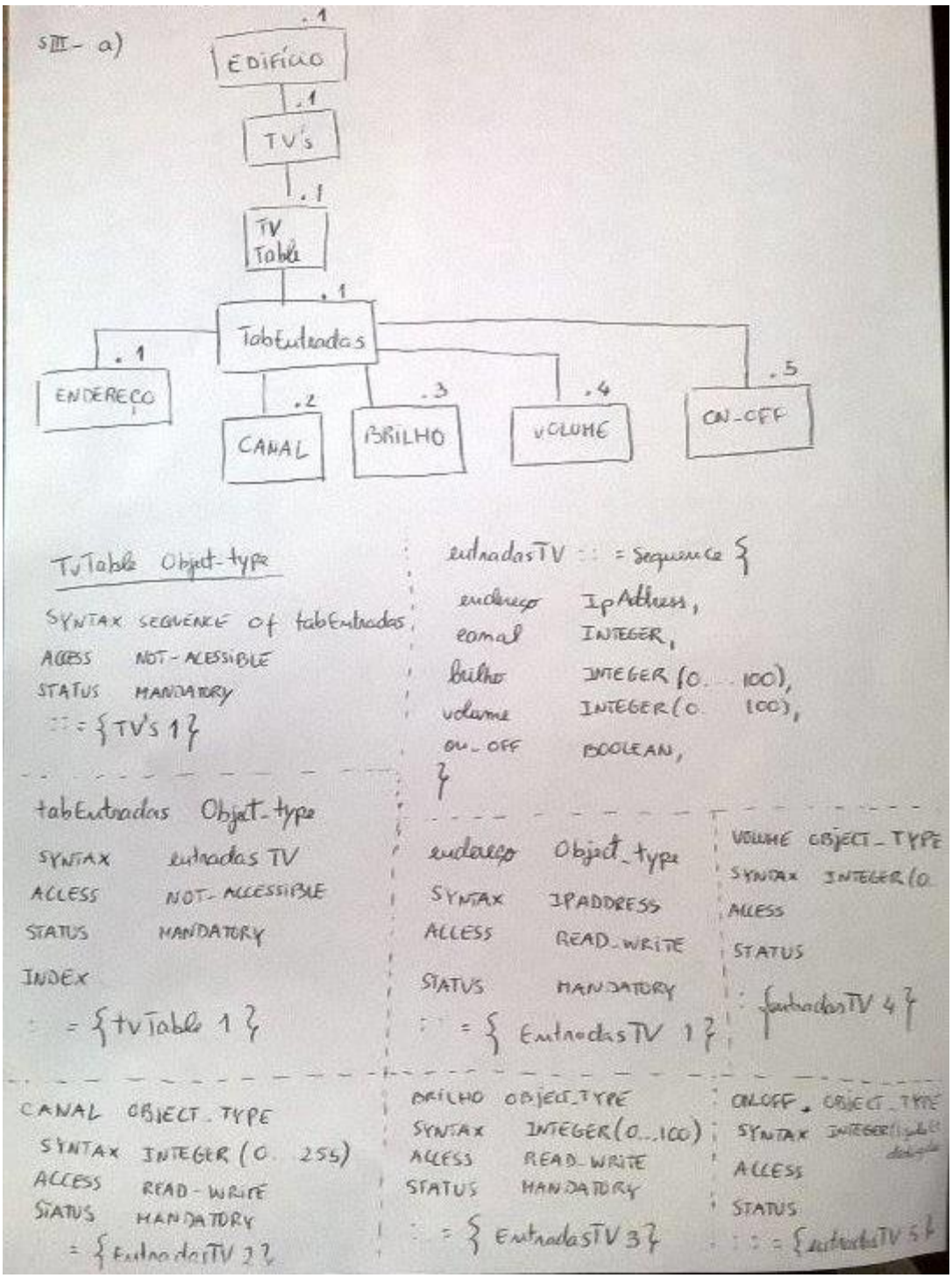
ANEXO

```
scanRFIDNumber OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION "The number of areas/antenas in scanRFIDTable."
    ::= { scanRFID 1 }

-- The scanRFIDTable table

scanRFIDTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF ScanRFIDEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION "This entity's table of antenas."
    ::= { scanRFID 2 }

scanRFIDEntry OBJECT-TYPE
    SYNTAX  ScanRFIDEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION "An entry (row) for information about data scanned by
one antenna in one area. The key for the table is the antenna's index
number."
    INDEX   { scanRFIDIndex }
    ::= { scanRFIDTable 1 }

ScanRFIDEntry::= SEQUENCE {
        scanRFIDIndex  INTEGER,
        scanRFIDTag    OCTET STRING[6],
        scanRFIDArea   DisplayString
    }

scanRFIDIndex OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION "The index of the antenna in this group of antennas."
    ::= { scanRFIDEntry 1 }

scanRFIDTag OBJECT-TYPE
    SYNTAX  OCTET STRING[6]
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION "The scanned value of the tag in the area. A value
        under "000000" means there's no tag in the area."
    ::= { scanRFIDEntry 2 }

scanRFIDArea OBJECT-TYPE
    SYNTAX  DisplayString
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION "A text containing the name identifying the production
area where the antenna is installed. Examples: "Motor", "Electric",
"Doors", "Tires", "Setas", "Interior", "Paiting"."
    ::= { scanRFIDEntry 3 }
```

