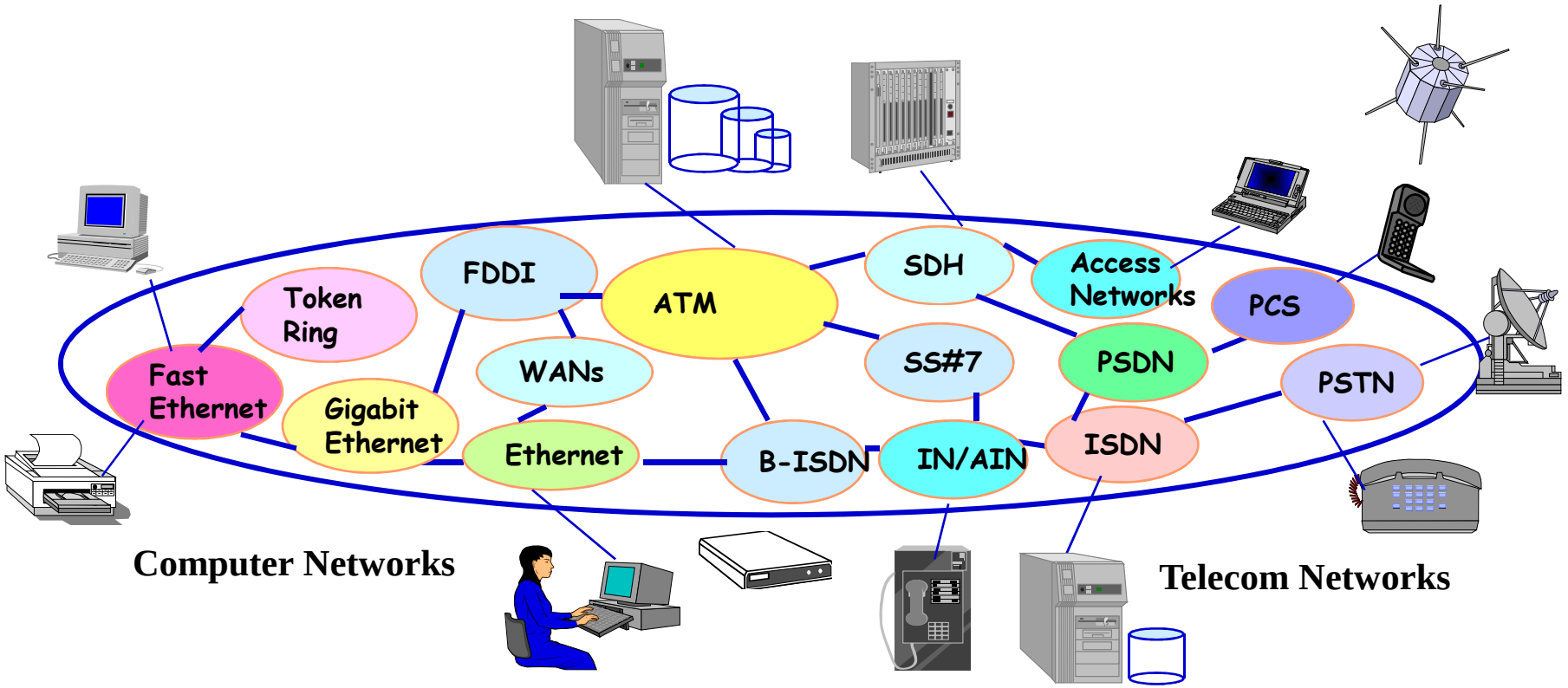
Protocole d 'administration des réseaux SNMP

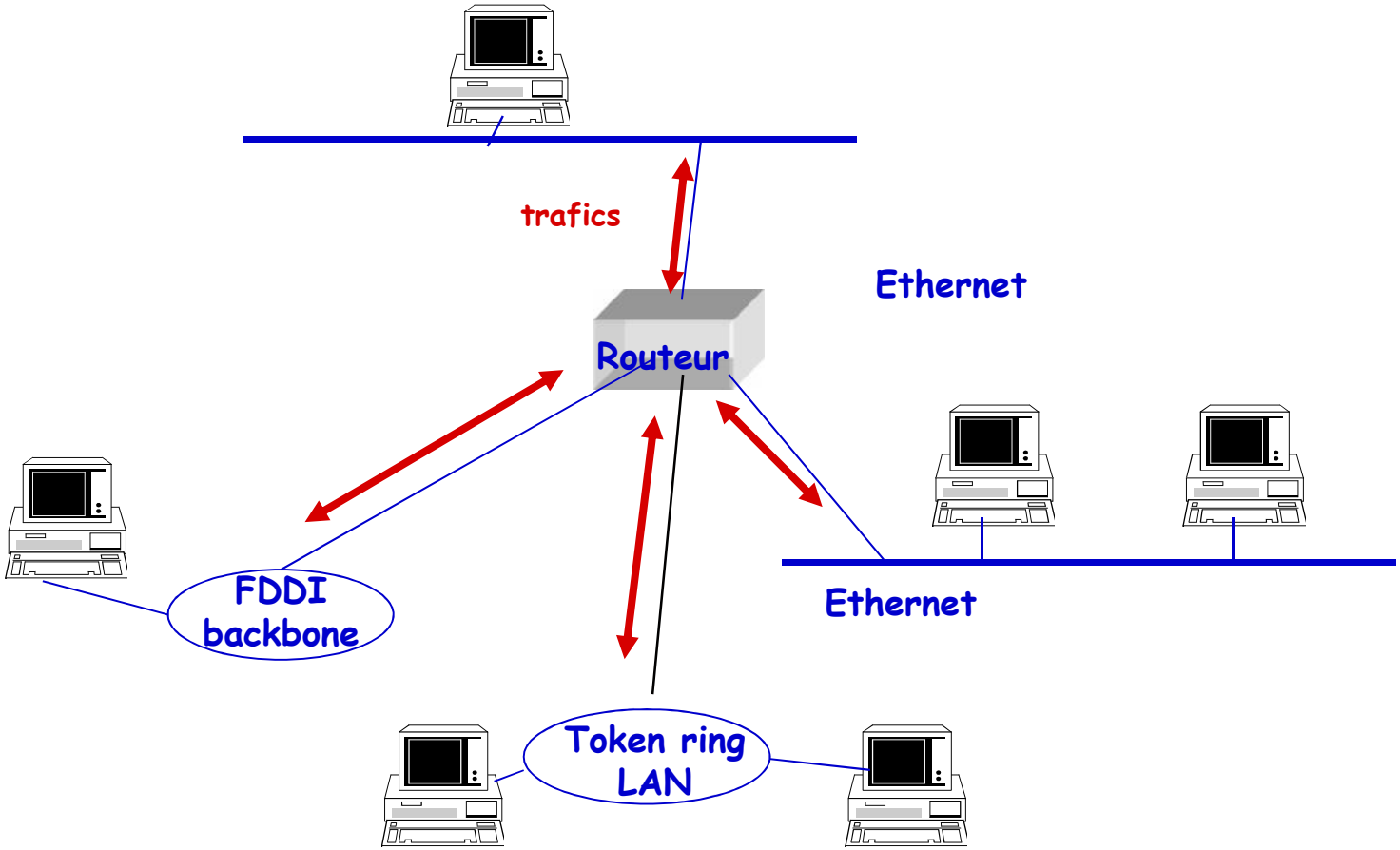
(Simple Network Management Protocol)

Motivation


FDDI
ATM
SDH
Access Networks
Token Ring
PCS
Fast Ethernet
WANs
SS#7
PSDN
PSTN
Gigabit Ethernet
Ethernet
B-ISDN
IN/AIN
ISDN
Computer Networks
Telecom Networks

Motivation

- Nécessité d 'avoir un protocole permettant de remonter des informations sur l 'activité des différentes ressources du réseau notamment les serveurs, les routeurs, les switchs, etc.



Introduction

- L 'administration des réseaux est un ensemble de techniques permettant de maîtriser les aspects techniques, financiers, organisationnels, et de sécurité d 'accès aux informations.
- L 'aspect technique doit garantir la qualité de service sous tous ses formes et en particulier la continuité de services.
- L 'aspect financier doit permettre d 'évaluer tous les coûts introduits par le mode d 'utilisation du réseau.
- L 'aspect organisationnel permet de contrôler la structure du réseau et de son évolution.
- L 'aspect sécurité couvre la confidentialité des données et le contrôle d 'accès à ces données.

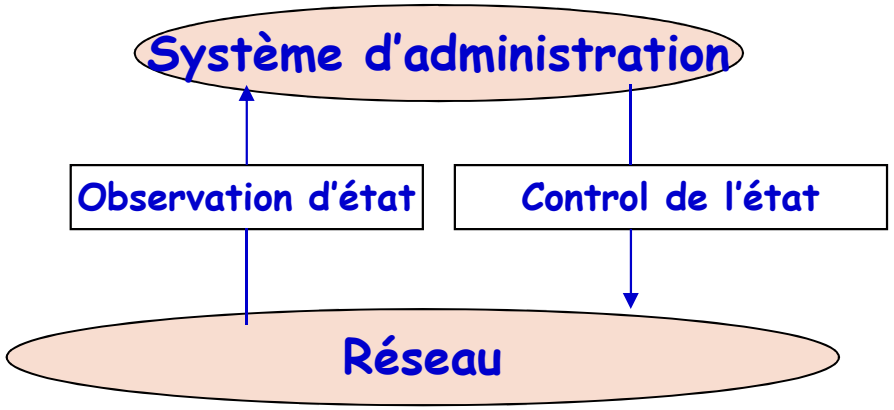
Administration réseaux

- Exploiter, Opérer, Gérer les ressources du réseau pour délivrer un service aux utilisateurs du système d'information

- Garantir une QoS à ces utilisateurs
  - La QoS se mesurera au travers de divers métriques caractérisant le fonctionnement, souvent un appelé contrat de service.

Administration réseaux

- Les fonctions d 'administration des réseaux peuvent être groupées en deux catégories : la surveillance et le contrôle.
- La surveillance ou l 'observabilité est reliée à la lecture, l 'observation et l 'analyse de l 'état du réseau et la configuration de ses éléments.
- Le contrôle est relié à l 'écriture et la modification de la configuration des paramètres de plusieurs éléments du réseau.



objectifs

- Gérer de nombreuses plates-formes réseau différentes et hétérogènes.
- Permettre l'accès à des équipements distants sur des réseaux physiques différents
- Automatiser le processus de supervision de l'utilisation et des performances du réseau
- Automatiser le processus de configuration et de contrôle des éléments du réseau
- Automatiser la gestion des erreurs sans surcharger le réseau
  - Ex: détection de liens surchargés pour changer l'architecture

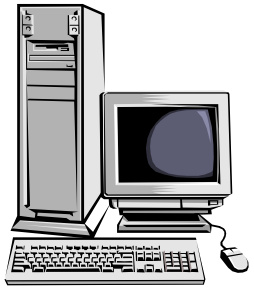
SNMP: Simple Network Management Protocol

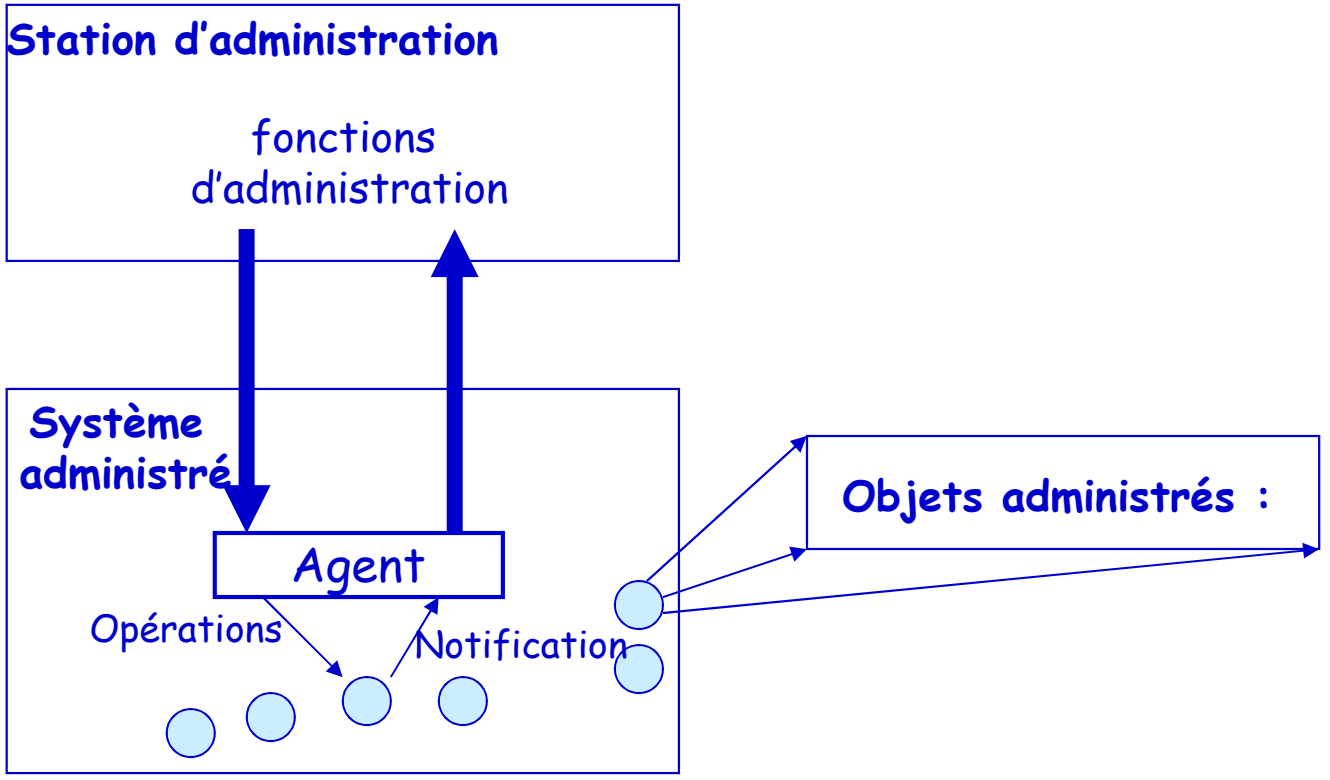
- Protocole de la **couche application** qui permet de gérer les équipements du réseau et de diagnostiquer les problèmes de réseau
  - Indépendance au type de réseau et à l'architecture de la machine administrés
- Permet d'accéder aux données d'informations d'administration
  - Ex: le nombre de paquets par seconde envoyés sur une interface, nombre de connexions TCP ouvertes, etc.
- Modèle organisationnel d'administration réseau SNMP
  - La station d'administration
  - L'agent de supervision
  - La base d'informations de management
  - Le protocole de gestion de réseau

Le modèle d 'administration des réseaux avec SNMP

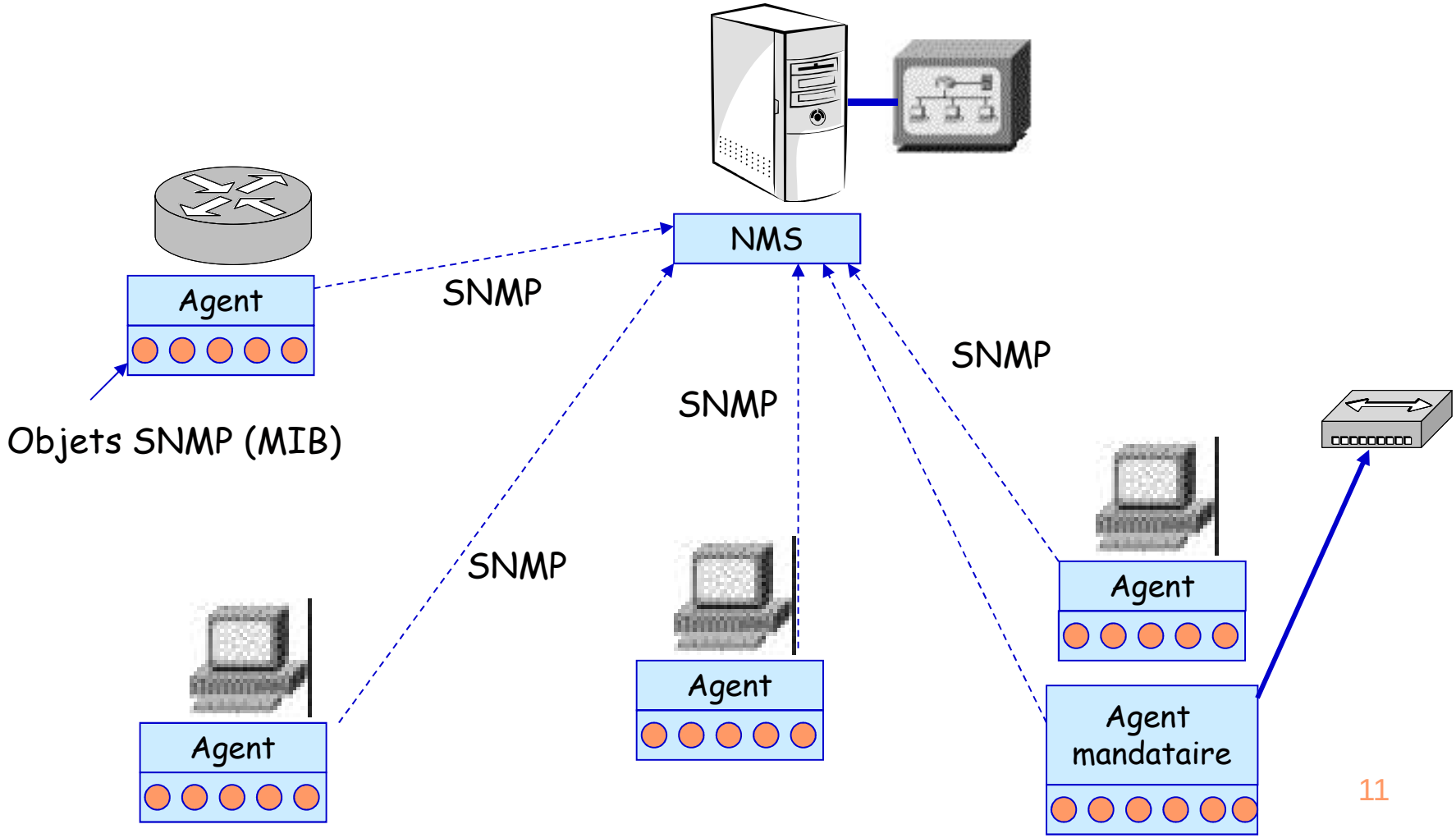
- Une administration SNMP est composée de **trois types d'éléments** :
  - des **agents** chargés de superviser un équipement. On parle d'agent SNMP installé sur tout type d'équipement.
  - une ou plusieurs **stations de gestion** capablent d'interpréter les données
  - une **MIB** (Management Information Base) décrivant les informations gérées (objets administrés).
- Un **protocole** activé par une API permet la supervision, le contrôle et la modification des paramètres des éléments du réseau.

Le modèle d 'administration des réseaux



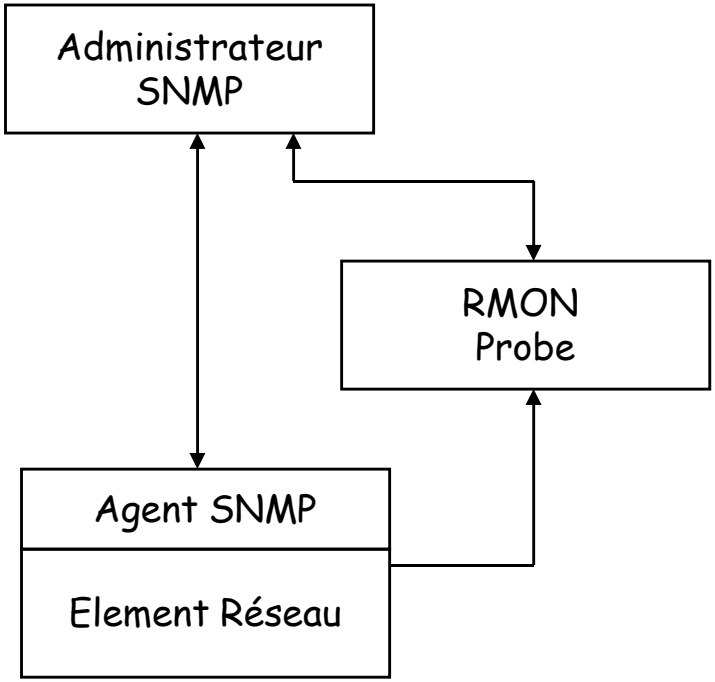


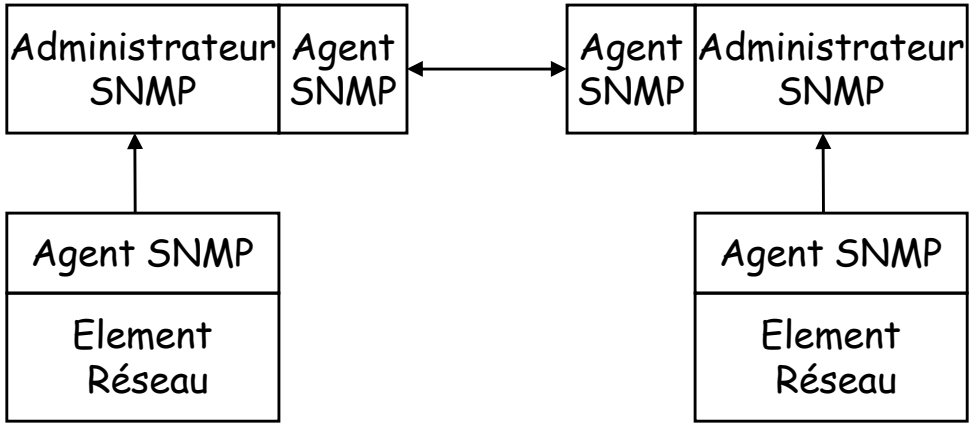
Modèle organisationnel d'administration réseau SNMP



Le modèle d 'administration des réseaux avec SNMP

- L 'architecture trois-tiers insère entre le Manager et l 'agent une sonde RMON ou une autre station d 'administration (modèle SNMPv2).
  - La sonde RMON permet de faire la collecte d 'informations d 'administration et quelques traitements sur le trafic.

