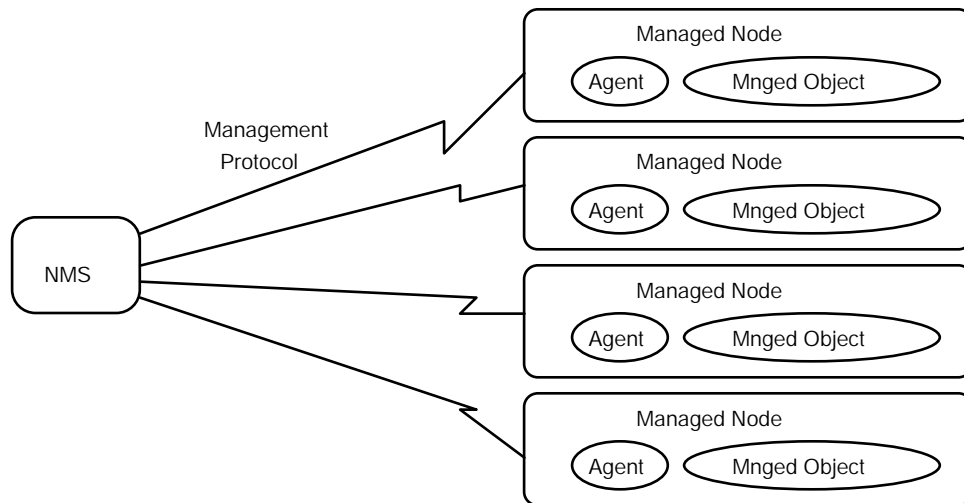


SNMP

Simple Network Management Protocol

I. Conceptos básicos



Managed Nodes

hosts, servers, routers, hubs, bridges

Agent

NMS "Network Management Station"

Management Protocol **SNMP**

Managed Objects & Management Information Base **MIB**

Management Operations

Read, Write, Traversal, Trap

Proxy Agents

II. Data Representation

ASN.1 (Abstract Syntax Notation One)

Abstract Syntax

p/definir formatos de PDUs

p/definir estructuras de datos asociadas a objetos administrables

Types

Define tipos de datos. Los labels empiezan con letra mayúscula.

Tipos elementales:

INTEGER

Número entero

Para valores lógicos usar: up(1) o down(2)

OCTET STRING

0 o más octetos que pueden tomar valores 0..255

OBJECT IDENTIFIER

denota un authoritatively (único) named object

Secuencia de enteros no negativos

Ej: 1.3.6.1.2.1 o sea iso.org.dod.internet.mgmt.mib

NULL

Tipos estructurados (constructed types):

SEQUENCE

Secuencia ordenada de 0 o más ASN.1 types

SEQUENCE OF TYPE

Secuencia ordenada de 0 o más elementos de un ASN.1 type

Sub-Tipos (subtypes):

IpAddress

String de 4 octetos

Counter

Entero positivo 0..2³²

Values

Instancias de un tipo de datos. Los labels empiezan con letra minúscula.

Macros

Para cambiar la gramática del lenguaje. Los labels llevan todas mayúsculas.

III. Managed Objects

SMI (Structure of Management Information)

Define las reglas para describir objetos administrables (managed objects)

Todo objeto administrable tiene asociada una sintáxis y una semántica

Una variable es una instancia de un objeto

SMI define el esquema para la database de los objetos administrables

MIB (Managed Information Base)

Es la database de los objetos administrables

Cada objeto tiene:

name (Object Identifier)

type (Object Type)

access (read-write, read-only, not-accessible, write-only) ;

status (mandatory, optional, obsolete)