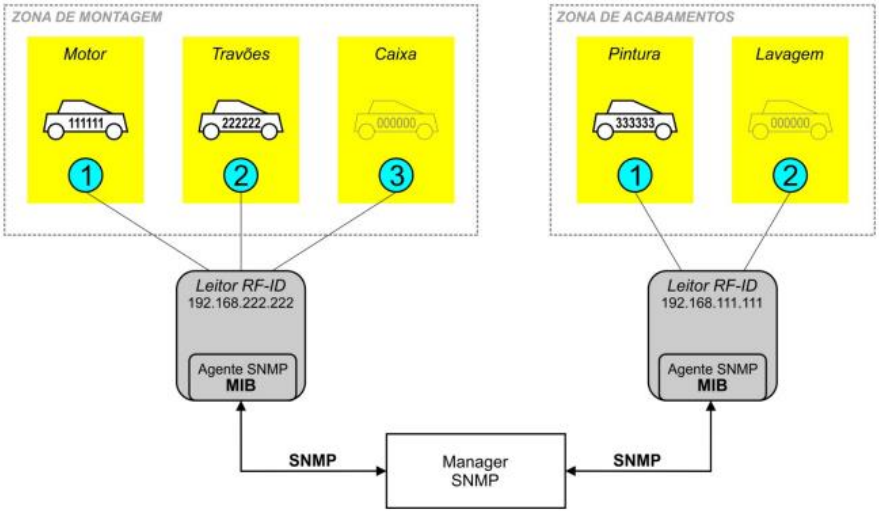
Teste/exercício

Suponha que é ficou incumbido de criar um agente SNMP para um modelo de televisão especial, instalado em edifícios públicos para informar e distrair os utilizadores de serviços públicos, como por exemplo, repartições de finanças, lojas do cidadão, supermercados, estações do correio, etc. Os parâmetros que interessa controlar e monitorizar remotamente são o canal a visualizar, o volume do aúdio, o valor do brilho e poder ligar e desligar remotamente o aparelho. Todas as TVs têm um endereço IP na rede local.

1. Defina uma MIB por forma a permitir a gestão remota deste aparelho através de uma aplicação central que estará instalada num PC na sala do gestor do edifício.



Exercício Aula



Defina uma MIB em que se pretende abstrair um conjunto de objetos que permita a gestão duma rede de leitores *Radio Frequency Identification* (RF-ID) utilizada para monitorizar a linha de produção duma fábrica de montagem de automóveis (ver Figura 1). A arquitetura deve considerar que um agente SNMP está instalado em cada leitor RF-ID e implementa uma instância desta MIB. A cada leitor estão ligadas um conjunto de N antenas. Cada antena é identificada por um número/índice unívoco em cada leitor. Cada antena faz o "scan" duma determinada área de montagem e nessa área só pode estar um determinado carro a ser montado. Um conjunto de áreas monitorizadas pelo mesmo leitor define uma Zona. Cada carro tem uma *tag* (etiqueta) RF-ID que identifica univocamente um carro. A informação armazenada numa *tag* é uma string de seis dígitos. Quando um carro está numa área de montagem a antena respetiva lê o valor da *tag* para a linha da tabela da MIB correspondente a essa antena. Quando não existe nenhum carro numa área a antena respetiva devolve o valor "000000" para indicar que não tem nenhuma *tag* válida ao alcance. A rede completa de leitores é monitorizada por uma aplicação de gestão SNMP: o *software* gestor interage com cada um dos agentes nos leitores. A aplicação gestora tem uma lista dos endereços IPs de todos os leitores. Tendo em conta a especificação da MIB responda às seguintes questões:

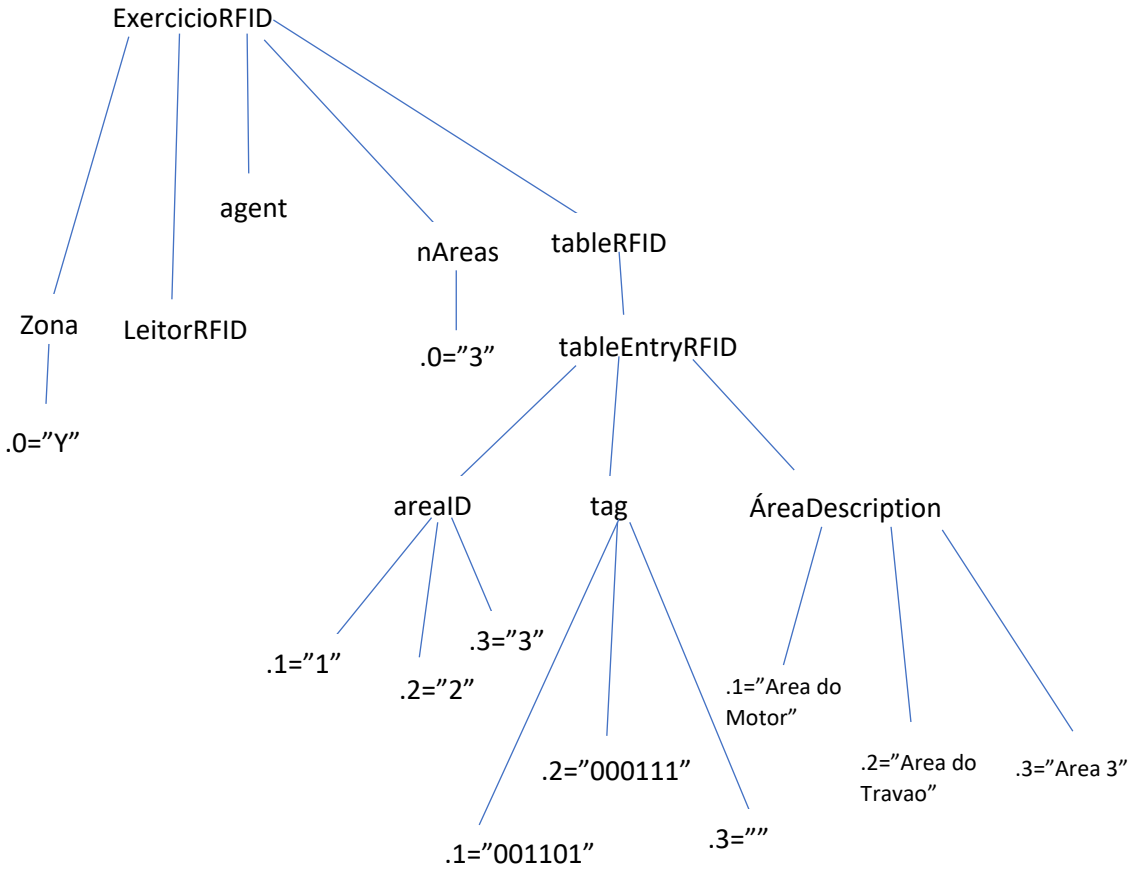
a) Defina um grupo de objetos numa MIB que permita a persecução dos requisitos descritos.

Definir um grupo de objetos:

Zona "Y"→String(DisplayString)
Leitor id→octet String
Agente SNMP→ipadress→ipAdress
Nº de antenas/nº de áreas→integer
Areas→sequence of entry

Areas	Tag	Descrição
1	"001101"	"Area de leitor"
2	"000111"	"Area dos travões"
3	" "	"Area 3"
INTEGER	OCTET STRING(SIZE(6))	DISPLAY STRING

Em arvore:



b) Que sequência de comandos (completos) do Net-SNMP usaria para saber se a área de "Motor" da zona de Montagem da Figura 1 estava ocupada com um carro? Considere que os agentes SNMP atendem na porta 161 e a sua *community string* é igual a *"public."* Esquematize um exemplo dos valores das instâncias dos objetos implementados nos agentes SNMP dos dois leitores e indique qual deveria ser o resultado dos comandos indicados.

1ºQuantas áreas?

Snmpget -v2c -c public 10.10.10.10:161 experimental.exercicioRFID.nAreas.0

(snmpget mais eficiente que snmpgetnext)

2ºObter descrições das 3 areas?

Snmpbulkget -v2c -c public 10.10.10.10:161 -Cn0 -Cr3 experimental.exercicioRFID.tableRFID.tableEntryRFID.areaDescription

3ºObter tag respetiva?

Snmpget -v2c -c public 10.10.10.10 experimental.exercicioRFID.tableRFID.tableEntryRFID.tag.1

Tag-(objeto)

1-(instancia)

QUESTAO TEÓRICA

A arquitetura de gestão SNMP também tem sido aplicada a contextos não exclusivamente ligados à gestão de equipamentos de rede. No entanto, existem, com certeza, casos de gestão de equipamentos ou aplicações distribuídas em que a sua aplicação não é aconselhada ou que se reveste de grande complexidade. Dê um exemplo destes explicando as razões da sua sugestão.

O facto de o SNMP ser um protocolo assíncrono, e não confirmado, impede que este seja aplicado em contextos de sistemas críticos tais como na aviação. Visto que no protocolo SNMP o gestor só sabe se o agente recebeu o seu pedido se o agente lhe enviar a resposta ao mesmo, faz com que seja altamente insatisfatório e perigoso no âmbito da aviação visto que qualquer pedido de monitorização por parte da aplicação gestora tem de ser imediatamente respondida pelo agente. Exemplificando concretamente, imaginemos que o agente falhava a verificação da quantidade de combustível presente nos tanques do avião porque estava demasiado sobrecarregado com outros pedidos ou então porque estes pedidos eram demasiado frequentes, é uma situação inimaginável a aplicação do SNMP nestes casos.
No caso da aviação é necessário um protocolo síncrono, e confirmado, requisitos que o SNMP não fornece.

OSI VS TCP/IP

TCP/IP(ENLACE É LOGICO)

OSI

Síncrono vs Assíncrono

Assíncrono responde qdo quer