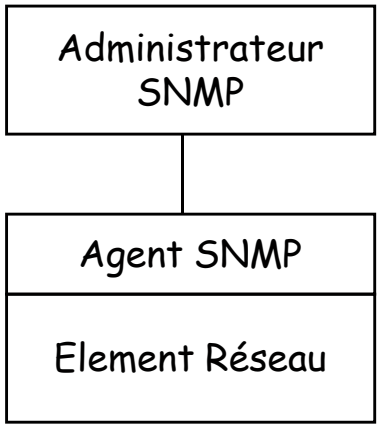


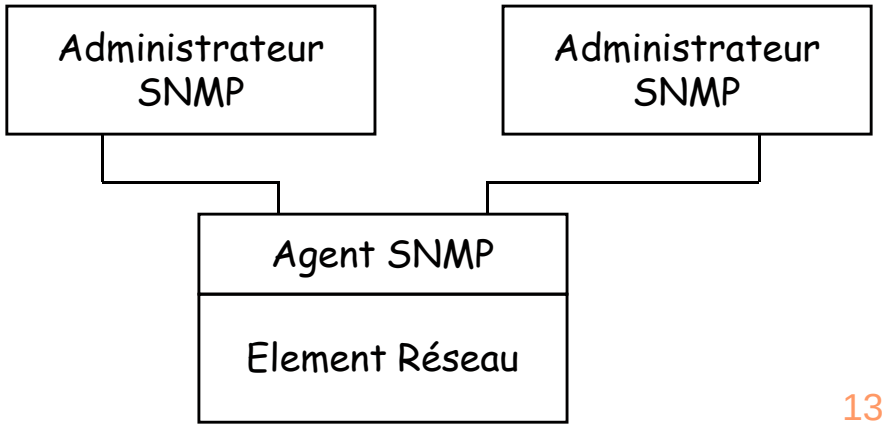


Supporté par snmpv2

Le modèle d 'administration des réseaux avec SNMP

- Le modèle « Manager-Agent » est appelé modèle deux-tiers.
  - L 'administrateur reçoit les informations de l 'agent et les traite.
  - Possibilité d 'avoir plusieurs stations d 'administration qui reçoivent les informations d 'un Agent SNMP.
  - Ce modèle ne permet pas d 'avoir directement des méta-données (trafic, statistiques ...) sans faire le traitement adéquat.





NMS: Network Management System

- Console au travers de laquelle les administrateurs peuvent réaliser des tâches d'administration.
  - Peut être mise en œuvre sur plusieurs systèmes.
- Exécute un client SNMP
- Contient un ensemble de logiciels pour l'administration et le supervision, appelés l'application d'administration réseau (NMA)
- Exemples de produits
  - HP Openview
  - IBM Tivoli Netwiew
  - Sun Enterprise Manager
  - EvidianOpenMaster
  - Net-snmp

Agent de supervision

- Logiciel modulaire résidant dans des équipements réseaux tels que hôtes, les routeurs, les ponts et les concentrateurs.
- Répond à des requêtes d'informations et des requêtes d'action émises par la NMS, telles que l'interrogation, et peut fournir à la NMS des informations très importantes, mais non demandées, telles que les Traps.
- Stocke les informations de supervision dans la base d'informations de management (MIB)
- Exécute un serveur SNMP

Agent de supervision

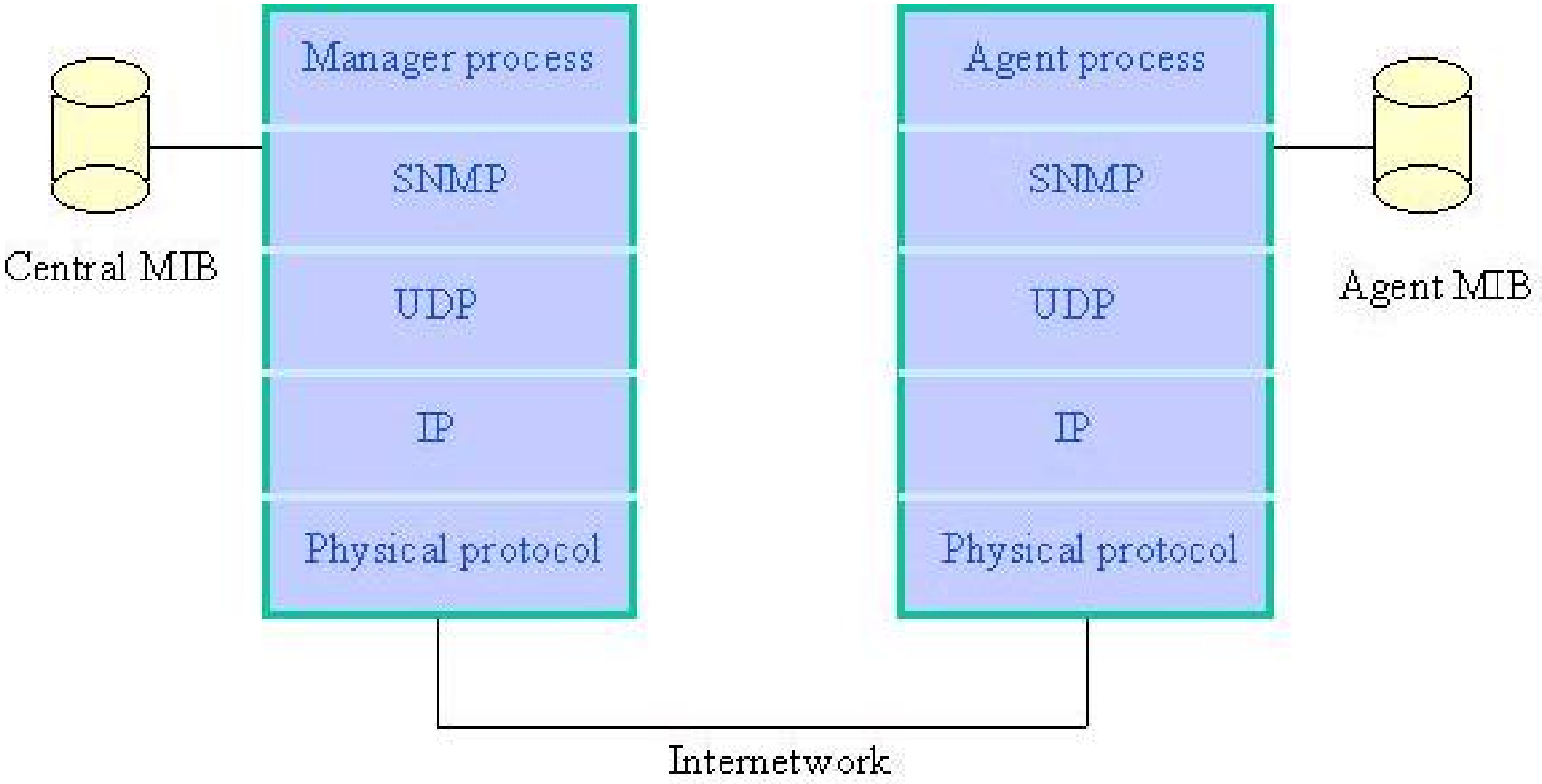
- Peut effectuer un suivi des éléments suivants:
  - Le nombre de certains types de messages d'erreur reçus
  - Le nombre d'octets et de paquets entrant et sortant de l'équipement
  - Les messages de diffusion envoyés et reçus
  - Les interfaces réseau qui se désactivent et s'activent
  - …

MIB: Management Information base

- Base de données maintenue par l'agent qui contient les informations de gestion de réseau (éléments de réseau et leurs attributs)
  - Ensemble d'objets structurés de manière arborescente.
  - Branches = catégorie logique
  - Feuilles = informations sur les objets
- La MIB est structurée par SMI (Structure of Management Information) et est décrite en ASN-1 (Abstract Syntax Notation One)
- Chaque équipement supervisé possède sa propre MIB.

L 'architecture de SNMP

- SNMP est désigné pour être exécuté au dessus du protocole UDP (User Datagram Protocol)



Les opérations SNMPv1

- SNMP offre 3 opérations simples :
  - GET :

    Permet à la station d'administration de retirer les valeurs d'un objet de la station administrée.
  - SET :

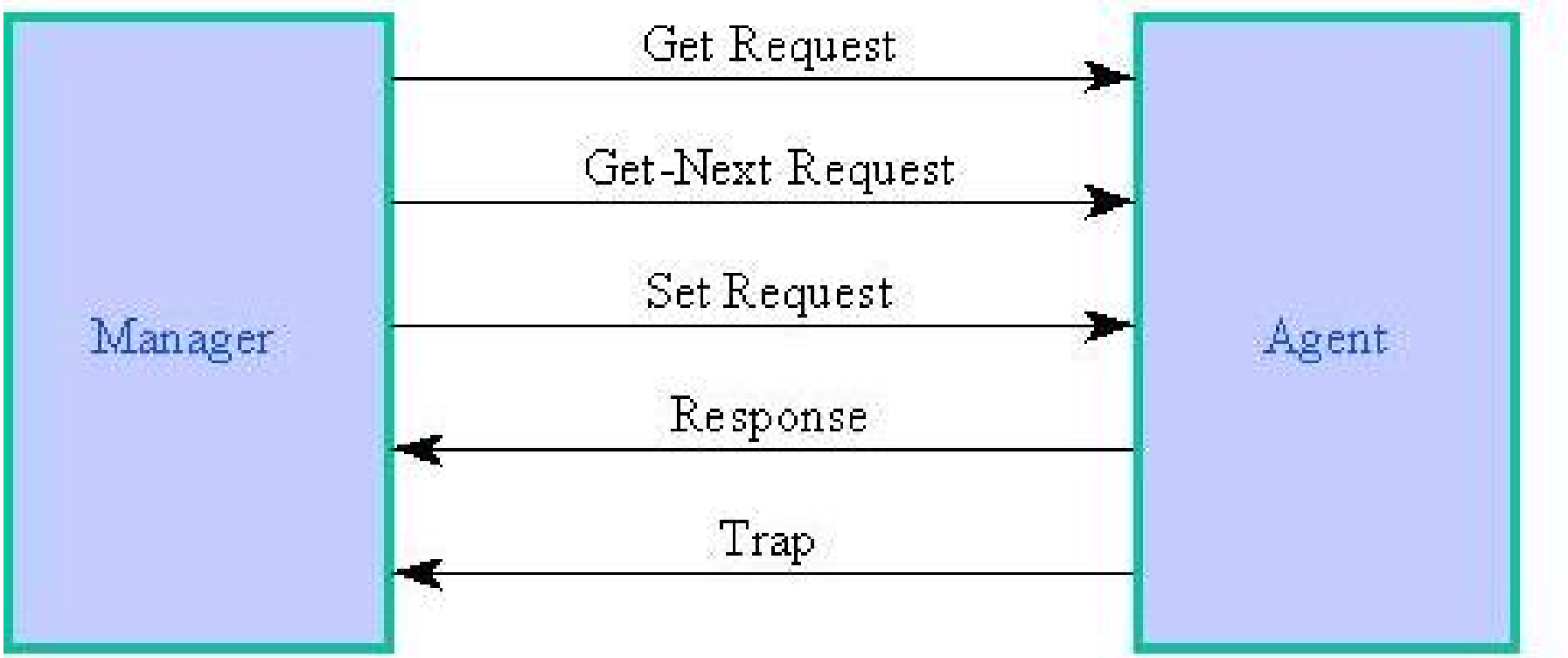
    Permet à la station d'administration d'affecter des valeurs à un objet dans la station administrée.
  - TRAP :

    Permet à une station administrée d'envoyer des notifications à la station d'administration pour les événements significatifs.
- SNMP permet un multiple accès avec une simple opération.

Les PDUs (Protocol Data Units) SNMP

- **Get Request** :
  - Utilisée pour obtenir de l'agent les valeurs des objets
- **Get-Next Request** :
  - Resemble à Get Request, mais il permet de retirer la valeur de l'objet suivant dans la MIB.
- **Set Request** :
  - Utilisée pour affecter une valeur à un objet
- **Response** :
  - Retourne une réponse à Get Request, Get-Next Request et Set Request PDUs
- **Trap** :
  - Une notification envoyée par l'agent suite à un événement.

Direction des SNMPv1 PDUs



SMI (Structure of Management Information)

- Donne les règles de définition, d'accès et d'ajout des objets dans la MIB (méta-modèle)
- **Objectifs** : encourager la simplicité et l'extension de la base d 'informations d 'Administration :
  - rendre un objet accessible de la même manière sur chaque entité du réseau
  - posséder une représentation identique des objets
- La MIB contient des éléments simples (scalaire et tableaux à deux dimensions de scalaires)
- SNMP ne permet que des interrogations de scalaires

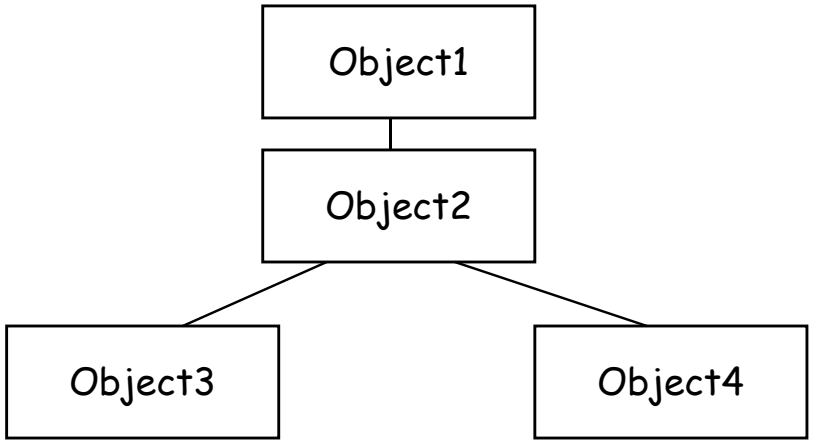
  ≠ OSI permet des structures et des modes de recherche complexes

MIB / SMI

- La SMI définit les types de données pouvant être utilisés pour stocker un objet, la manière dont ces objets sont nommés et celle dont ils sont cryptés pour être transmis sur un réseau.
- Chaque objet :
  - un nom (OID): Suite d'entiers séparés par des points
    - Préfixe Internet géré par IANA : .1.3.6.1
    - Nom symbolique décrit dans SMI & MIB: .iso.org.dod.internet.
  - une syntaxe SMI (ASN-1) qui définit
    - son type
    - son codage
  - un droit d'accès read/write

La structure des informations d'Administration (SMI)

- Un objet peut agréger plusieurs objets :

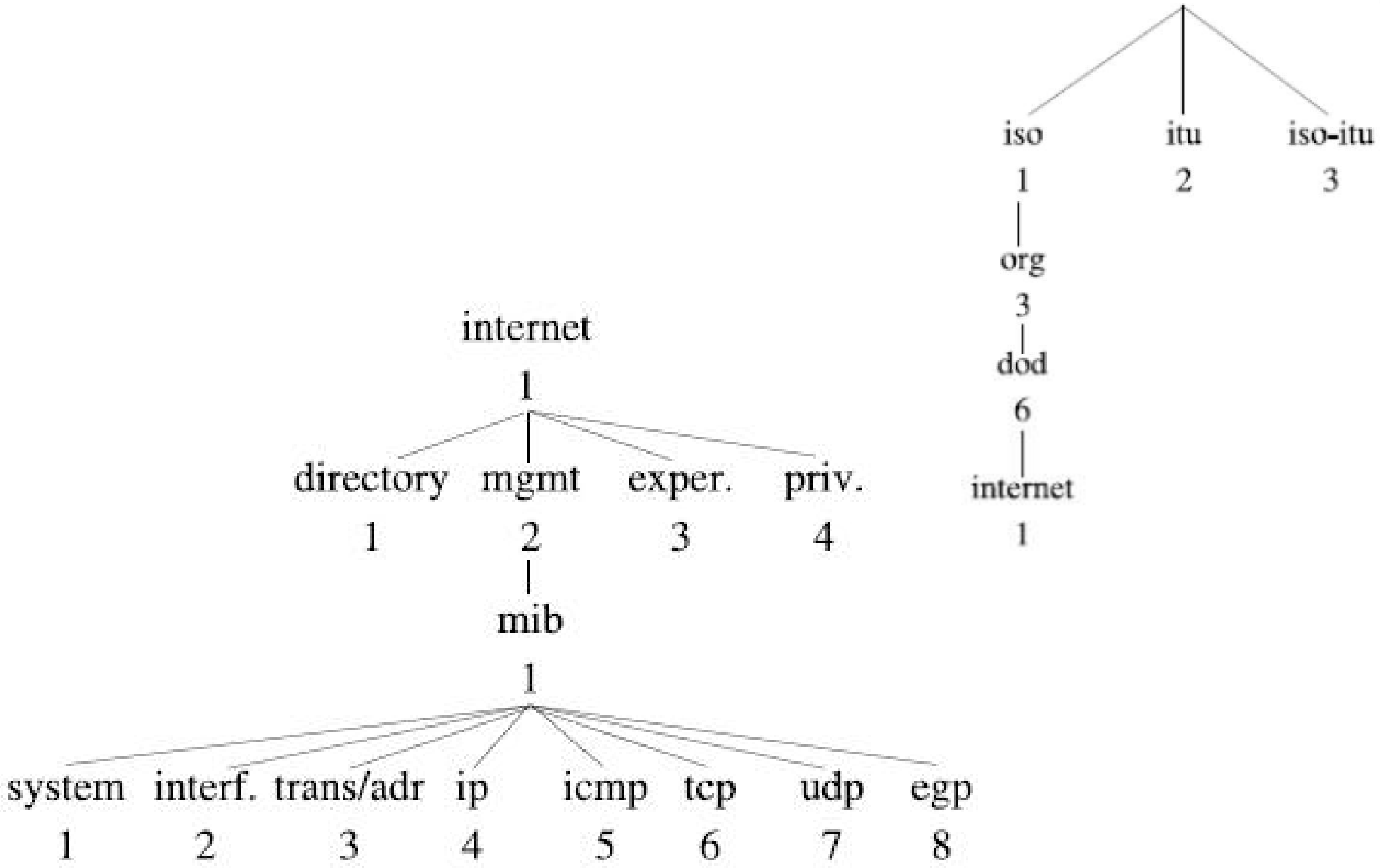


object3 OBJECT IDENTIFIER {object2 1}

object4 OBJECT IDENTIFIER {object2 2}

object2 OBJECT IDENTIFIER {object1 1}

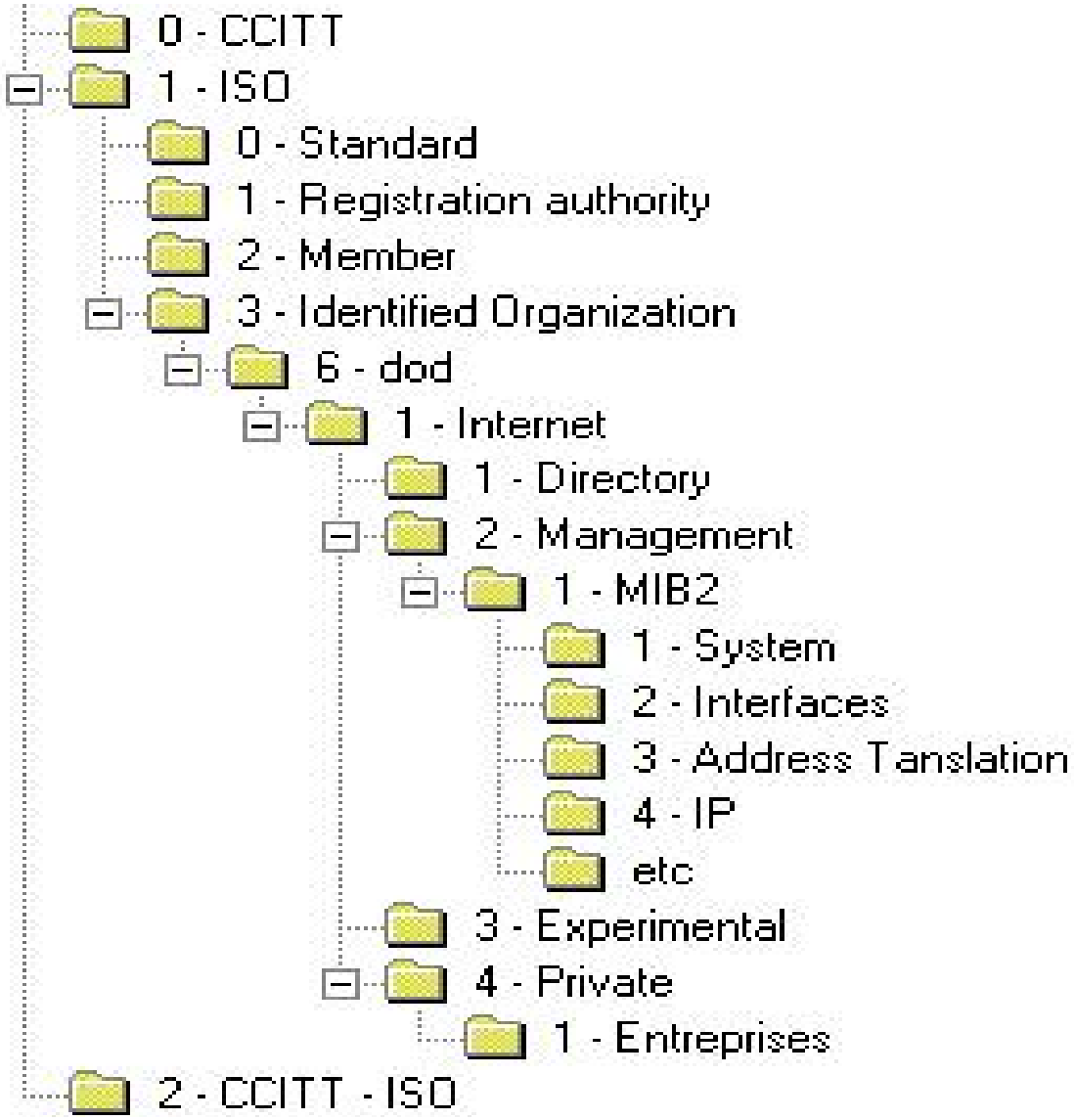
Hiérarchie d'enregistrement



La spécification de l'arbre des MIB accessibles

- En commençant de la racine, il existe trois nœuds : ccitt, iso et join-osi-ccitt.
- Sous iso, il existe les autres organisation (org) qui contient l 'objet qui représente le département de défense américaine (dod).
- Une seule sous- arborescence de dod sera réservée pour l 'administration des activités de l 'Internet (IAB)
- directory : réservée pour une utilisation future avec l 'OSI directory
- mgmt : est une sous-arborescence pour les objets définis dans le document approuvé de l 'IAB.
- Experimental : est une sous-arborescence utilisée pour identifier les objets utilisés dans l 'expérimentation de l 'Internet.
- Private : est une sous-arborescence utilisée pour identifier les objets définis d 'une façon unilatérale.