Ej:

sysDescr OBJECT-TYPE

SYNTAX OCTET STRING

ACCESS read-only

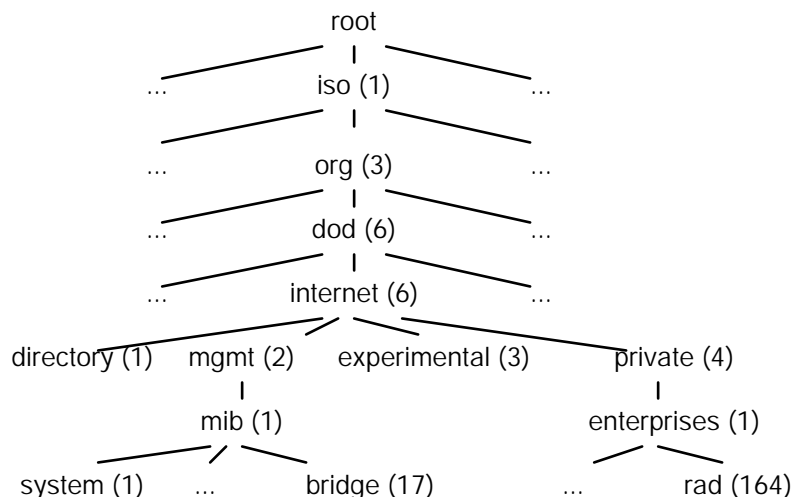
STATUS mandatory

::= {system 1}

Object Names

Es el OBJECT IDENTIFIER de los objetos administrables.

Ej: internet OBJECT IDENTIFIER ::= {iso org(3) dod(6) 1} (o sea 1.3.6.1)



OBJECT IDENTIFIER.instance identifier

Para manipular las variables MIB de un objeto administrable, ya que éste es una instancia de objeto "object instance", se usa el protocolo SNMP haciendo referencia a cada variable MIB con el OBJECT IDENTIFIER seguido por el sufijo "instance identifier", tanto en el caso de objetos *escalares* como de objetos *tabulares*.

Ej:

sysDescr.0 (1.3.6.1.2.1.1.1.0) (instancia de variable escalar)

mib.interfaces.ifTable.ifEntry.ifDesc.2 (instancia de variable tabular)

IV. El protocolo SNMP

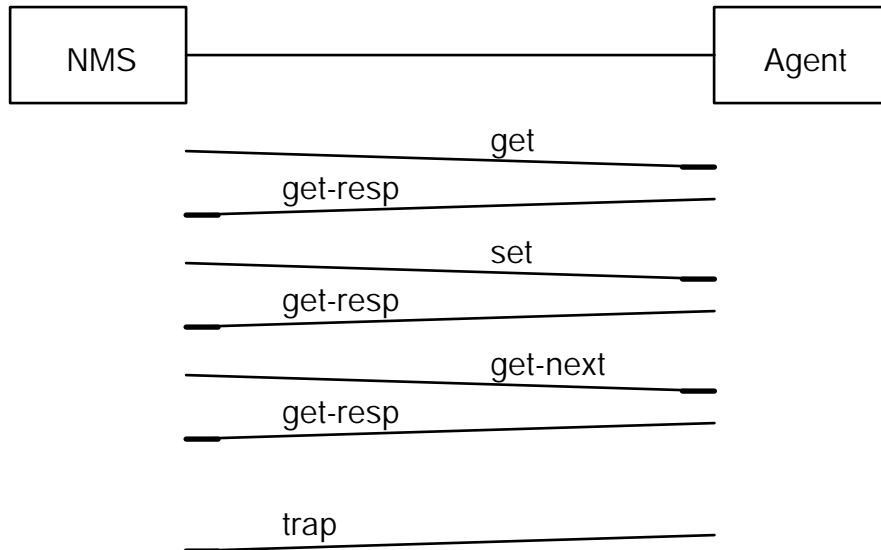
SNMP: **Simple Network Management Protocol**

Dicho protocolo permite manipular las variables MIB de los objetos administrables.

Es el protocolo "hablado" entre la **NMS** (Network Management Station) , y el **SNMP Agent**.