La spécification de l'arbre des MIB accessibles

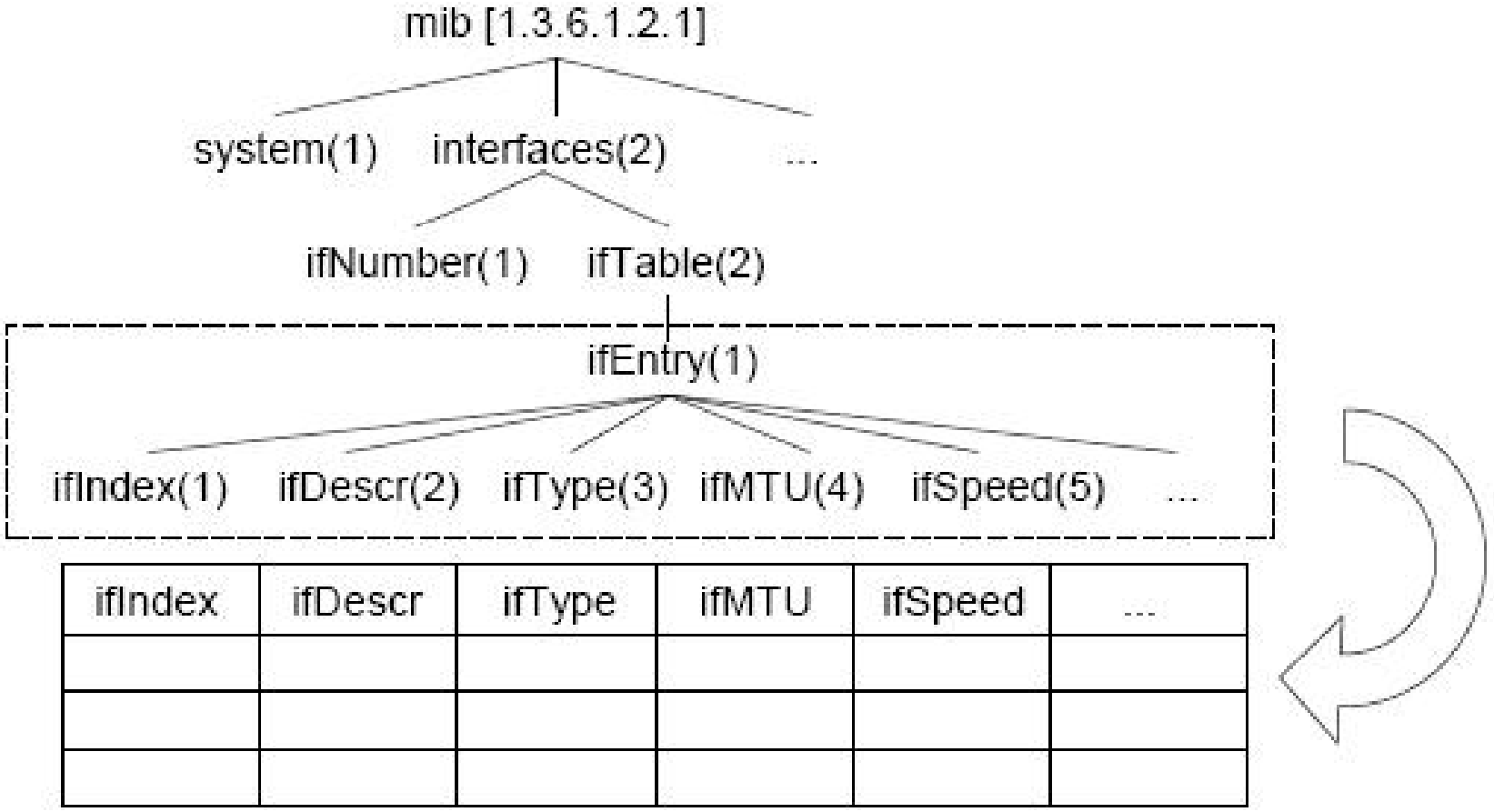


La MIB (Management Information Base)

- Modèle de données associé à SNMP:
- Base de données contenant les informations sur les éléments du réseau à gérer
  - MIB = Collection structurée d'objets
  - chaque nœud dans le système doit maintenir une MIB qui reflète l'état des ressources gérées
  - une entité d'administration peut accéder aux ressources du nœud en lisant les valeurs de l'objet ou en les modifiant.
- 2 objectifs :
  - Un schéma commun : SMI (Structure of Management Information)
  - Une définition commune des objets et de leur structure

Hiérarchie d'enregistrement: exemple

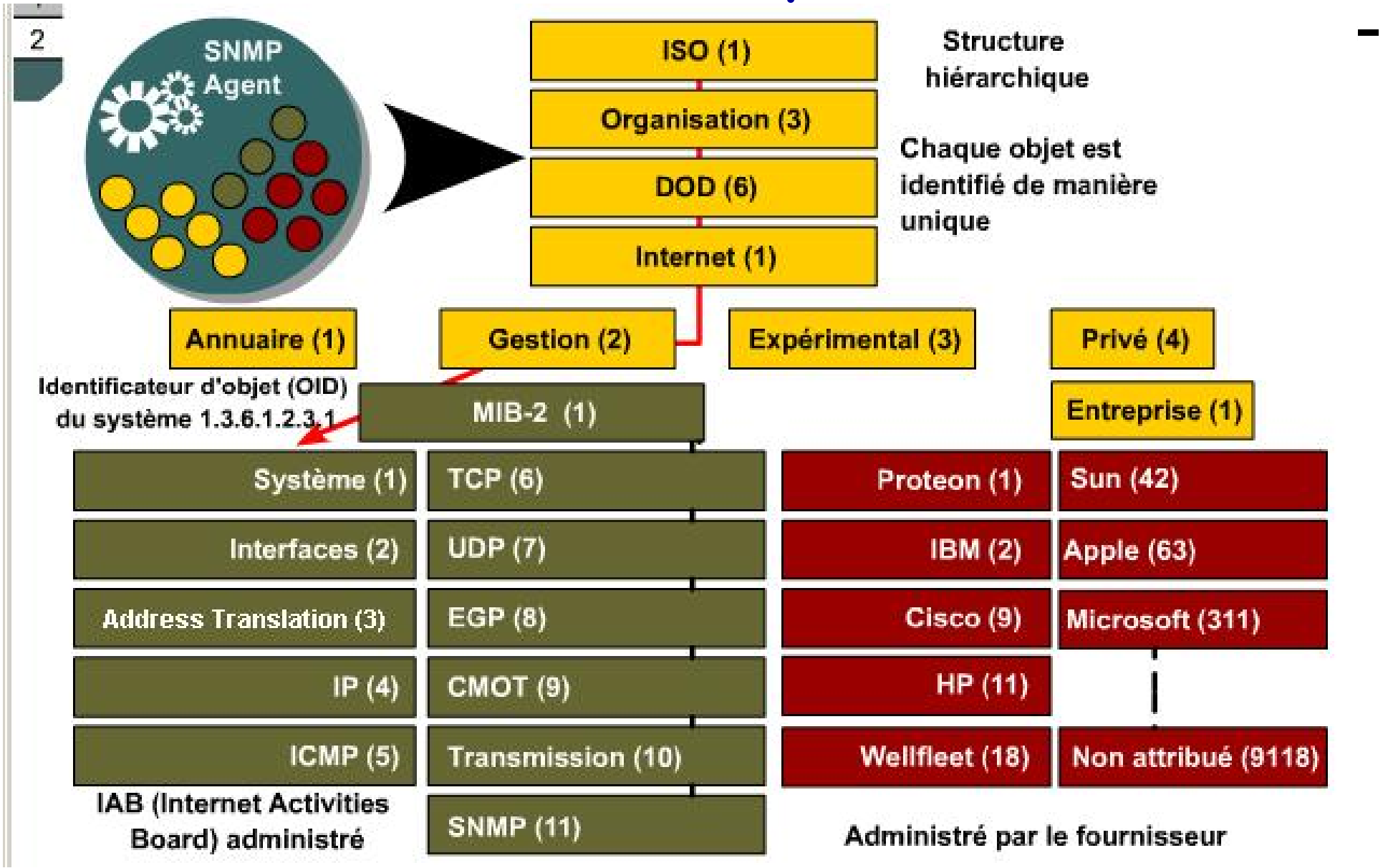
- Table des interface: ifTable [1.3.6.1.2.1.2.2]
  - Décrit toues les interfaces présentes sur le nœud



Hierarchie d'enregistrement: exemple

- **ifDescr**: Description de l 'interface
  ifType: Identification du type d 'interface
  IfMtu: Taille max en octets des datagrammes IP
  ifSpeed: débit de l 'interface en bits/s
  ifPhysAddress: adresse physique de l 'interface
  ifAdminStatus: état administratif de l 'interface (up/down)
  ifOperStatus: état opérationnel de l 'interface (up/down)
  ifLastChange: date du dernier passage de l 'interf à l 'état opérationnel
  ifInOctets: nb total d 'octets reçus sur l 'interface
  ifInUcastPkts: nb de paquets unicast transmis au niveau supérieur
  ifInNUcastPkts: idem pour les paquets broadcast et multicast
  ifInDiscards: nb de paquets reçus et volontairement détruits
  ifInErrors: nb de paquets reçus contenant des erreurs
  ifInUnknownProtos: nb de paquets détruits car d 'un protocole inconnu
  ifOutOctets: nb total d 'octets transmis sur l 'interface
  ifOutUcastPkts: nb de paquets unicast reçus du niveau supérieur

Hierarchie d'enregistrement: exemple



La structure des informations d 'Administration (SMI)

- Les objets administrables sont une abstraction des ressources physiques (interfaces, équipements,etc.) et logiques (connexion TCP, paquets IP, etc.)
- Ils possèdent :
  - un nom (Descripteur + identificateur d 'objet)
  - une syntaxe utilisant ASN.1 (Abstract Syntaxe Notation)
  - une définition qui est un texte de description de l 'objet
  - un accès qui spécifie les droits d 'accès à l 'objet (read only, read-write or not accessibe)
  - status qui spécifie si l 'objet est courant (mondatory ou optional) ou obsolète.
  - un schéma de codage BER (Basic Encoding Rules)

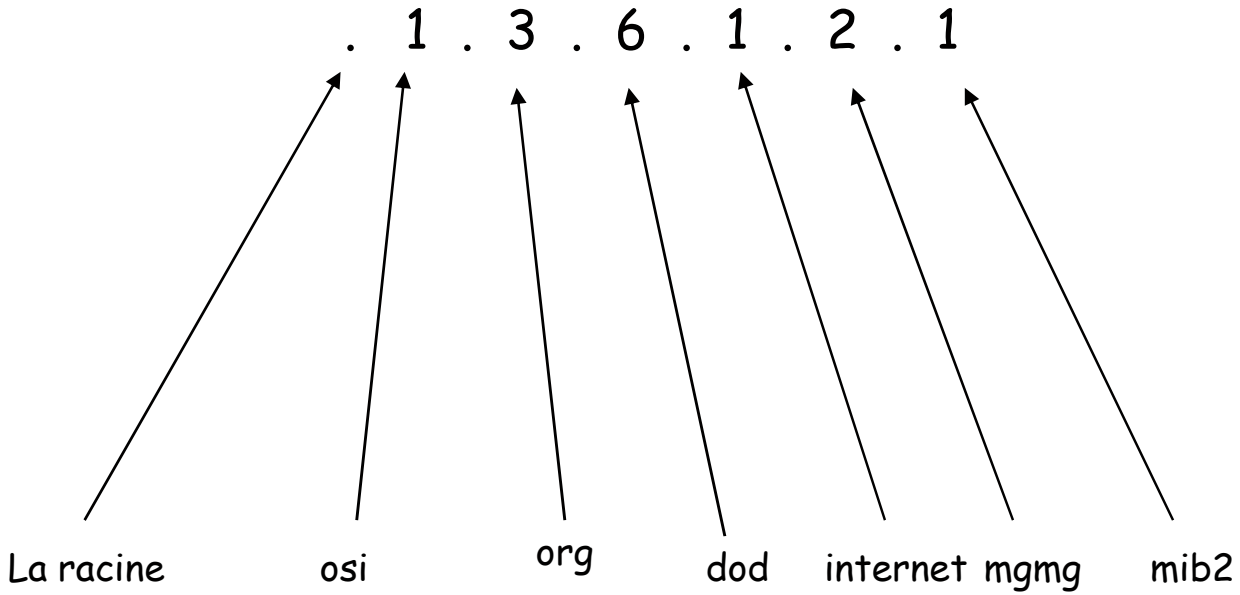
La structure des informations d'Administration (SMI)

- Les caractéristiques d'un objet sont regroupées dans la définition d'une macro qui définie la structure d'un type d'objet :

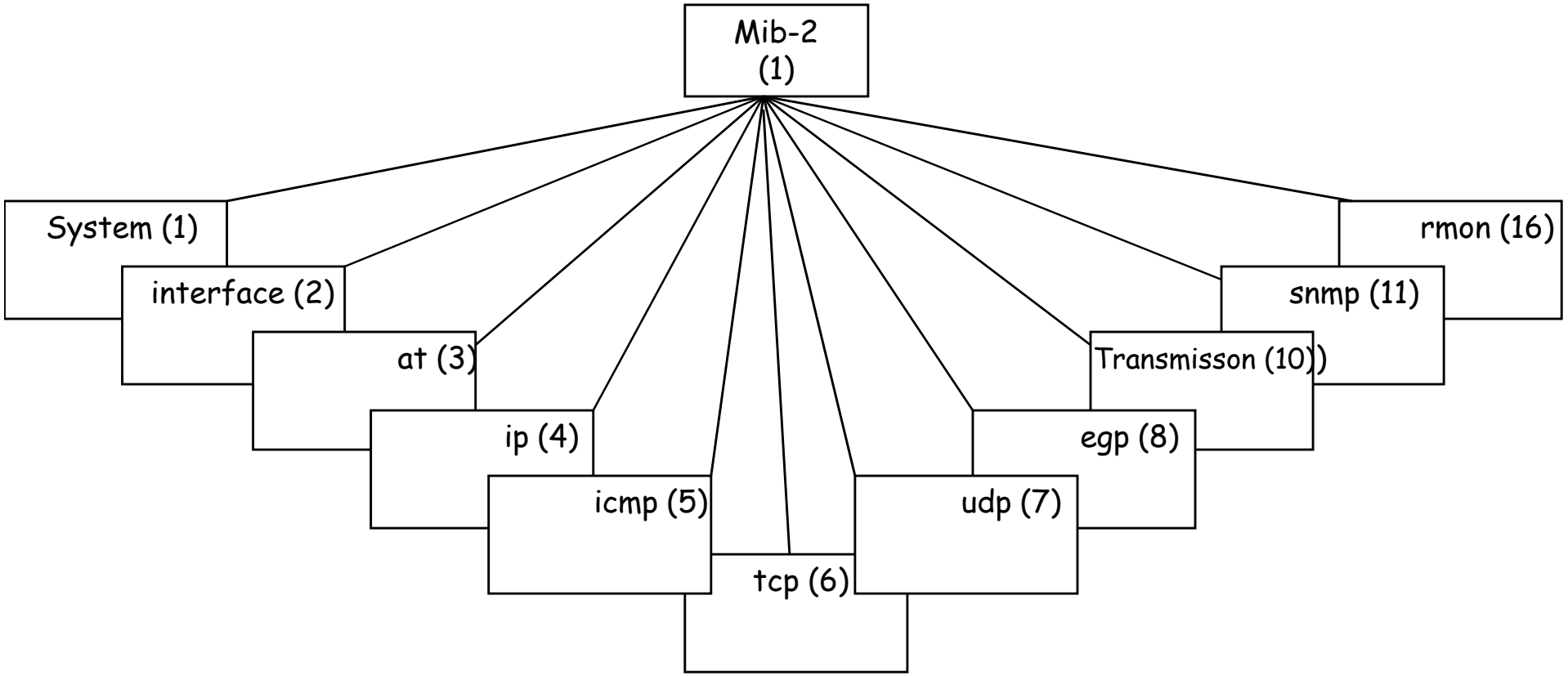
```
OBJECT-TYPE MACRO ::=
BEGIN
    TYPE NOTATION ::=
        "SYNTAX" type (TYPE ObjectSyntax)
        "ACCESS" Access
        "STATUS" Status
    VALUE NOTATION ::= value (VALUE ObjectName)
    Access ::= "read-only"
        |"read-write"
        |"write-only"
        |"not-accessible"
    Status ::= "mandatory"
        |"optional"
        |"obsolete"
        |"deprecated"
    END
```

La MIB 2

- Un identificateur d 'un objet est un identificateur unique qui consiste en une séquence d 'entiers dont chaque entier représente la position de ces successeurs dans l 'arbre.
- Exemple : l 'identificateur de l 'objet MIB2 :



Le groupe MIB-2



Le groupe MIB-2

MIB-2

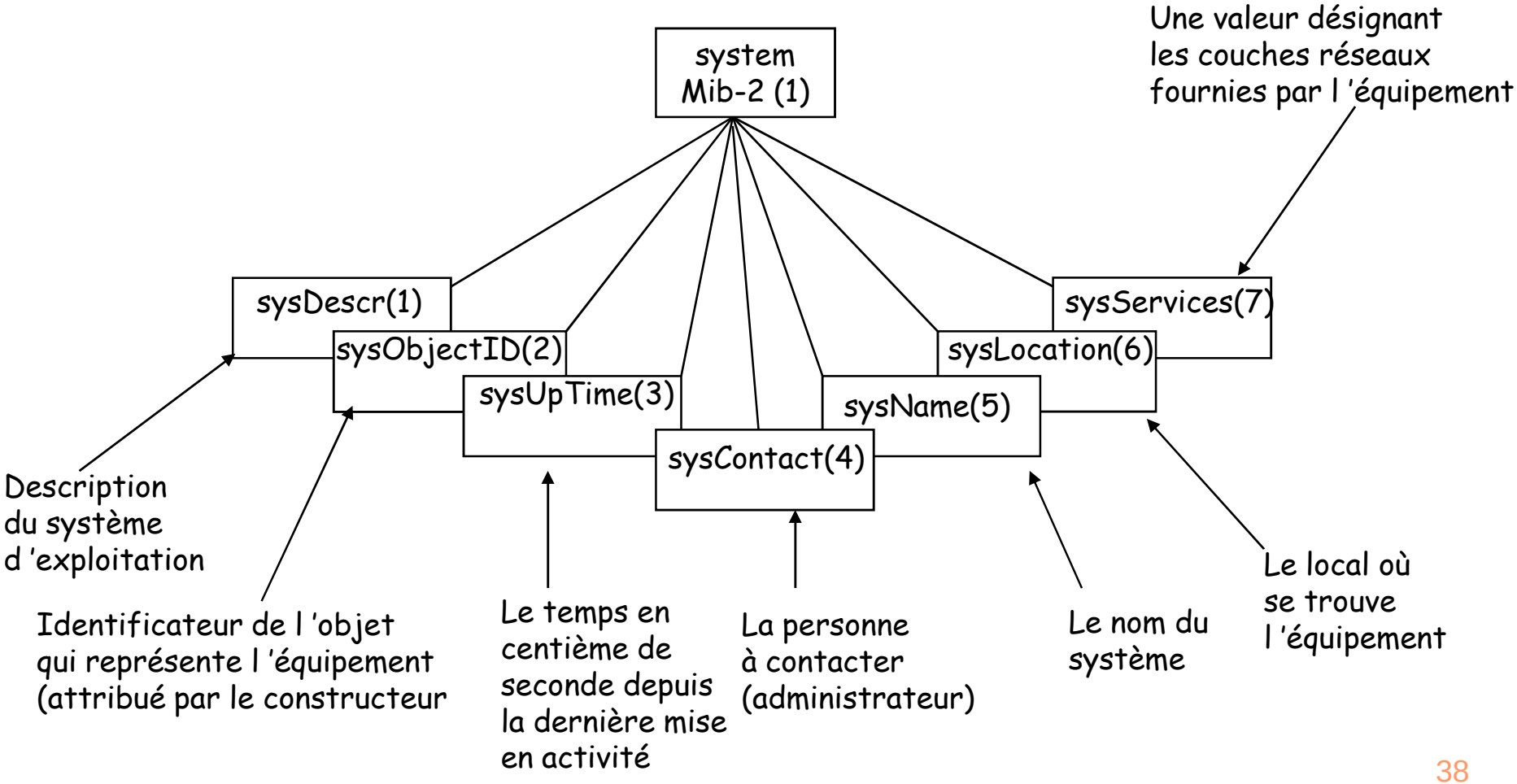
group	nbre éléments	commentaire
system	7	nœud dans le réseau
interfaces	25	interfaces réseau
at	5	IP address translation
ip	65	Internet Protocol
icmp	26	Internet Control Message Protocol
tcp	21	Transmission Control Protocol
udp	8	User Datagram Protocol
egp	22	Exterior Gateway Protocol
transmission	114	information sur la transmission
snmp	28	SNMP
rmon	218	Remote network monitoring

La structure numérique de la MIB-2

system	1.3.6.1.2.1.1
interfaces	1.3.6.1.2.1.2
at	1.3.6.1.2.1.3
ip	1.3.6.1.2.1.4
icmp	1.3.6.1.2.1.5
tcp	1.3.6.1.2.1.6
udp	1.3.6.1.2.1.7
egp	1.3.6.1.2.1.8
cmot	1.3.6.1.2.1.9
transmission	1.3.6.1.2.1.10
snmp	1.3.6.1.2.1.11

Le groupe « System »

- **system** : correspond au nom de l'agent, num de version, type de la machine, nom du système d'exploitation, etc.



Le groupe « System »

exemple d'interrogation : Accès à des variables d'administration

sur la machine gam :

sysDescr[0]= "Linux OS", sysObjectID[0]=1.3.6.1.4.1.464.1, sysUpTime[0]=449144

sysContact[0]= "Mohamed Jarraya", sysName[0]= "gam.isi.rnu.tn",
sysLocation[0]= "Pièce43", sysServices[0]=79

sysServices = $\sum_{l \in L} 2^{L-1}$, où L est le numéro de la couche qui implémente les services et S l 'ensemble de couches qui fournissent les services.

Exemple : sysServices = 79 = $2^0 + 2^1 + 2^2 + 2^3 + 2^6$, signifie que l 'équipement implémente les services pour les couches 1, 2, 3, 4, 7.

Couche	Fonctionnalité	Couche	Fonctionnalité
1	Physique	2	Liaison de données
3	Internet	4	transport
7	Application		

Le groupe « Interface »

ifNumber : le nombre d 'interfaces

ifIndex : Index de l 'interface (son numéro)

ifDescre : Description de l 'interface

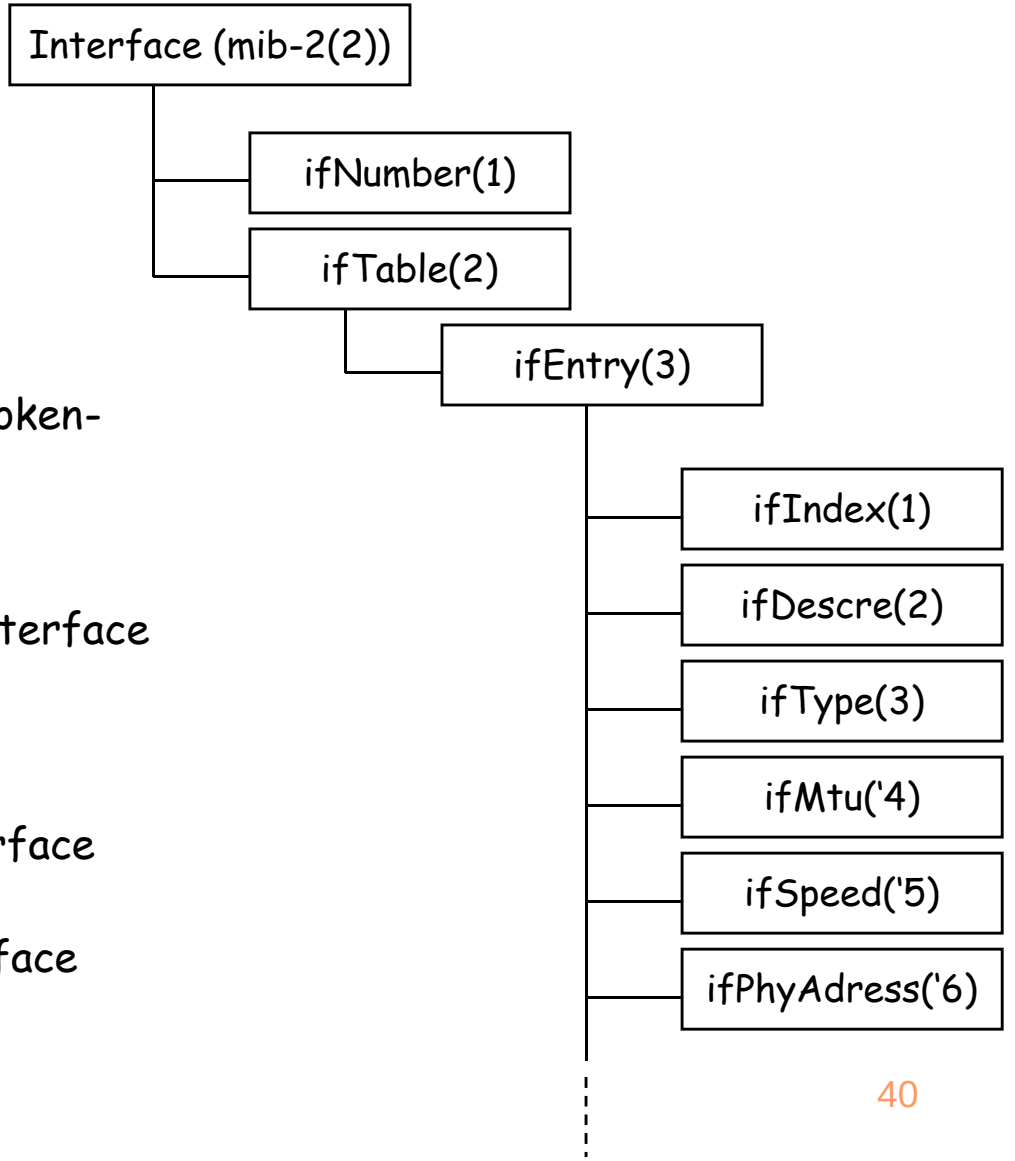
ifType : le type de l 'interface (Ethernet, Token-Ring,...)

ifMtu : le nombre maximum d 'octet que l 'interface peut envoyer ou recevoir

ifSpped : Une estimation du débit de l 'interface

ifPhyAdress : l 'adresse physique de l 'interface

....



Le groupe « IP »

ipForwarding : Agit comme passerelle, ou non

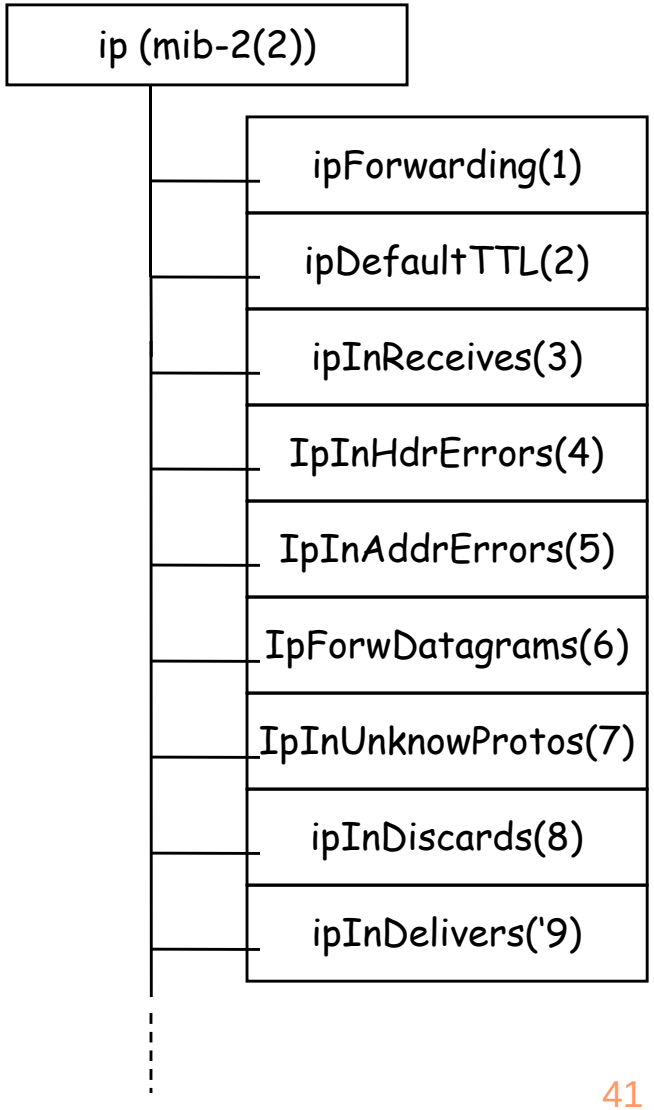
ipDefault TTL : la valeur par défaut du TTL ajouté dans un paquet IP

ipInReceives : Le nombre total de paquets IP reçus

IpInHdrErrors : Le nombre total de paquets écartés dus à une erreur sur l 'en-tête

IpInAddrErrors : Le nombre total de paquets écartés dus à une erreur sur l 'adresse de destination

IpForwDatagrams : Le nombre total de paquets dont l 'entité réceptrice ne représente pas la destination finale.



Les autres groupes

icmp : 26 compteurs

- pour chaque message icmp, 2 compteurs pour compter les messages reçus et émis
- 4 compteurs pour compter le nombre total de messages icmp reçus, reçus par erreur ou non envoyés,

tcp : rend compte des connexions TCP en cours et leurs paramètres

de type nombre max de connexions simultanées permises, nombre d'ouvertures actives, l'état de chaque connexion (écoute, time-wait,...).

udp : - 4 compteurs renseignent sur le nombre de datagramme

UDP envoyés, reçus, en erreur, ...

egp : gère le protocole egp (External gateway protocol)(routage

des paquets entre routeurs). On a le nbre de paquets entrants, sortants, en erreur, la table des routeurs adjacents, des infos sur les routeurs...

snmp : requis pour chaque entité mettant en oeuvre le protocole

SNMP. Contient le nombre de messages SNMP entrants et sortants, le nombre de mauvaises versions reçues ou de nom de communauté invalide, la répartition du type de requêtes reçues et envoyées (get, get_next, set et trap)

Exemples de définition d 'objets

Exemple d'objets défini par le SMI du RFC1155

```
tcpConnTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF TCPConnEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "A table containing TCP connection-sepecific information"
    ::={tcp 13}
    tcpConnEntry OBJECT-TYPE
    SYNTAX      TcpConnEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "Information about current TCP connection,..."
    INDEX {tcpConnLocalAdress, tcpConnLocalPort, tcpConnRemAdress,
    tcpConnRemPort}
    TcpConnEntry::=SEQUENCE{tcpConnState       INTEGER,
    tcpConnLocalAdress  IpAdress, tcpConnLocalPort INTEGER (0..65535),
    tcpConnRemAdress IpAdress, tcpConnRemPort INTEGER(0..65535)
```

Exemples de définition d 'objets

```
TcpConnState OBJECT-TYPE
    SYNTAX          INTEGER{ closed(1), listen(2), synSent(3), synReceive(4),
                                established(5), finWait1(6), finWait2(7), closeWait(8),
                    lastAck(9), closing(10), timeWait(11), deleteTCB(12)}
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION "the state of this TCP connection..."
    ::={tcpConnEntry 1}
TcpConnLocalAdress OBJECT-TYPE
    SYNTAX          IpAdress
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION  "the local port IP adress for this TCP connection... "
    ::={tcpConnEntry 1}
TcpConnLocalPort OBJECT-TYPE
    SYNTAX          INTEGER(0..65535)
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION  "the local port number for this TCP connection ..."
    ::={tcpConnEntry 1}
....
```

Les mécanismes du protocole SNMP

- Le protocole SNMP implémente 3 mécanismes :
  - L 'authentification,
  - l 'autorisation (politique d 'accès)
  - L 'identification de l 'objet
- L 'authentification se fait par le choix d 'un nom de communauté afin de restreindre l 'accès aux agents que par les administrateurs réseaux.
  - Le nom de communauté est vérifié pour chaque requête SNMP.
  - Il est relié au mode d 'accès aux objets de la MIB (lecture-écriture).
- Chaque communauté définie un mode d 'accès qui peut être soit Read-only, soit read-write.

Les mécanismes du protocole SNMP

- L 'autorisation est l 'intersection entre le mode d 'accès défini par la communauté et l 'accès à l 'objet défini parmi les caractéristiques de l 'objet.

Mode d'accès	read-only	read-write	write-only	not-accessible
read-only	3	3	1	1
read-write	3	2	4	1

où les classes sont defines par :

1 no right

2 get, get-next, set, trap

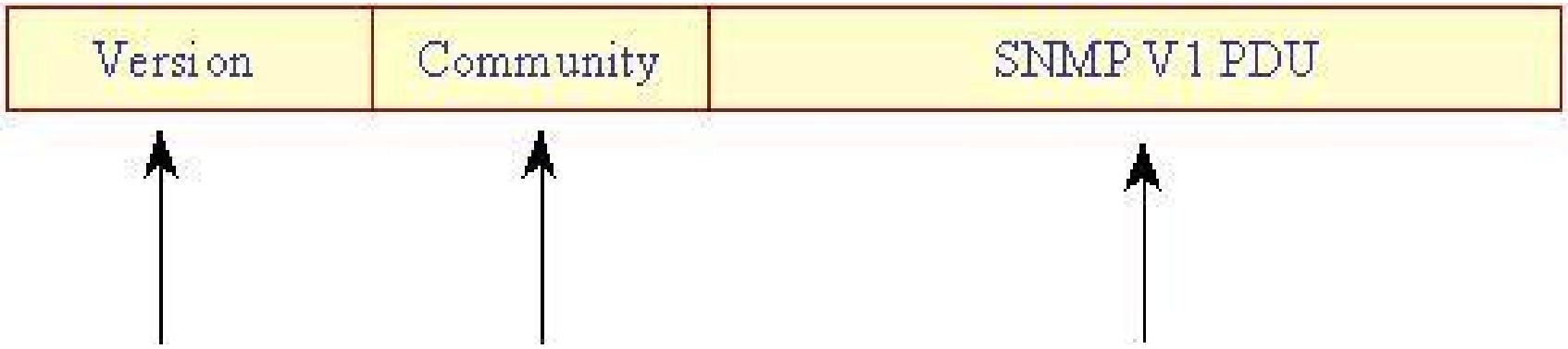
3 get, get-next, trap

4 set, trap

Les mécanismes du protocole SNMP

- L 'identification de l 'objet se fait par son OID (Object Identifier) suivi par un suffixe.
  - Le suffixe =0 pour une variable simple (non un tableau)
  - le suffixe ≠ 0 pour désigner l 'index dans le cas d 'une variable composée (un tableau)
- Chaque message SNMP (sauf les traps) contient :
  - un identificateur de requête,
  - une liste de variables (noms et valeurs)
  - éventuellement une liste d 'erreurs (tooBig, noSuchName, badValue, readOnly, etc.)
  - un indice pour les erreurs (le nombre de variables contenant des erreurs).

Format général du Message SNMP



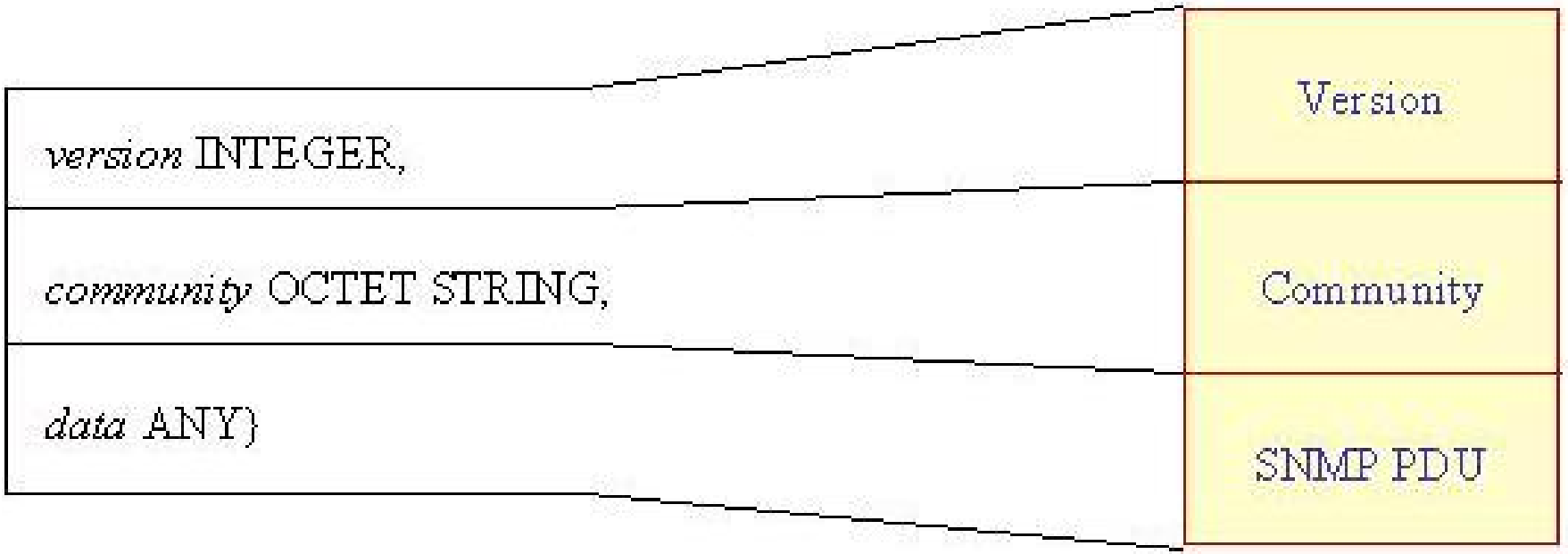
- SNMP community = Un ensemble d'administrateurs autorisés à utiliser l'agent
- Chaque communauté est utilisée en utilisant un nom unique
- Les administrateurs doivent préciser le nom de la communauté dans les requêtes SNMP

Définition ASN.1 du Message

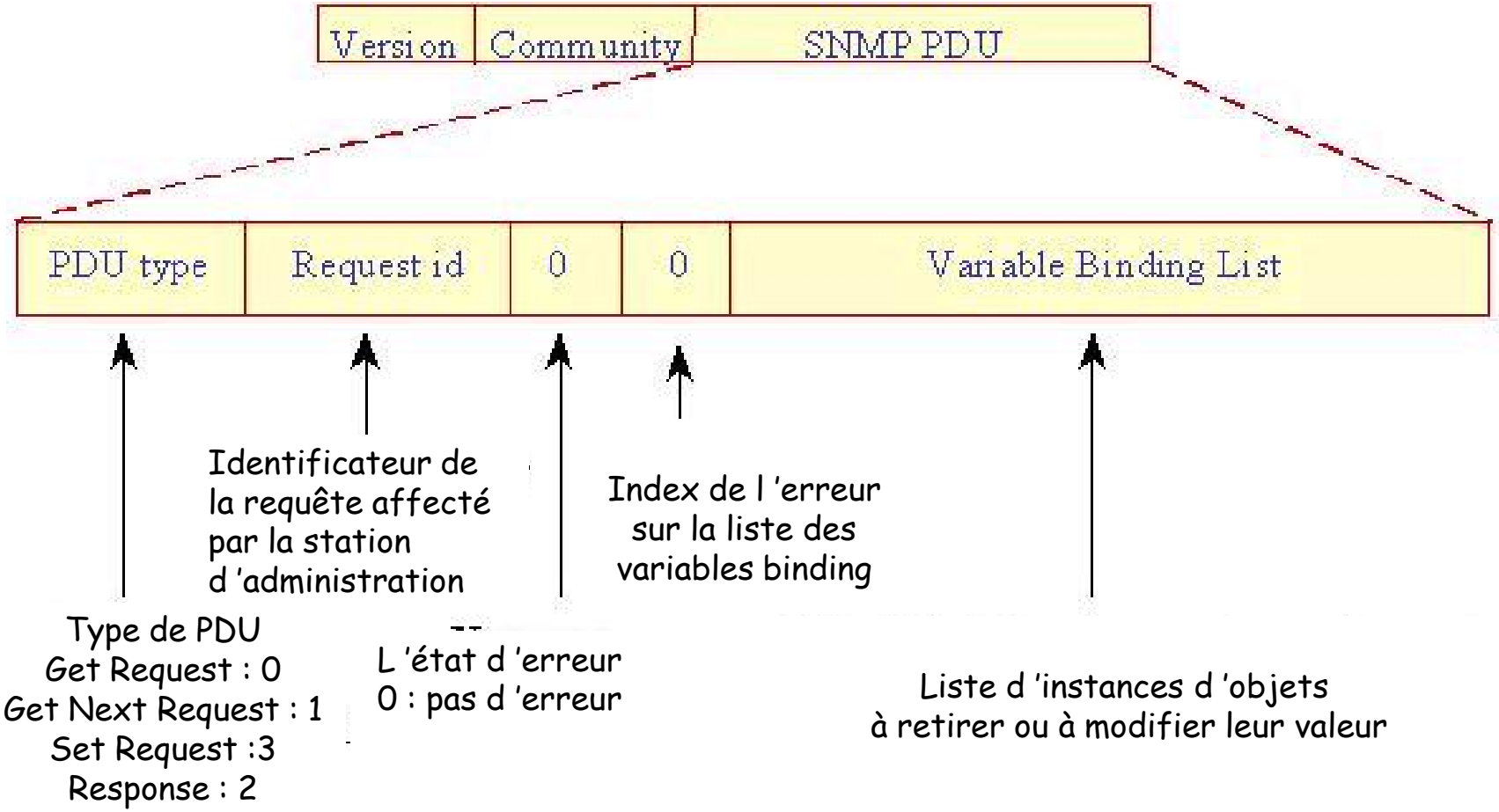
RFC1157-SNMP DEFINITIONS ::= BEGIN

IMPORTS *ObjectName*, *ObjectSyntax*, ... FROM *RFC1155-SMI*;

Message ::= SEQUENCE {



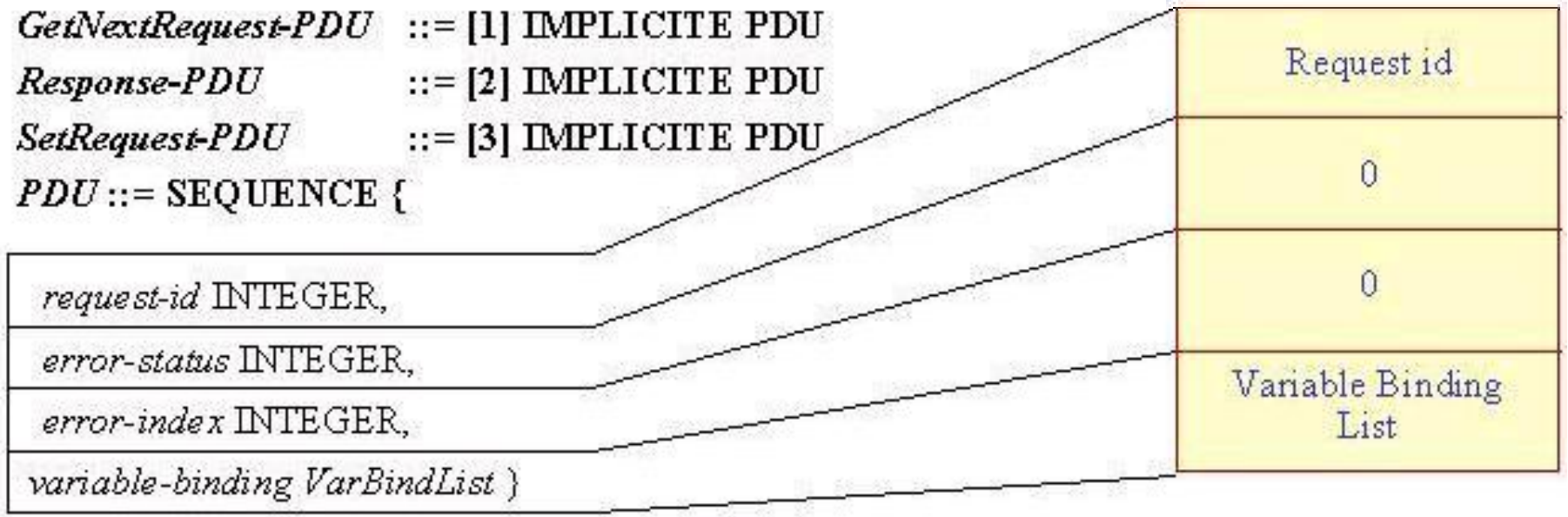
Format des Get, Get-Next et Set



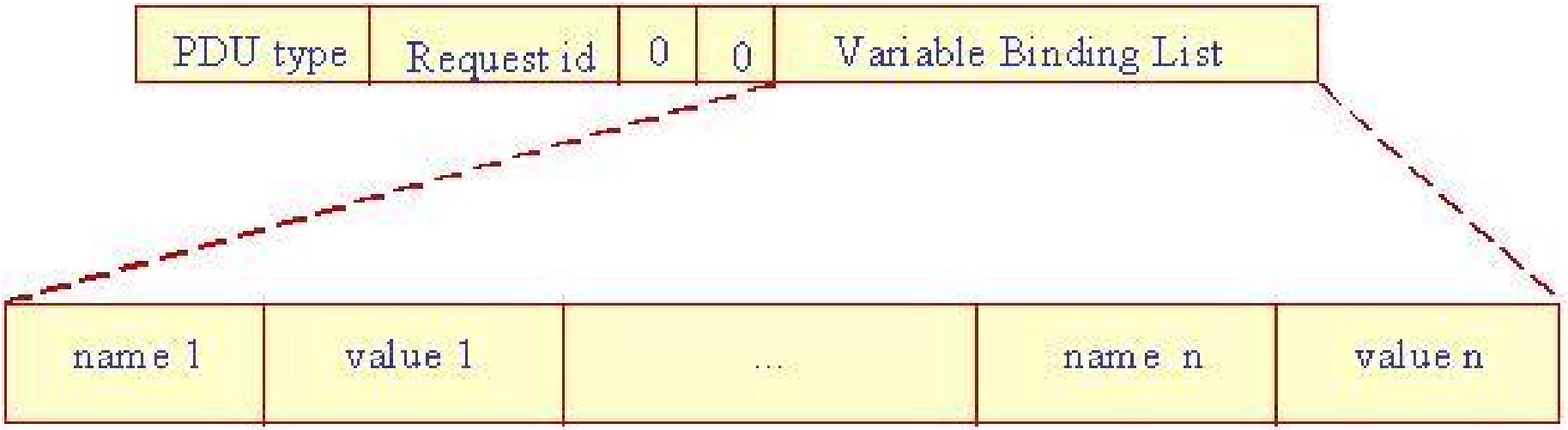
- La liste de variables binding regroupe un nombre d'opérations de même type (get, set, trap) dans la même requête.