El protocolo SNMP soporta las operaciones:

<u>Operación</u>	<u>Descripción</u>
get	para recuperar el valor de una variable MIB
set	para modificar el valor de una variable MIB
get-next	para recuperar el valor de una variable MIB pero en modo de acceso "traversal"
get-response	para devolver el valor de una variable MIB
trap	para reportar eventos extraordinarios



Autenticación en SNMP

View: subset of manageable objects
 Access Mode: read-only, read-write
Community profile: View + Access Mode

SNMP entities: NMS & Agent

Community name

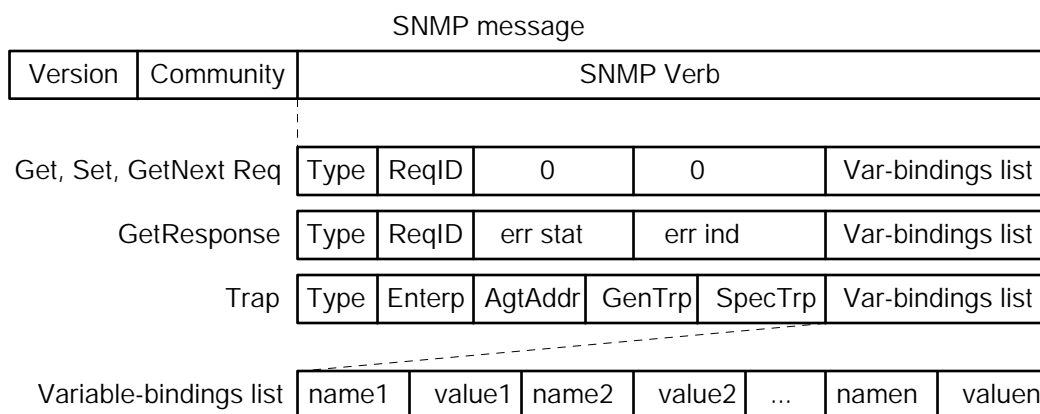
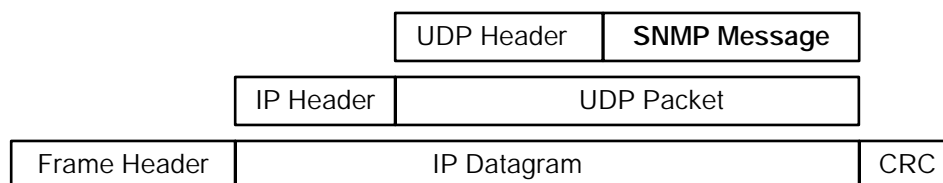
Authentic SNMP messages

Authetication failure

Ejemplo de configuración de un agent:

<u>community name</u>	<u>IP address</u>	<u>access</u>	<u>traps</u>	<u>view name</u>
public	0.0.0.0	read-only	no	all
public	192.9.200.1	read-only	yes	all
priv1	192.9.200.17	read-write	yes	all
priv2	150.30.25.4	read-write	yes	interfaces
priv3	192.9.200.55	read-only	yes	statistics

Formato de mensaje SNMP



Version: SNMP o SNMPv2

ReqID: p/matchear responses con requests

P.S.: La SNMP PDU no tiene longitud máxima. Si SNMPPDU > MTU ==> Error "Too big"

V. MIB-II (sintaxis ASN.1)

-- MIB-II groups

```
system      OBJECT IDENTIFIER ::= { mib-2 1 }
interfaces  OBJECT IDENTIFIER ::= { mib-2 2 }
at          OBJECT IDENTIFIER ::= { mib-2 3 }
ip          OBJECT IDENTIFIER ::= { mib-2 4 }
icmp       OBJECT IDENTIFIER ::= { mib-2 5 }
tcp        OBJECT IDENTIFIER ::= { mib-2 6 }
udp        OBJECT IDENTIFIER ::= { mib-2 7 }
egp       OBJECT IDENTIFIER ::= { mib-2 8 }
-- cmot    OBJECT IDENTIFIER ::= { mib-2 9 }
transmission OBJECT IDENTIFIER ::= { mib-2 10 }
snmp       OBJECT IDENTIFIER ::= { mib-2 11 }
```