Definitions ASN.1 des Get, Get Next and Set

```
PDUs ::= CHOICE {   get-request          GetRequest-PDU,
                    get-next-request     GetNextRequest-PDU,
                    response             Response-PDU,
                    set-request          SetRequest-PDU,
                    trap                 Trap-PDU}
GetRequest-PDU      ::= [0] IMPLICITE PDU
GetNextRequest-PDU  ::= [1] IMPLICITE PDU
Response-PDU        ::= [2] IMPLICITE PDU
SetRequest-PDU      ::= [3] IMPLICITE PDU
PDU ::= SEQUENCE {
    request-id INTEGER,
    error-status INTEGER,
    error-index INTEGER,
    variable-binding VarBindList }
```



Format de Variable Binding List



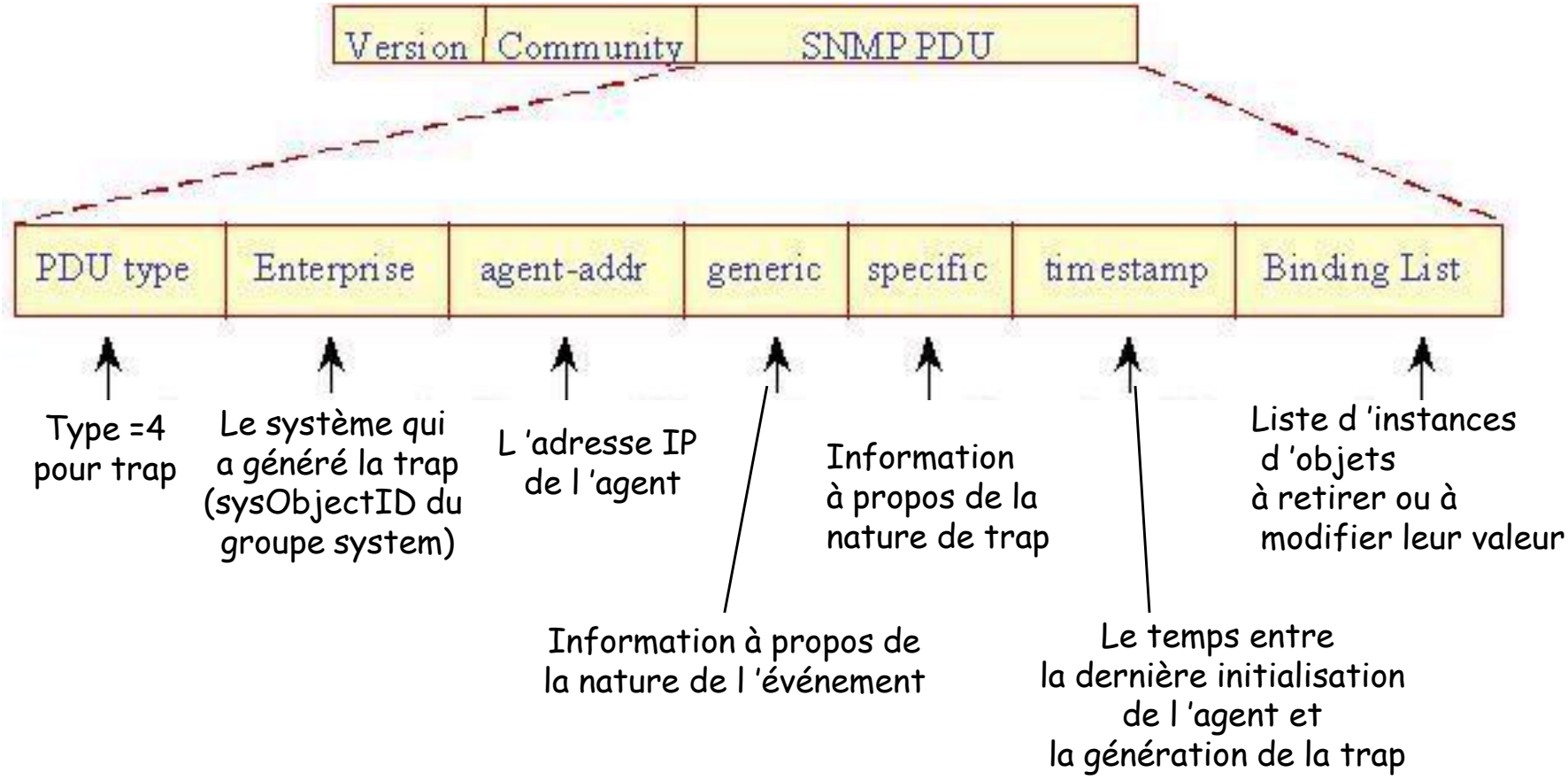
VarBind ::= **SEQUENCE** {

name ObjectName,

value ObjectSyntax }

VarBindList ::= **SEQUENCE OF** *VarBind*

Format de Trap

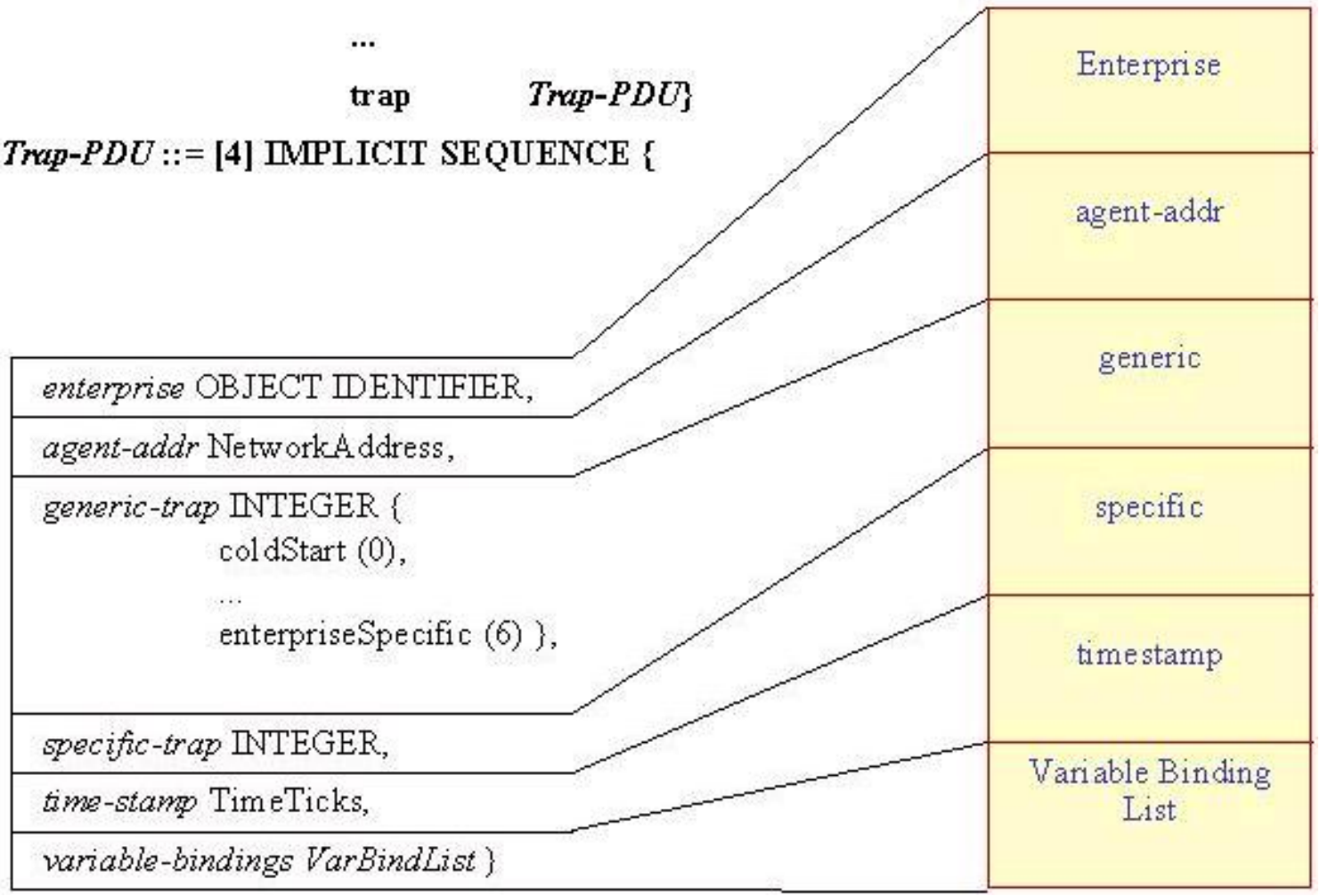


Les champs "Generic et Specific"

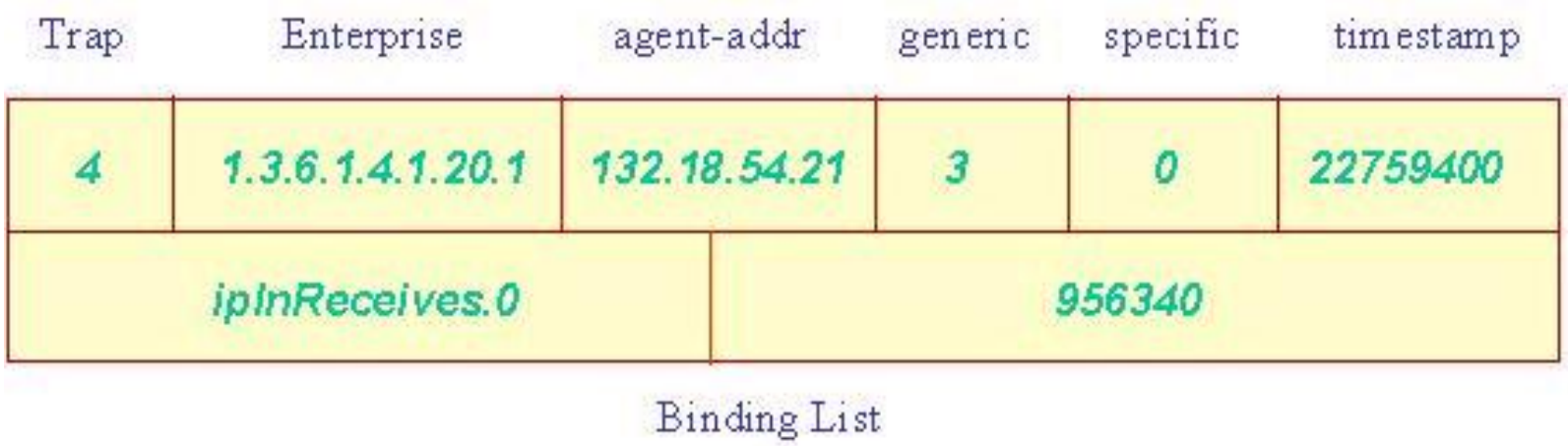
- Le champ générique peut prendre une des valeurs suivantes :
  - **coldStart (0)** : Une réinitialisation inatendue due à une défaillance.
  - **warmStart (1)** : Une défaillance mineur
  - **linkDown (2)** : Une défaillance survenue sur une interface physique.
  - **linkUp (3)** : Une interface devient active.
  - **authenticationFailure (4)** : L'agent a reçu un message avec une authentification impropre
  - **egpNeighborLoss (5)** : Un routeur voisin utilisant EGP (External Gateway Protocol) est décalrée comme étant non focntionnel
  - **enterpriseSpecific (6)** : L'événement relatif à "enterprise-specific" est servenu

Définition ASN.1 du Trap

```
PDUs ::= CHOICE {    get-request GetRequest-PDU,
                     ...
                     trap        Trap-PDU}
Trap-PDU ::= [4] IMPLICIT SEQUENCE {
    enterprise OBJECT IDENTIFIER,
    agent-addr NetworkAddress,
    generic-trap INTEGER {
                     coldStart (0),
                     ...
                     enterpriseSpecific (6) },
    specific-trap INTEGER,
    time-stamp TimeTicks,
    variable-bindings VarBindList }
```

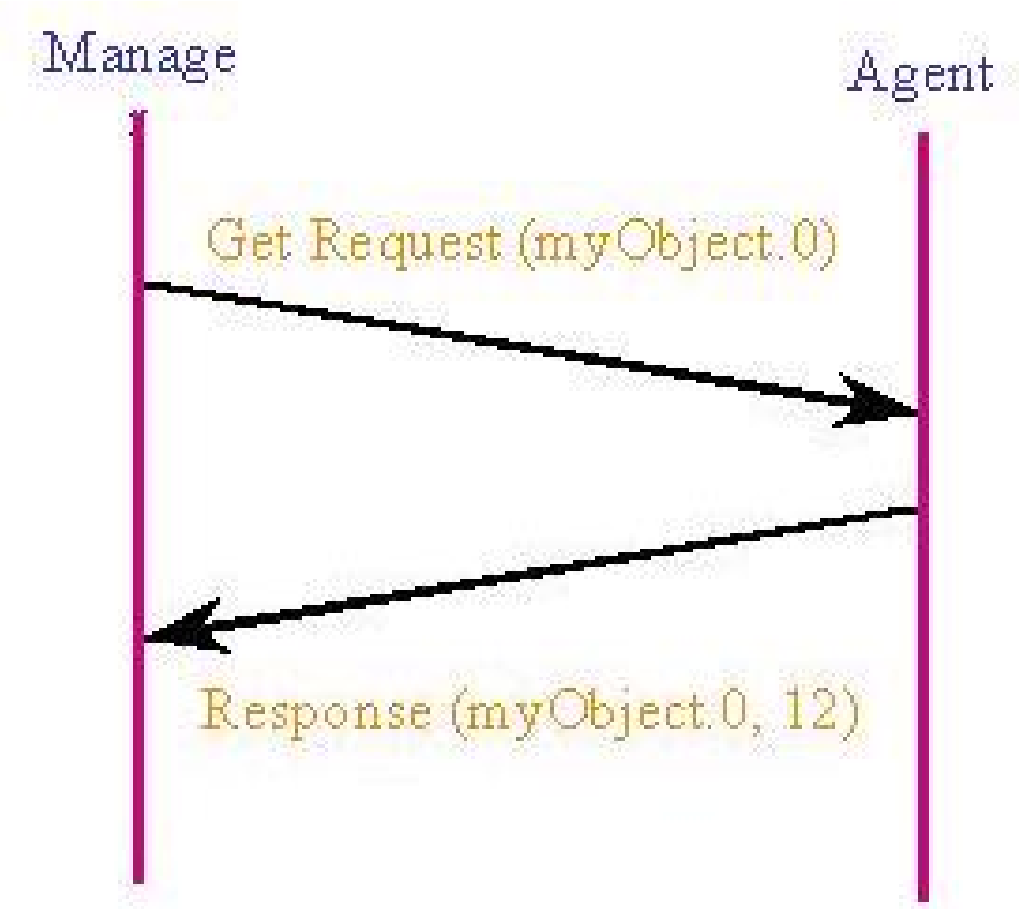


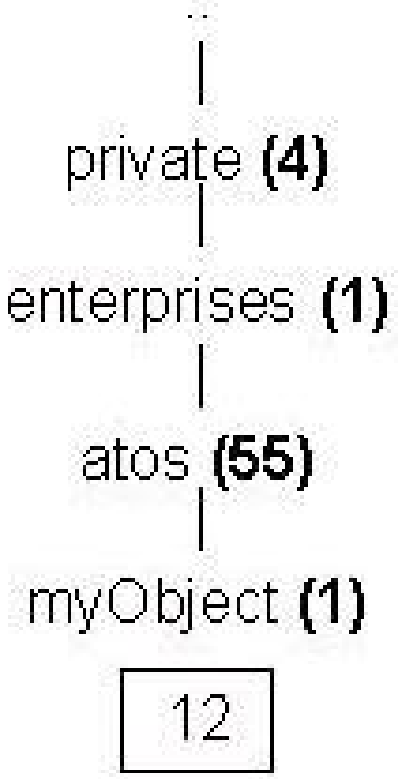
Exemple de Trap



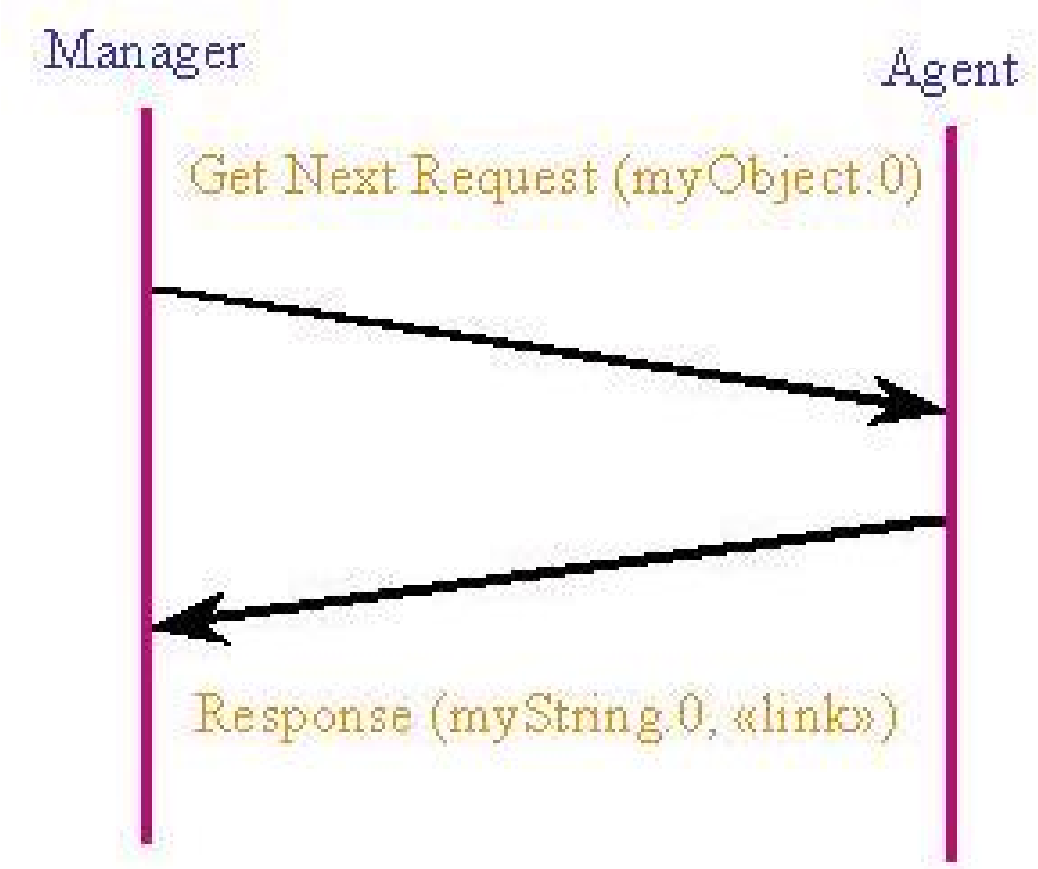
- L 'adresse IP de agent émetteur : 132.18.54.21
- L 'objet concerné par la trap est : 1.3.6.1.4.1.20.1 (MIB privée)
- Type de de trap : link up
- Indication : les nombre de paquets reçu est 956340
- La dernière réinitialisation de l'agent : 6 heures passées.

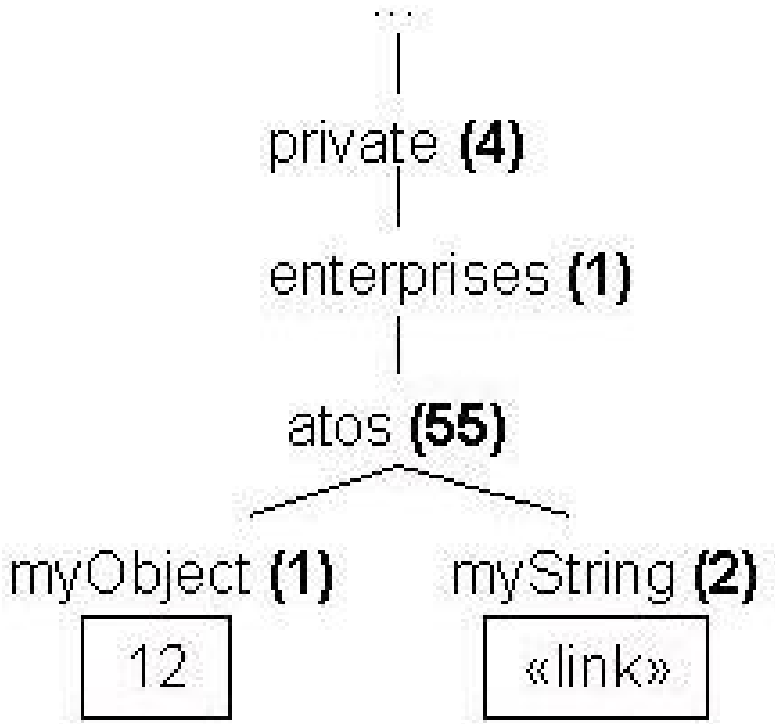
La requête GET



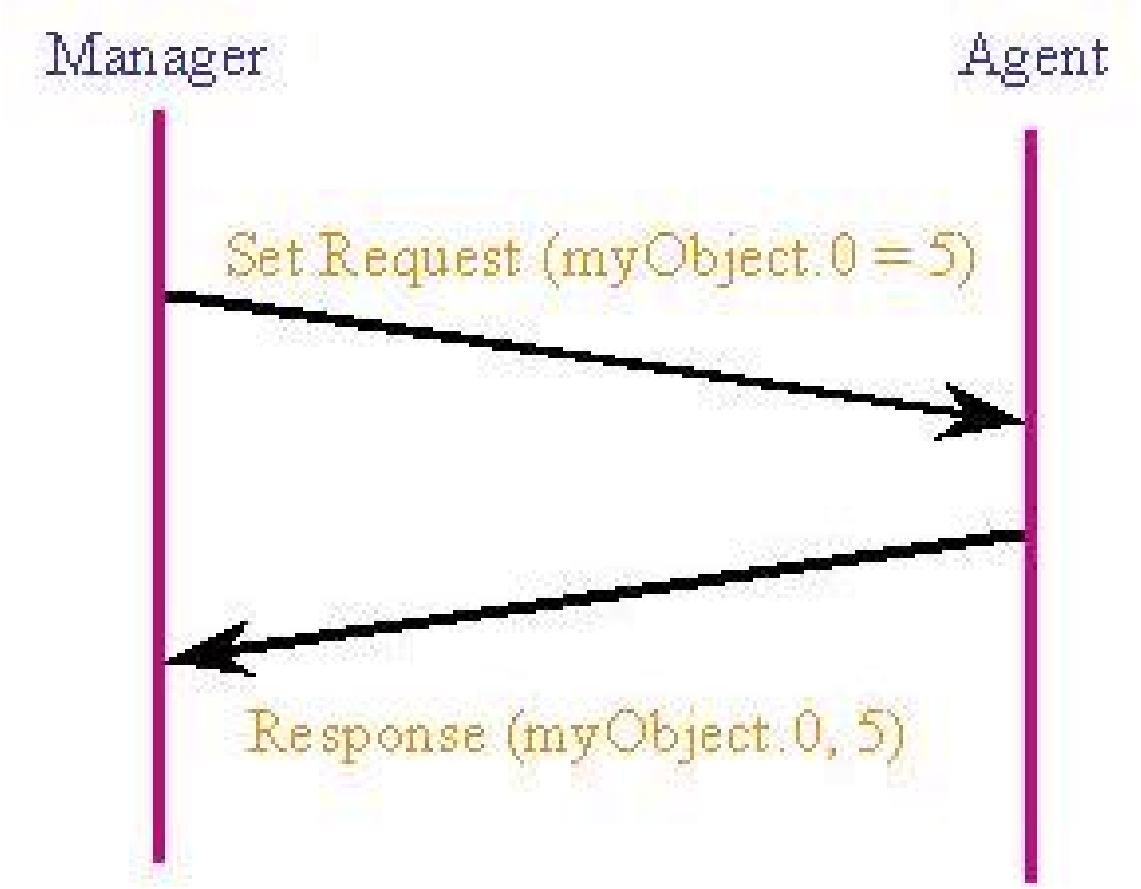


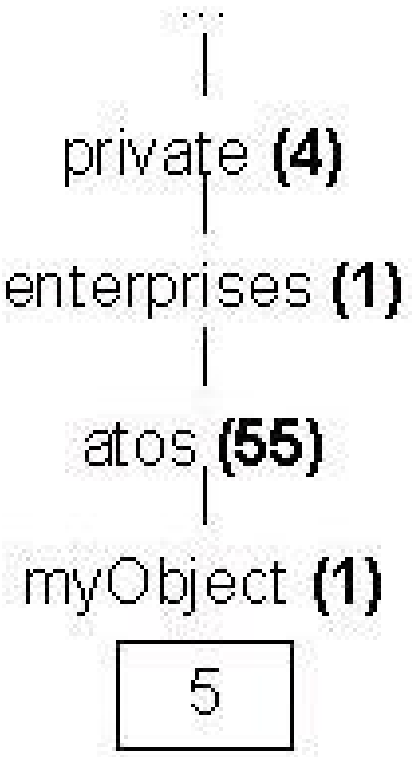
La requête GETNext



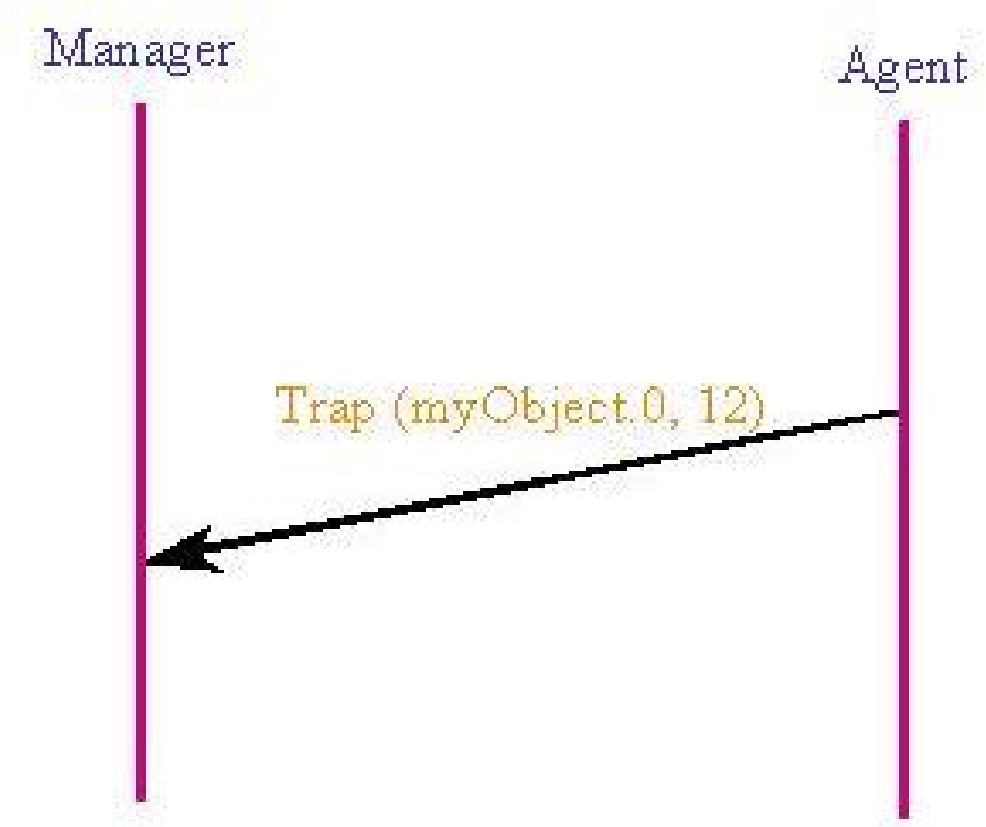


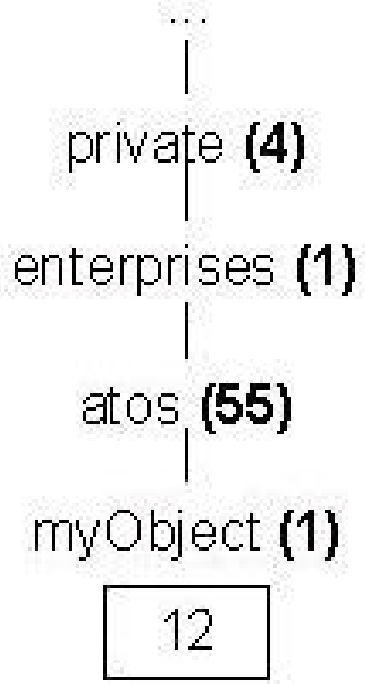
La requête Set





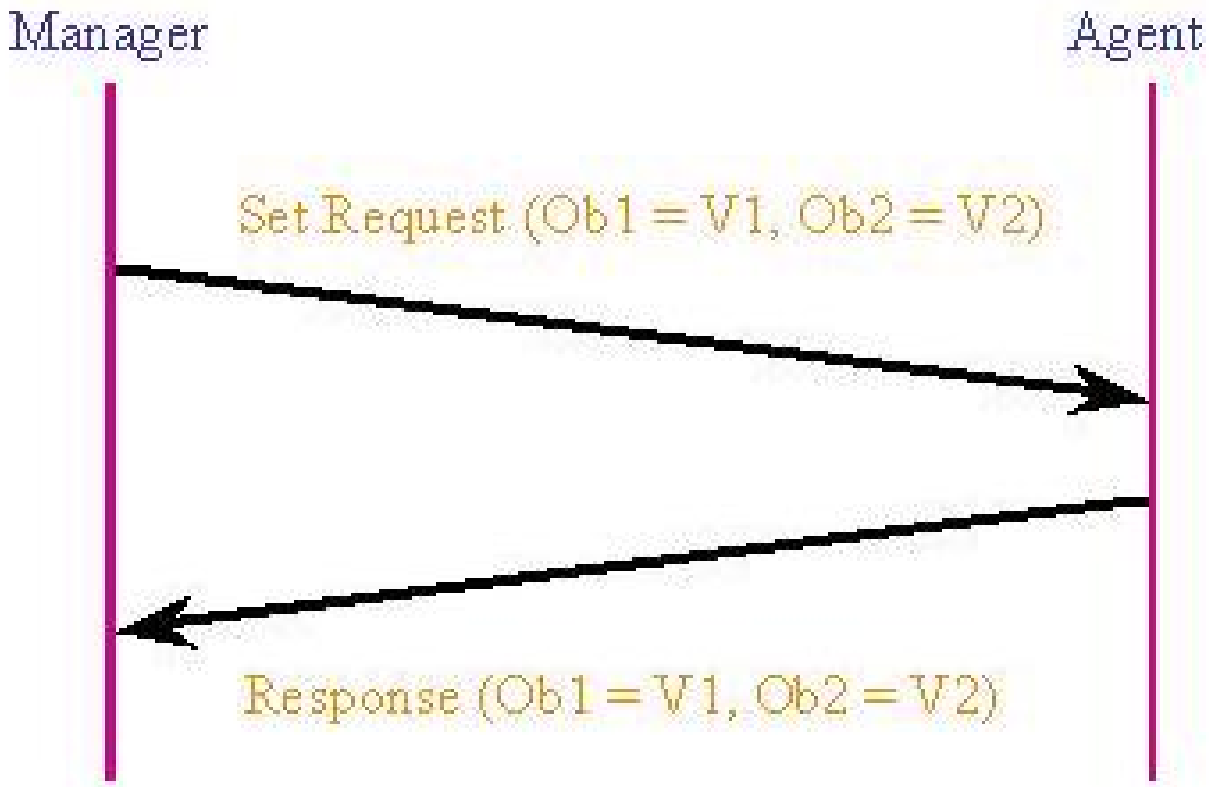
La notification TRAP



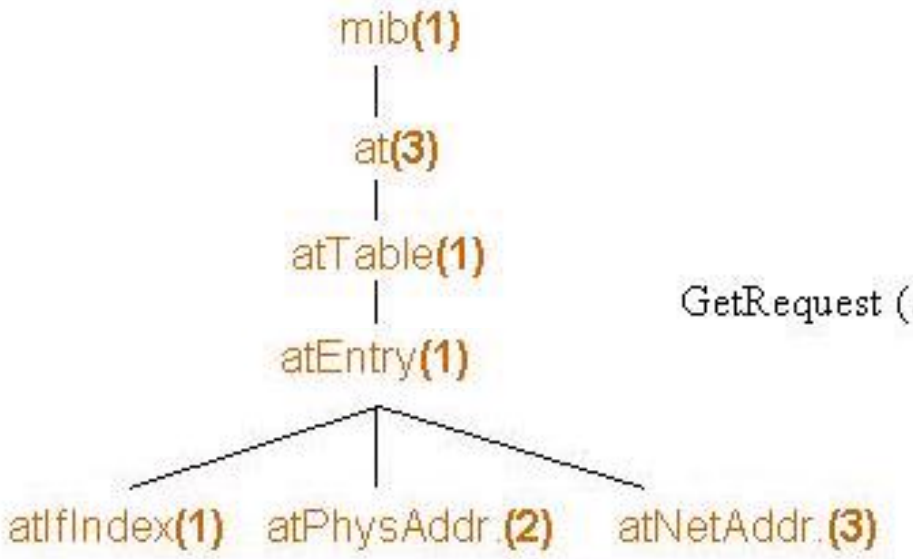


Les requêtes multiples

- Les requêtes Get, Get Next and Set Requests peuvent préciser plusieurs objets à lire ou à modifier leurs valeurs.



Exemple de Get Request



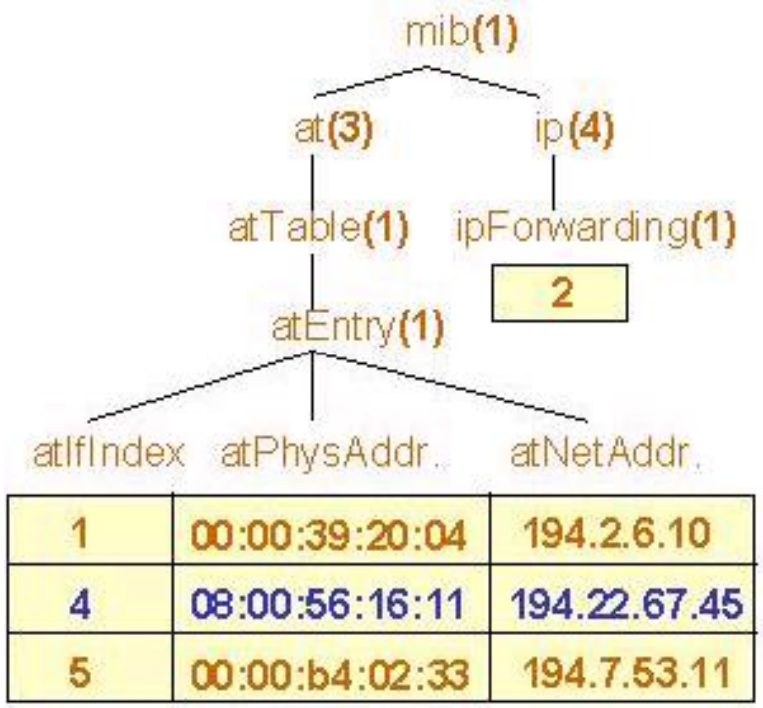
1	00:00:39:20:04	194.2.6.10
4	08:00:56:16:11	194.22.67.45
5	00:00:b4:02:33	194.7.53.11

To get the second row

GetRequest (atIfIndex.4, atPhysAddress.4, atNetAddress.4)

Response (atIfIndex.4 = 4,
atPhys.4 = 08:00:56:16:11,
atNet.4 = 194.22.67.45)

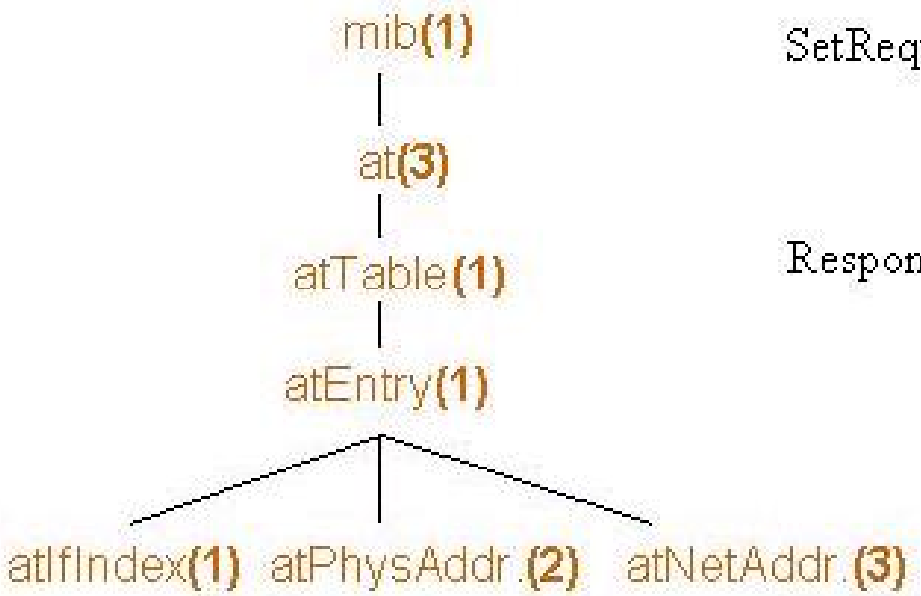
Exemple de GetNext Request



GetNextRequest (atIfIndex.1, atPhys.1, atNet.1)

Response (atIfIndex.4 = 4,
atPhys.4 = 08:00:56:16:11,
atNet.4 = 194.22.67.45)

Exemple de Set Request



1	00:00:39:20:04	194.2.6.10
4	00:00:77:b1:45	194.22.67.45
5	00:00:b4:02:33	194.7.53.11

SetRequest (atPhysAddress.4 = 00:00:77:b1:45)

Response (atPhysAddress.4 = 00:00:77:b1:45)

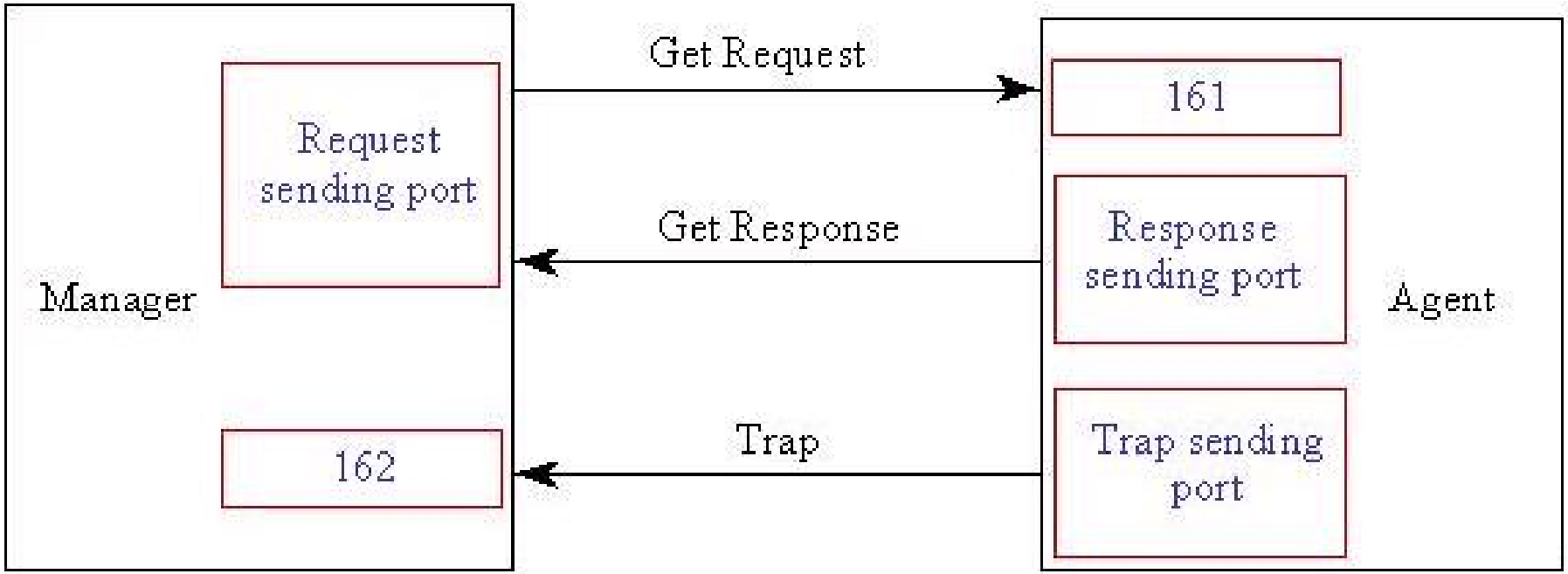
Numéros des Ports de SNMP

- Par convention, les numéros de port utilisés pour SNMP sont 161 (Requêtes) et 162 (Traps).

- Comportement de la station d'administration :
  - Écoute le port local 162 pour les traps envoyées par l'agent.
  - Envoie les requête à travers le port 161.

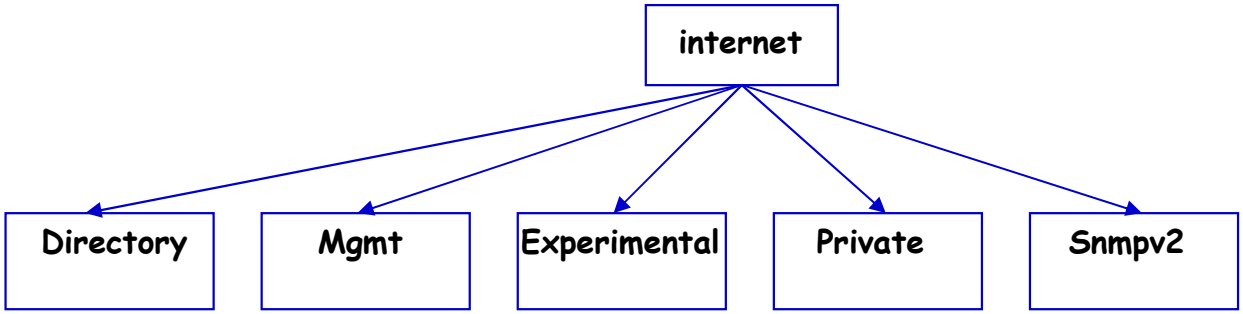
- Comportement de l'agent :
  - Écoute le port local 161 pour recevoir les requêtes envoyées par la station d'administration.
  - Envoie les traps à travers le port 162.