-- system group

```
sysDescr OBJECT-TYPE
    SYNTAX DisplayString (SIZE(0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A textual description of the entity. ..."
 ::= { system 1 }
```

```
sysObjectID OBJECT-TYPE
    SYNTAX OBJECT IDENTIFIER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The vendors authoritative identification ...
        within the SMI enterprises subtree (1/3.6.1.4.1) ..."
 ::= { system 2 }
```

```
sysUpTime OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The time (in 1/100 secs) since...reinitialized."
 ::= { system 3 }
```

sysContact OBJECT-TYPE
SYNTAX DisplayString (SIZE(0..255))
ACCESS read-write
STATUS mandatory
DESCRIPTION
 "The textual id of the contact person ..."
 ::= { system 4 }

sysName OBJECT-TYPE
SYNTAX DisplayString (SIZE(0..255))
ACCESS read-write
STATUS mandatory
DESCRIPTION
 "An administratively-assigned name for this managed node.
 By convention, node's fully-qualified domain name"
 ::= { system 5 }

sysLocation OBJECT-TYPE
SYNTAX DisplayString (SIZE(0..255))
ACCESS read-write
STATUS mandatory
DESCRIPTION
 "The physical location of this node ..."
 ::= { system 6 }

sysServices OBJECT-TYPE
SYNTAX INTEGER (0..127)
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "A value which indicates the set of services ...
 layer functionality flags:
 1 physical (eg: repeater)
 2 datalink (eg: bridges)
 3 internet (eg: ip router)
 4 end-to-end (eg: ip host)
 7 aplications (eg: mail relay)
 ..."
 ::= { system 7 }

-- **interfaces** group

ifNumber OBJECT-TYPE
SYNTAX INTEGER (0..127)
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "The number of network interfaces ..."
 ::= { interfaces 1 }

-- the **interfaces** table

ifTable OBJECT-TYPE
SYNTAX SEQUENCE OF IfEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
 "A list of interface entries ..."
 ::= { interfaces 2 }

ifEntry OBJECT-TYPE
SYNTAX IfEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
 "... containing objects at the sublayer and below for a particular interface."
INDEX { ifIndex }
 ::= { ifTable 1 }

IfEntry ::=
SEQUENCE {
 ifIndex INTEGER,
 ifDescr DisplayString,
 ifType INTEGER,
 ifMtu INTEGER,
 ifSpeed Gauge,
 ifPhysAddress PhysAddress,
 ifAdminStatus INTEGER,
 ifOperStatus INTEGER,
 ...
 ifInOctets Counter,
 ifInErrors Counter,
 ...
 ifOutOctets Counter,
 ifOutErrors Counter,
 ...
}

ifIndex OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "A unique value for each interface. Ranges between 1 and value of ifNumber..."
 ::= { ifEntry 1 }

ifDescr OBJECT-TYPE
SYNTAX DisplayString (SIZE(0..255))
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "A textual description of the interface..."
 ::= { ifEntry 2 }

ifType OBJECT-TYPE
SYNTAX INTEGER {
 other(1),
 ...
 rfc877-x25(5),
 ethernet-csmacd(6),
 iso88023-csmacd(7),
 iso88024-tokenBus(8),
 iso88025-tokenRing(9),
 ...
 fddi(15),
 lapb(16),
 ...
 propPointToPointSerial(22),
 ppp(23),
 ...
 slip(28),
 ...
 frame-relay(32)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "The type of the interface..."
 ::= { ifEntry 3 }

ifMtu OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "The MTU of the interface..."
 ::= { ifEntry 4 }

ifSpeed OBJECT-TYPE
SYNTAX Gauge
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "Nominal or estimated bandwidth in bps..."
 ::= { ifEntry 5 }

ifPhysAddr OBJECT-TYPE
SYNTAX PhysAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "The inetface's address at the protocol layer inmediately below ..."
 ::= { ifEntry 6 }

ifAdminStatus OBJECT-TYPE
SYNTAX INTEGER {
 up(1),
 down(2),
 testing(3),
 }
ACCESS read-write
STATUS mandatory
DESCRIPTION
 "The desired state of the interface ..."
 ::= { ifEntry 7 }

ifOperStatus OBJECT-TYPE
SYNTAX INTEGER {
 up(1),
 down(2),
 testing(3),
 }
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "The current operational state of the interface ..."
 ::= { ifEntry 8 }
 ...

ifInOctets OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The total number of octets received ..."

::= { ifEntry 10 }

...

ifInErrors OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The number of inbound packets that contained errors ..."

::= { ifEntry 14 }

ifOutOctets OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The total number of octets transmitted ..."

::= { ifEntry 16 }

...

ifOutErrors OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The number of outbound packets that could not be transmitted because of errors ..."