Numéros des Ports de SNMP

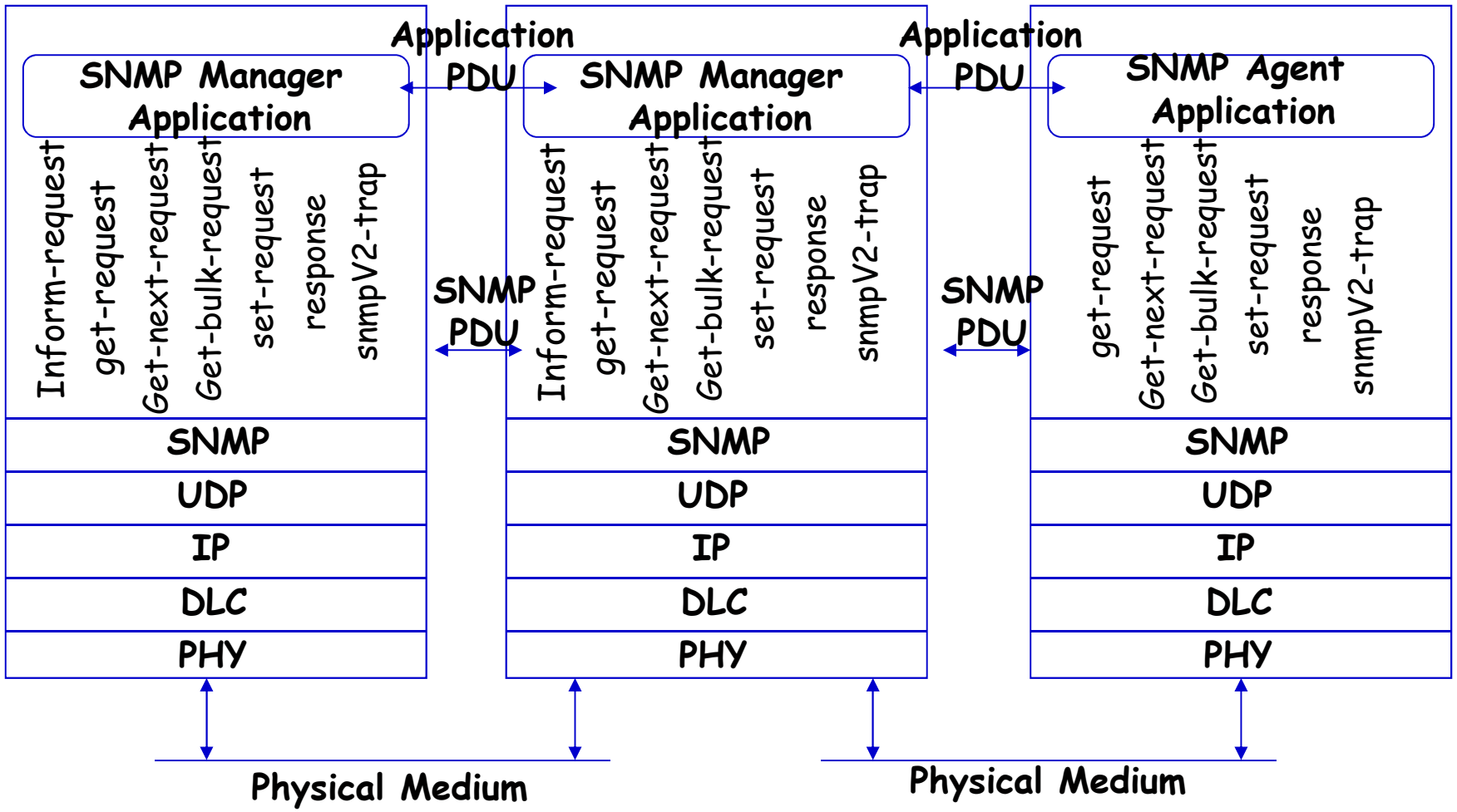


La version SNMPv2

- Les éléments de base de l 'administration des réseaux avec SNMPv2 sont les mêmes qu 'avec SNMPv1
- Le changement le plus significatif est qu ' un agent et un manager remplissent la même fonction.
- deux messages sont ajoutés :
  - la possibilité de demander et de recevoir un volume de données en utilisant le message get-bulk
  - l 'interopérabilité entre deux systèmes d 'administration : la communication entre deux systèmes d 'administration



L 'architecture de SNMPv2



Les opérations de SNMPv2

- Les messages **get-request**, **get-next-request**, et **set-request** sont les mêmes que ceux de SNMPv1 et ils sont générés par l 'application d 'administration.
- Le message **response** est le même aussi que celui de SNMPv1, mais il est généré dans ce cas par l 'agent ou le manager.
- Le message inform-request est généré par le manager et envoyé à un autre manager.
- Le message get-bulk-request est généré par le manager afin de transférer une grande quantité de données de l 'agent vers le manager.
- L 'événement SNMPv2-trap (notification) est généré et transmis par l 'agent quand une situation exceptionnelle apparaît.

Les opérations de SNMPv2

- La structure de données PDU dans SNMPv2 a été uniformisée pour tous les messages (sauf pour le message get-bulk-request) afin d 'améliorer les performances d 'échange.
- L 'amélioration la plus significative est que la structure de données des traps est la même que les autres

PDU Type	RequestID	Error Status	Error Index	VarBind 1 name	VarBind 1 value	...	VarBind n name	VarBind n value

  - avec SNMPv1 Les VarBinds ne sont pas toutes retournées dans le cas d 'une erreur (Error Status $\neq$ 0)
  - avec SNMPv2 uniquement la varBind qui génère l 'erreur est ignorée et le reste sera retournée dans la réponse.

Les opérations de SNMPv2

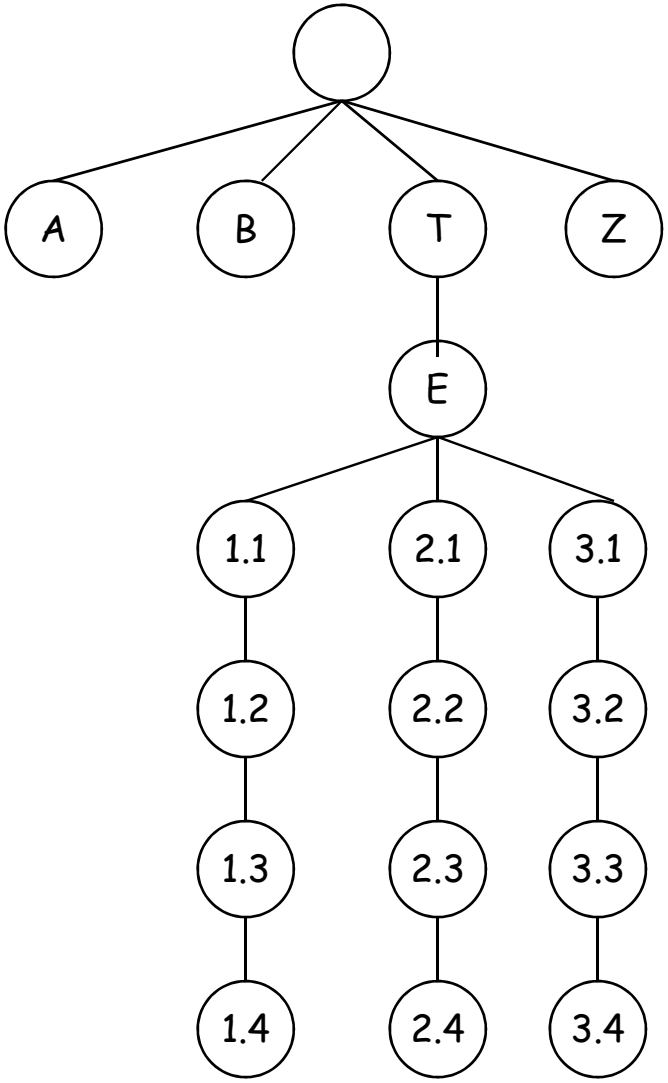
- La structure de données PDU get-bulk-request est :

PDU Type	RequestID	Non-Repeaters	Max Repetitions	VarBind 1 name	VarBind 1 value	...	VarBind n name	VarBind n value

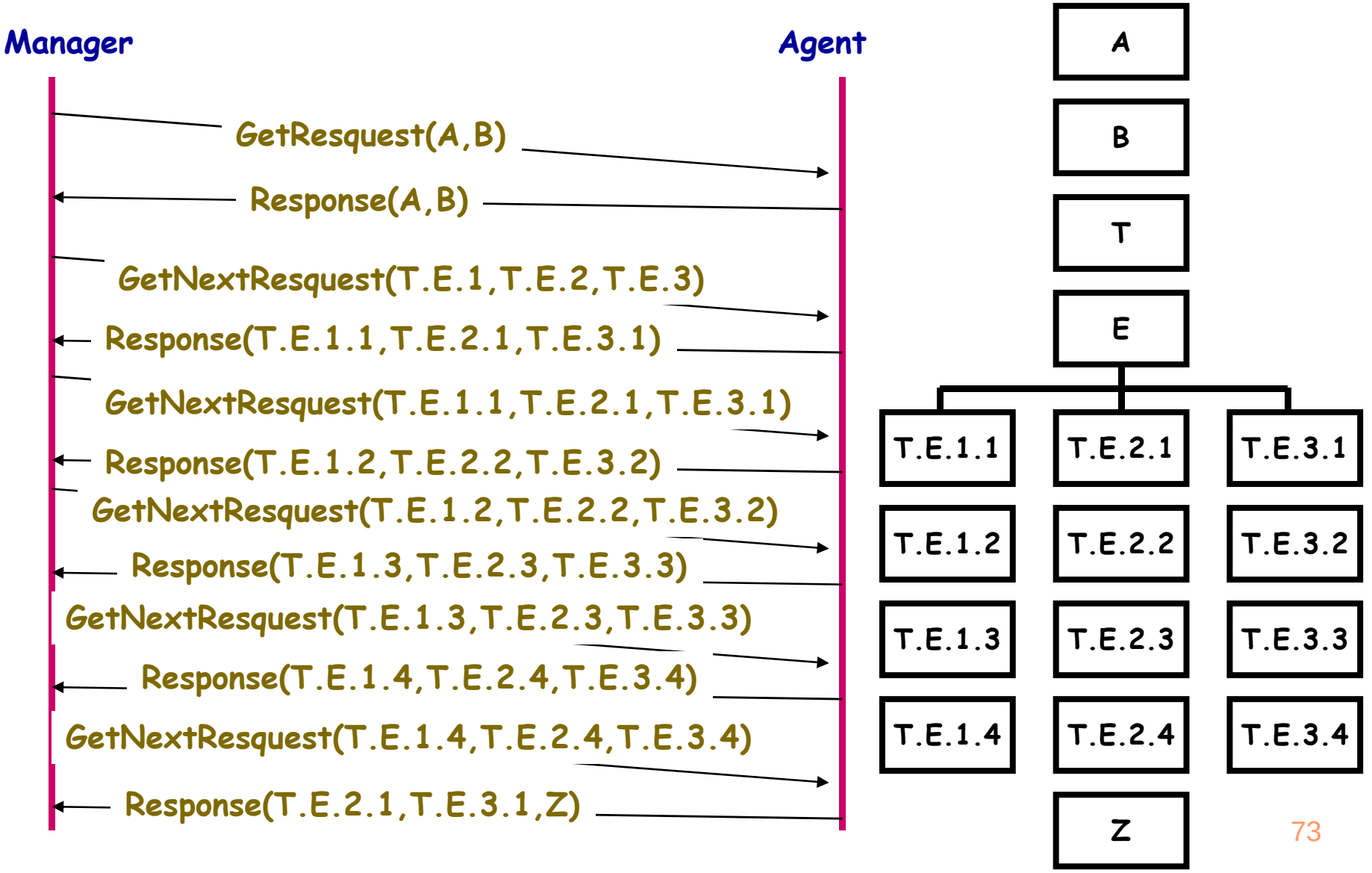
  - non-repeaters indique le nombre de variables non répété à retourner (variables atomiques)
  - max-repeaters indique le nombre de lignes à retourner (variables composées)
- get-next-request ne peut retourer qu 'une seule ligne (la ligne qui suit celle précisée par les varBinds)

Les opérations de SNMPv2

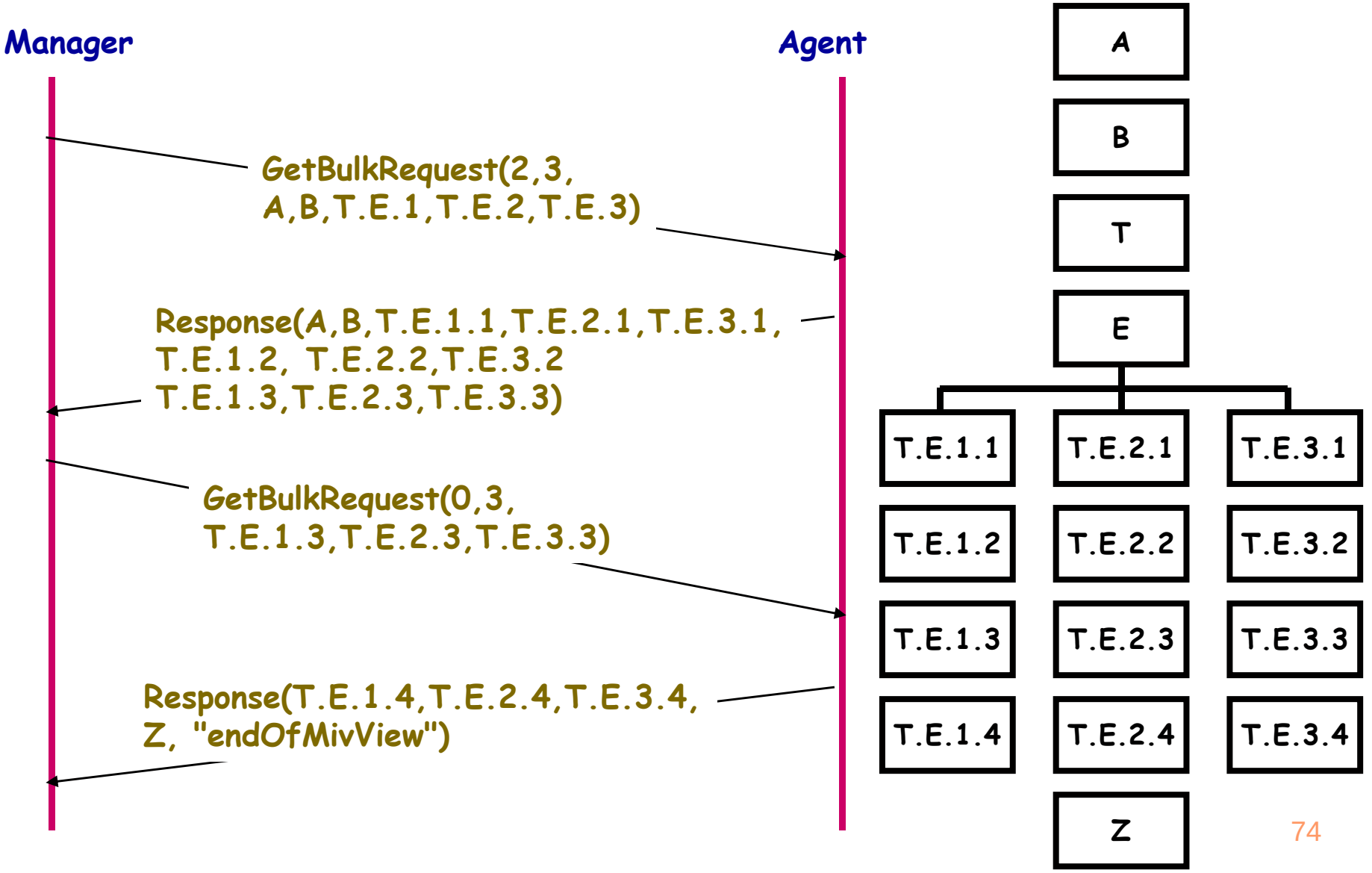
- Exemple :



Les opérations de SNMPv2



Les opérations de SNMPv2



La structure des informations d 'administration SNMPv2 (SMI2)

- La clause MAX-ACCESS :

```
«MAX-ACCESS» Access
Access ::= «read-only» | «read-write» | «read-create»
           | «not-accessible» | «accessible-for-notify»
```

SNMP V2

read-only : read access
read-write : read and write acceses
read-create : read, write and create accesses
not-accessible : not-accessible for any operation
accessible-for-notify : accessible only via a notification (trap)

```
«ACCESS» Access
Access ::= «read-only» | «read-write» | «write-only» |
           «not-accessible»
```

SNMP V1

La structure des informations d 'administration SNMPv2 (SMI2)

- La clause STATUS :

```
«STATUS» Status
Status ::= «current» | «obsolete» | «deprecated»
```

SNMP V2

current : the object is valid for the current standard
obsolete : the object should not be implemented
deprecated : the object is obsolete, but may be implemented to provide interoperability with older implementations

```
«STATUS» Status
Status ::= «mandatory» | «optional» | «obsolete»
           | «deprecated»
```

SNMP V1

La structure des informations d 'administration SNMPv2 (SMI2)

- La clause DESCRIPTION : même clause comme SNMPv1, mais obligatoire pour SNMPv2

```
DescrPart ::= «DESCRIPTION» Text
Text ::= « OCTET STRING »
```

SNMP V2

```
DescrPart ::= «DESCRIPTION» value (DisplayString)
                | empty
```

SNMP V1

La structure des informations d 'administration SNMPv2 (SMI2)

- La clause INDEX : plus complexe que celle de SNMPv1

```
IndexPart ::= «INDEX» «{» IndexType, IndexType, ... «}»
            | «AUGMENTS» «{» Entry «}»
            | empty
IndexType ::= «IMPLIED» value (ObjectName) | value (ObjectName)
Entry ::= value (ObjectName)
```

SNMP V2

```
IndexPart ::= «INDEX» «{» IndexType, IndexType, ... «}»
            | empty
IndexType ::= value (ObjectName)
```

SNMP V1

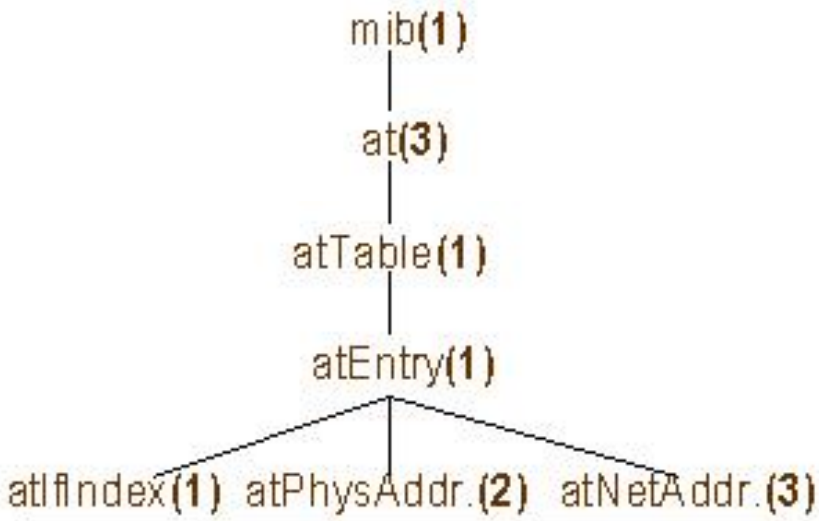
La structure des informations d 'administration SNMPv2 (SMI2)

- Exemple de table :

```
atTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF AtEntry
        MAX-ACCESS  not-accessible
        STATUS  deprecated
        DESCRIPTION  «Translation table»
        ::= { at 1 }

atEntry OBJECT-TYPE
        SYNTAX  AtEntry
        MAX-ACCESS  not-accessible
        STATUS  deprecated
        DESCRIPTION  «Translation entry»
        INDEX     {atIfIndex, atPhysAddress}
        ::= { atTable 1 }

AtEntry ::= SEQUENCE {
        atIfIndex          INTEGER,
        atPhysAddress      OCTET STRING,
        atNetAddress       IpAddress
        }
```



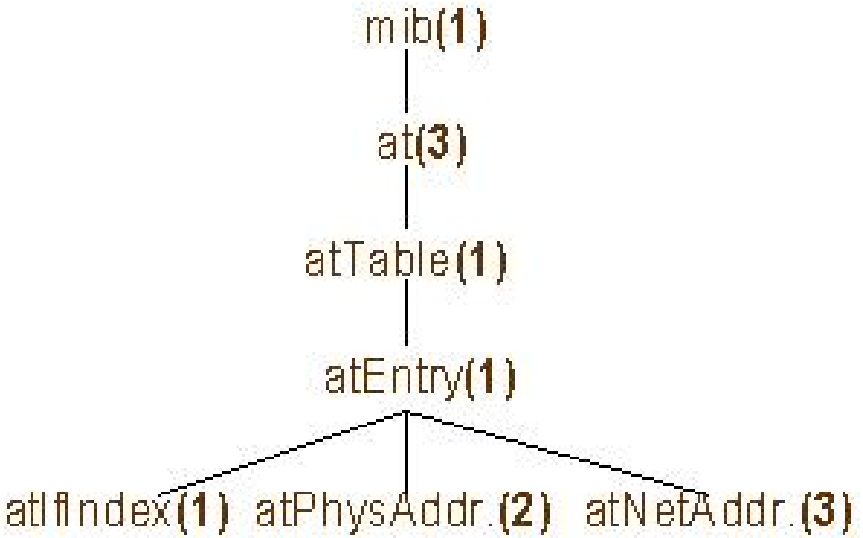
1	00:00:39:20:04	194.2.6.10
4	08:00:56:16:11	194.22.67.45
5	00:00:b4:02:33	194.7.53.11

La structure des informations d 'administration SNMPv2 (SMI2)

atIfIndex **OBJECT-TYPE**
SYNTAX INTEGER
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION «Index»
::= { *atEntry* 1 }

atPhysAddress **OBJECT-TYPE**
SYNTAX OCTET STRING
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION «Physical Address»
::= { *atEntry* 2 }

atNetAddress **OBJECT-TYPE**
SYNTAX IpAddress
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION «Network Address»
::= { *atEntry* 3 }



1	00:00:39:20:04	194.2.6.10
4	08:00:56:16:11	194.22.67.45
5	00:00:b4:02:33	194.7.53.11

Création et suppression de lignes dans des tables SMIv2

- L 'introduction d 'une nouvelle colonne d 'état appelée RowStatus.
- La création d 'une nouvelle ligne se fait par deux méthodes :
  - Créer une ligne et la mettre active immédiatement : RowStatus=CreateAndGo(4)
  - Créer une ligne et la mettre active ultérieurement : RowStatus=CreateAndWait(5)
- Les opération de création et suppression se fait par la requête Set-request
  - création : SetRequest(RowStatus=4 ou 5)
  - supression : SetRequest (RowStatus=6)