::= { ifEntry 20 }

...

-- ip group

ipForwarding OBJECT-TYPE
ipDefaultTTL OBJECT-TYPE
ipInReceives OBJECT-TYPE
ipInDelivers OBJECT-TYPE
ipForwDatagrams OBJECT-TYPE
ipOutNoRoutes OBJECT-TYPE
ipReasmOKs OBJECT-TYPE
ipFragOKs OBJECT-TYPE
ipAddrTable OBJECT-TYPE -- (~ ifconfig -a))
ipRoutingTable OBJECT-TYPE -- (routing table)
ipNetToMediaTable OBJECT-TYPE -- (arp table)

...

-- icmp group

-- por cada icmp message type tiene un counter de in y otro de out (26 counters)
-- además 4 counters de icmp messages received, sent, recinerror, notsenterr
icmplnEchos OBJECT-TYPE

...

-- tcp group

tcpMaxConn OBJECT-TYPE
tcpCurrEstab OBJECT-TYPE
tcpInSegs OBJECT-TYPE
tcpConnTable OBJECT-TYPE

...

-- udp group

udpInDatagrams OBJECT-TYPE

...

-- egp group

egpInMsgs OBJECT-TYPE

...

-- snmp group

snmplnReadOnlys OBJECT-TYPE
snmplnGetRequests OBJECT-TYPE
snmplnTraps OBJECT-TYPE
snmpOutTraps OBJECT-TYPE

...

VI. Traps

Un trap es enviado por un Agent a una NMS.

El mensaje de tipo trap es enviado indicando:

sysObjID del agent que lo generó (enterpriseID)

network address del agent que lo generó

Generic Trap number

Specific Trap Number

timestamp que es el sysUpTime de cuando se produjo el evento

variable list que provee información adicional relacionada con el trap

Hay 7 generic trap numbers:

coldStart

warmStart

linkDown

linkUp

authenticationFailure

egpNeighborLoss

enterpriseSpecific

Hay una enorme cantidad de **Enterprise Traps**, por ejemplo:

"Temperature has reached danger point"

"Load balance conflict"

En el agent se pueden definir niveles de **threshold** para decidir si un evento debe o no generar un trap.

VII. MIB Extensions

Ya que MIB-II sólo provee información estadística sobre IP, TCP, UDP, SNMP, para fines de monitoreo y medición de performance se han definido **MIB extensions** que están formalmente descritas en RFCs.

También ha sido necesario que se definan **Private MIB Objects** para satisfacer necesidades tales como load balancing, filtering en bridges y routers, protocolos distintos de la tcp/ip protocol suite, etc.

RMON

RFC 1271

Remote Network Management Goals

for Remote Monitoring of Networks:

network traffic statistics, hosts address table, hosts statistics, historical statistics, thresholds, packet/protocol analysis, ...

mgmt.mib.rmon(16) groups:

Statistics, History, Alarms, Hosts, HostTopN, TrafficMatrix, PacketCapture, Events.

REPEATER

RFC 1516

for link testing, network traffic statistics, MAC address table, hosts statistics

BRIDGE

RFC 1493

for link testing, network traffic statistics, STP performance, WAN Link performance

HOST

RFC 1514

for host job counts, host file system info

VIII. Generic & Specific NMS Applications

Hay varias Generic NMS Applications que trabajan con SNMP:

OpenView (HP)

SunNetManager (Sun)

NetView/6000 (IBM)

PC/SNMP Tools (FTP Soft)

Hay también un gran número de Specific NMS Applications para monitorear y controlar dispositivos en red que son provistas por los mismos fabricantes de dichos dispositivos, pero que no pueden ser usadas cuando se dispone de una gran cantidad de dispositivos multivendedor.

IX . Otros protocolos de Network Management

La ISO propuso una serie de protocolos de Network Management conformes a su modelo de referencia OSI.

Estos se denominan **CMIS** (Common Management Information Service) y **CMIP** (Common Management Information Protocol).

El protocolo **CMOT** es la implementación de CMIS/CMIP sobre conexiones TCP (CMis/ip Over Tcp)

Referencias Bibliográficas

1. Comer D., "Internetworking with TCP/IP", Prentice-Hall, 1991