Création et suppression de lignes dans des tables SMIv2

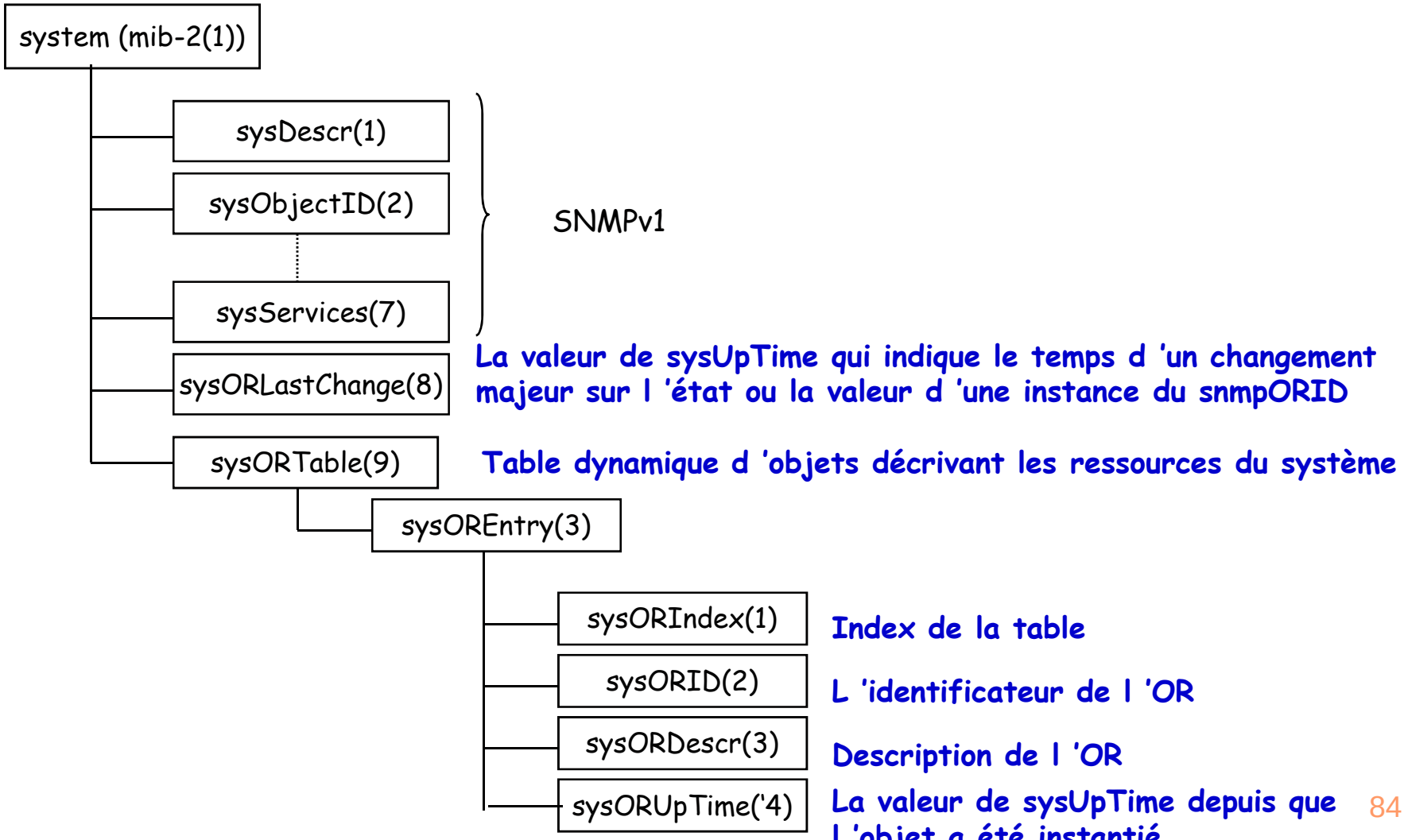
Etat	Valeur	Description
active	1	la ligne existe et active
notInService	2	l'opération sur la ligne est suspondue
NotReady	3	La ligne n'a pas prète à être opérationnelle
CreateAndGo	4	processus de création à une seule étape et se met en activité immédiatement
CreateAndWait	5	une ligne en cours de création
destroy	6	destruction de ligne

La MIB (Management Information Base)

- La MIB de SNMPv2 est définie dans la RFC 1907
- Trois nouveaux groupes dansSNMP V2 MIB :
  - *system* group :
    - extension du groupe original "MIB-II *system*"
    - le groupe SNMP V1 system + de nouveaux objets
  - *snmp* group :
    - rafinement du groupe original "MIB-II *snmp*"
    - le groupe SNMP V1 snmp + de nouveaux objets
  - *snmpMIBObjects* group : traite les "SNMPv2-Trap PDUs"
    - snmpTrap subgroup : Informations à propos des traps générés par les agents
    - snmpSet subgroup : Utilisé pour résoudre des problèmes qui proviennent des opérations SET.

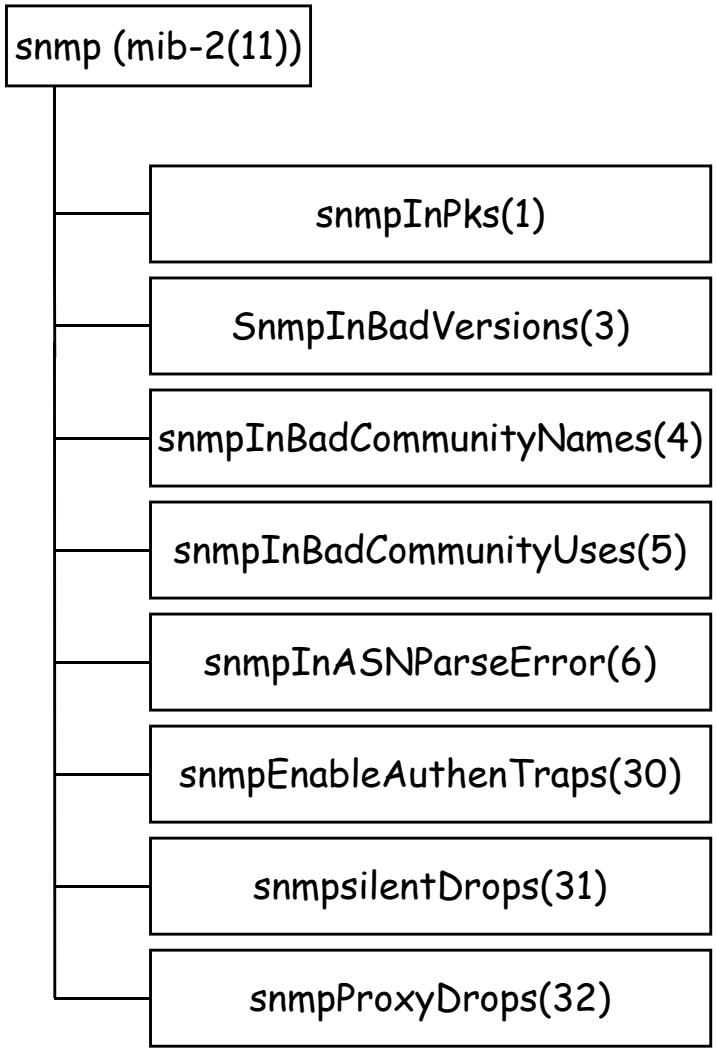
La MIB (Management Information Base)

- Le groupe "system"



La MIB (Management Information Base)

- Le groupe "snmp"



Le nombre total de messages délivrés du service de transport

Le nombre total de messages du service de transport dont la version est invalide

Le nombre total de messages du service de transport dont le nom de communauté est inconnu

Le nombre total de messages du service de transport dont le nom de communauté est non autorisé par l 'opération

Le nombre total d 'erreurs sur ASN.1 et BER

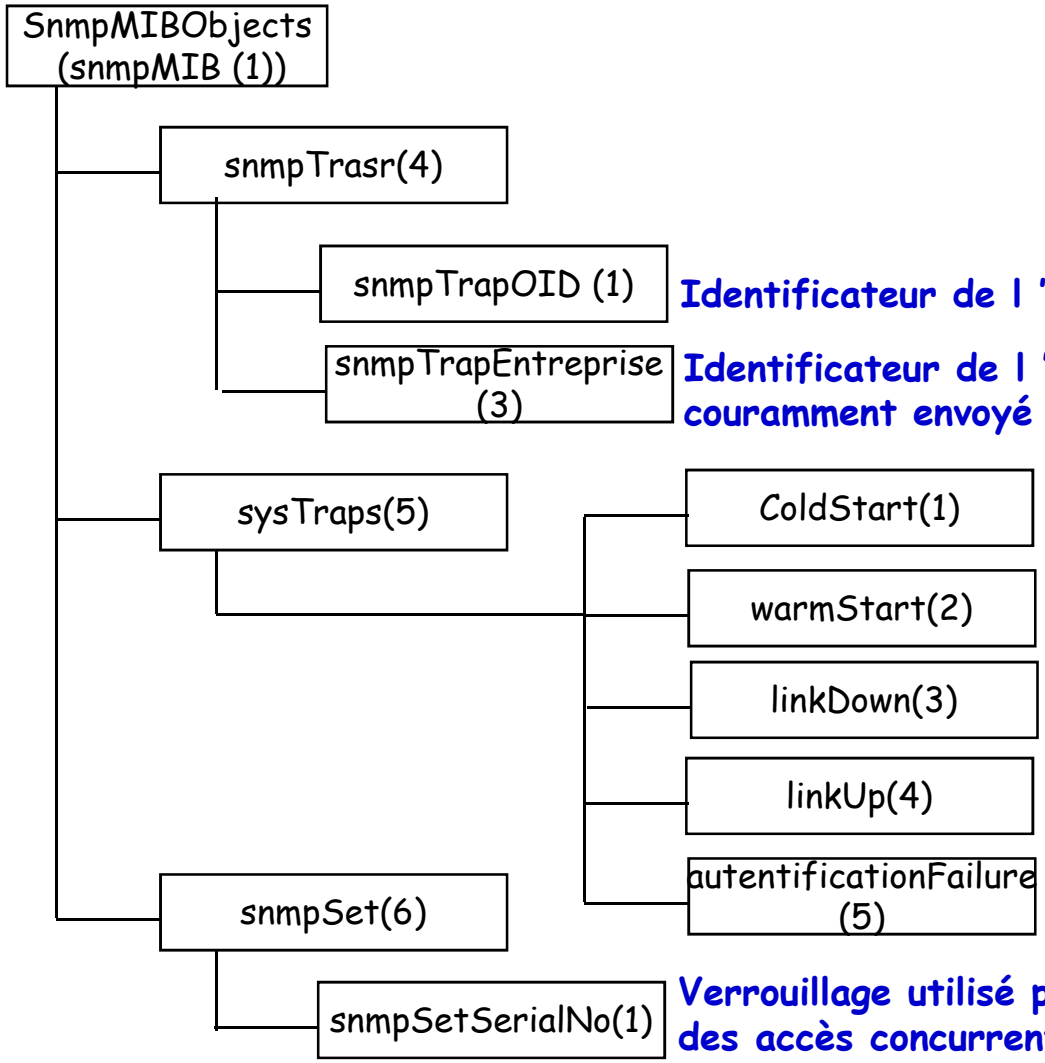
Indique si l 'entité SNMPv2 est autorisée à générer des traps pour un échec sur l ' 'authentification.

Le nombre total de requêtes échouées à cause d 'une réponse qui dépasse la taille maximale ou une exception sur varbinds

Le nombre total de requêtes échouées à cause d 'une inaccessibilité avec un proxy

La MIB (Management Information Base)

- Le groupe "snmpMIBObjetcs"



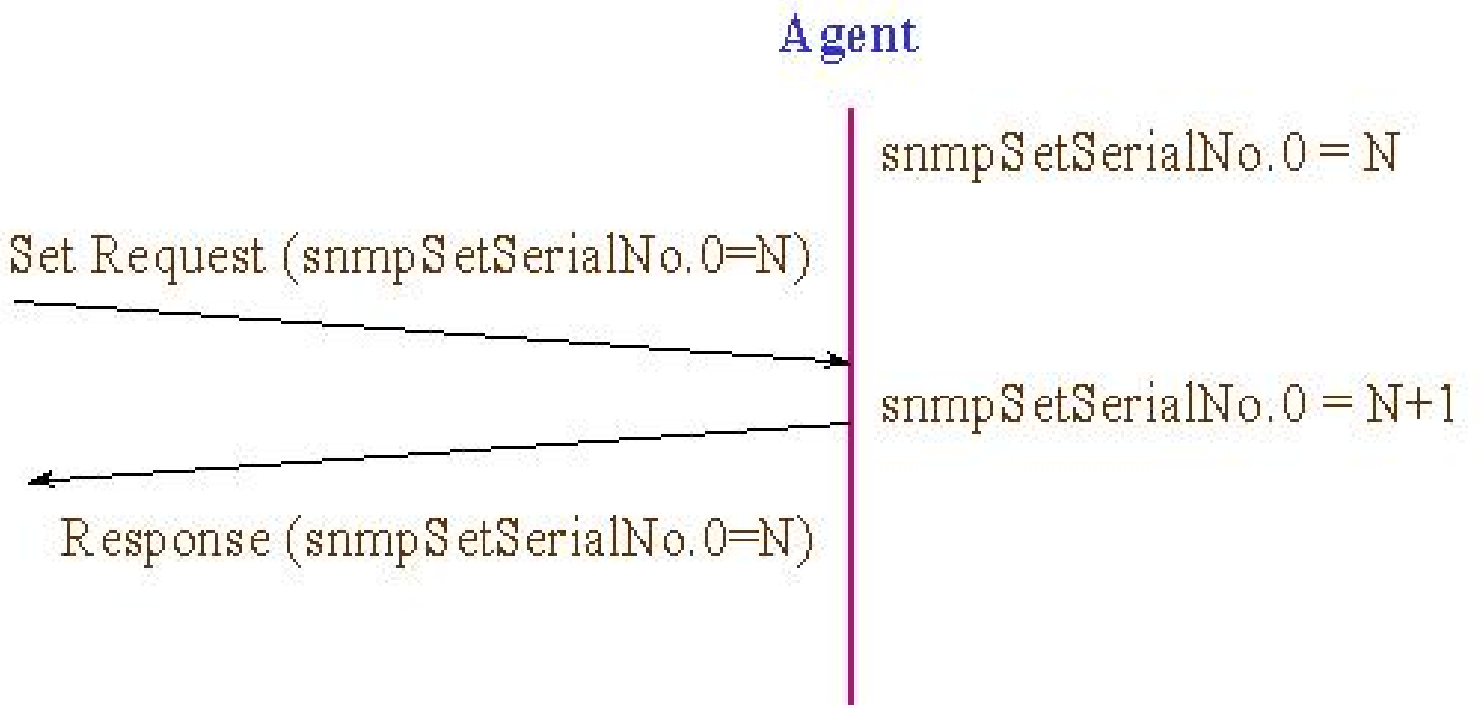
Identificateur de l 'objet du trap couramment envoyé

Identificateur de l 'objet de l 'entreprise associé au trap couramment envoyé

Verrouillage utilisé pour protéger les objets de la MIB contre des accès concurrents de plusieurs Managers avec l 'opération Set

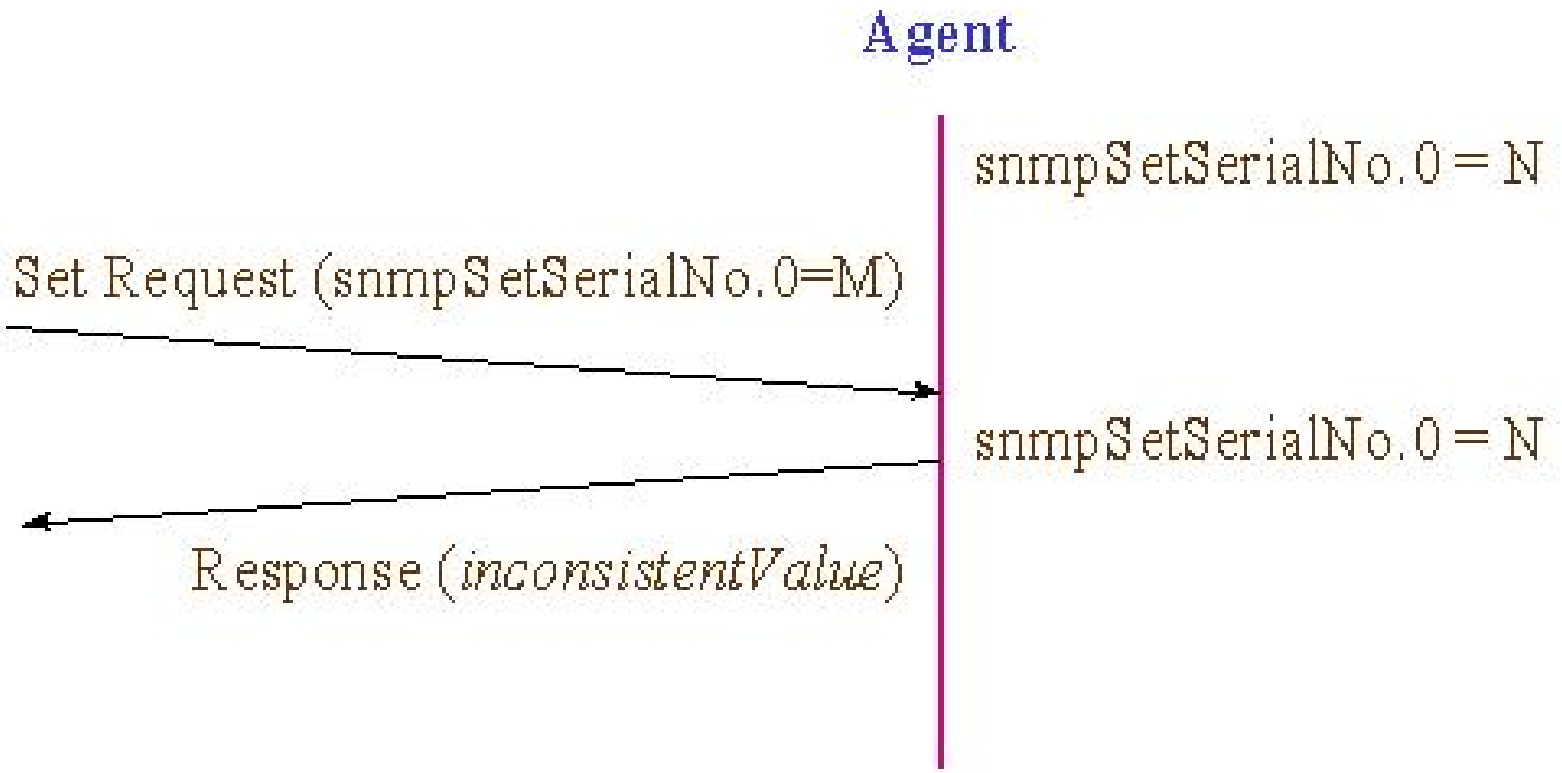
La MIB (Management Information Base)

- L'agent accepte l'opération SET sur le snmpSetSerialNo si la valeur invoquée est la même que celle de la valeur courante
- la valeur de snmpSetSerialNo est incrémentée de 1

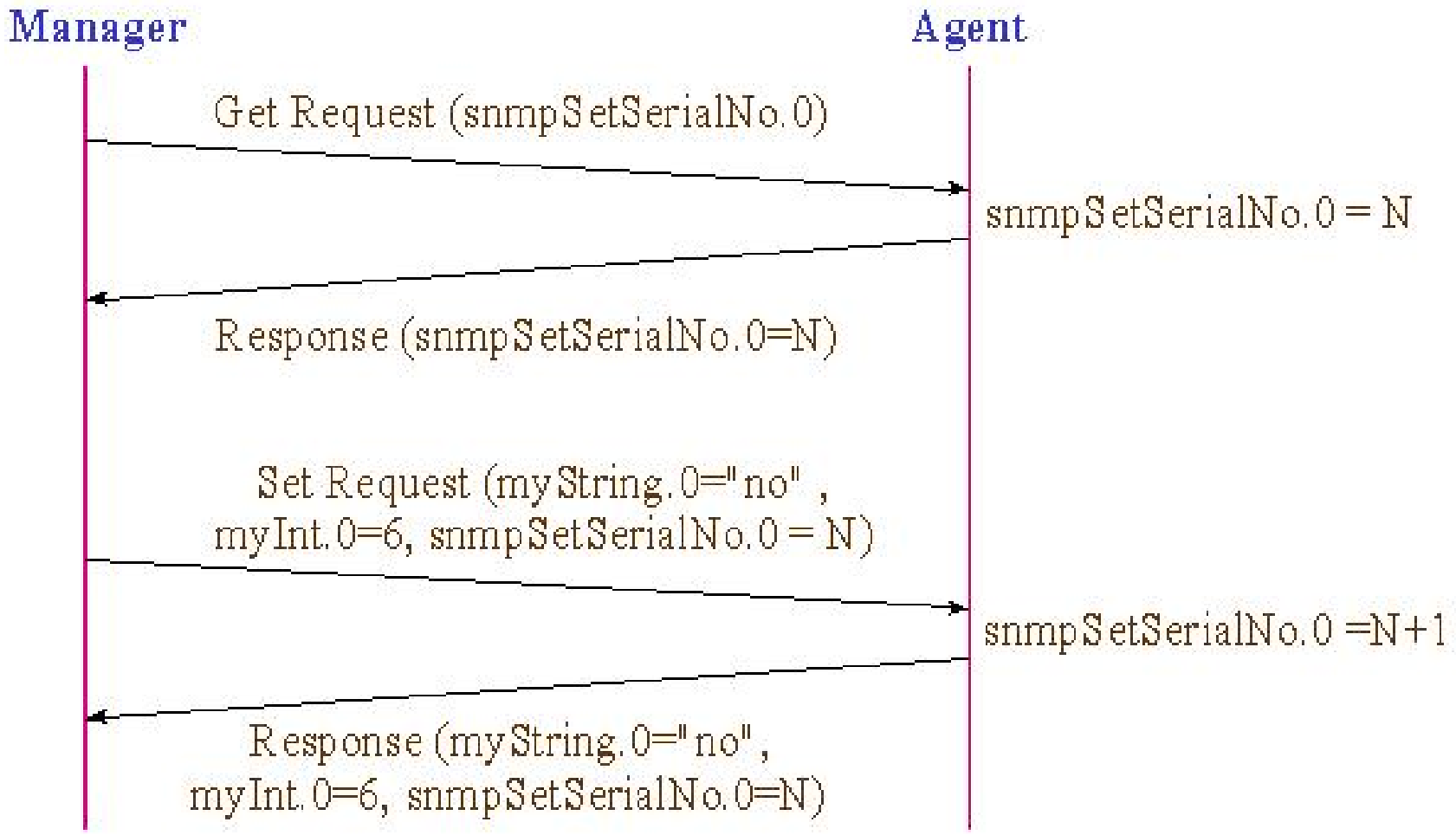


La MIB (Management Information Base)

- L'agent refuse l'opération SET sur snmpSetSerialNo si la valeur invoquée est différente de la valeur courante



La MIB (Management Information Base)



La MIB (Management Information Base)