

SNMP

Simple Network Management Protocol

Pr B. REGRAGUI

SOMMAIRE

- ✓ Introduction
- ✓ Architecture du protocole SNMP
- ✓ M.I.B. (*Management Information Base*)
- ✓ Échanges de données
- ✓ Avantages / Inconvénients

SNMP

En 1988, l'**IAB** (Internet Activities Board) approuva le développement du protocole **SNMP** (Simple Network Management Protocol).

L'IAB imposa que SNMP utilise une base d'objets gérables. Donc une **SMI** (Structure of Management Information) et une **MIB** (Management Information Base) communes devaient être définies et utilisés.

En 1989, SNMP a été adopté par de nombreux constructeurs et est devenu à ce jour un standard très répandu de gestion réseau.

Modèle OSI

Le modèle OSI décompose le réseau en sept niveaux « couches ».

- Les couches qui se rapportent au *matériel* ont pour rôle de faire suivre physiquement les données et d'en assurer la sécurisation et la synchronisation au sein du réseau.
- Les couches orientés *transport* régissent le transport et la distribution des données (*routage et commutation*).
- Les couches qui se rapportent aux *applications* ont pour rôle d'établir la session et d'y mettre fin, d'assurer le transfert des données et de présenter les données à l'utilisateur.

Architecture du protocole SNMP

Modèle OSI		Modèle TCP/IP
7	Application	SNMP
6	Présentation	
5	Session	
4	Transport	UDP
3	Réseau	IP
2	Liaison	Interface Réseau
1	Physique	

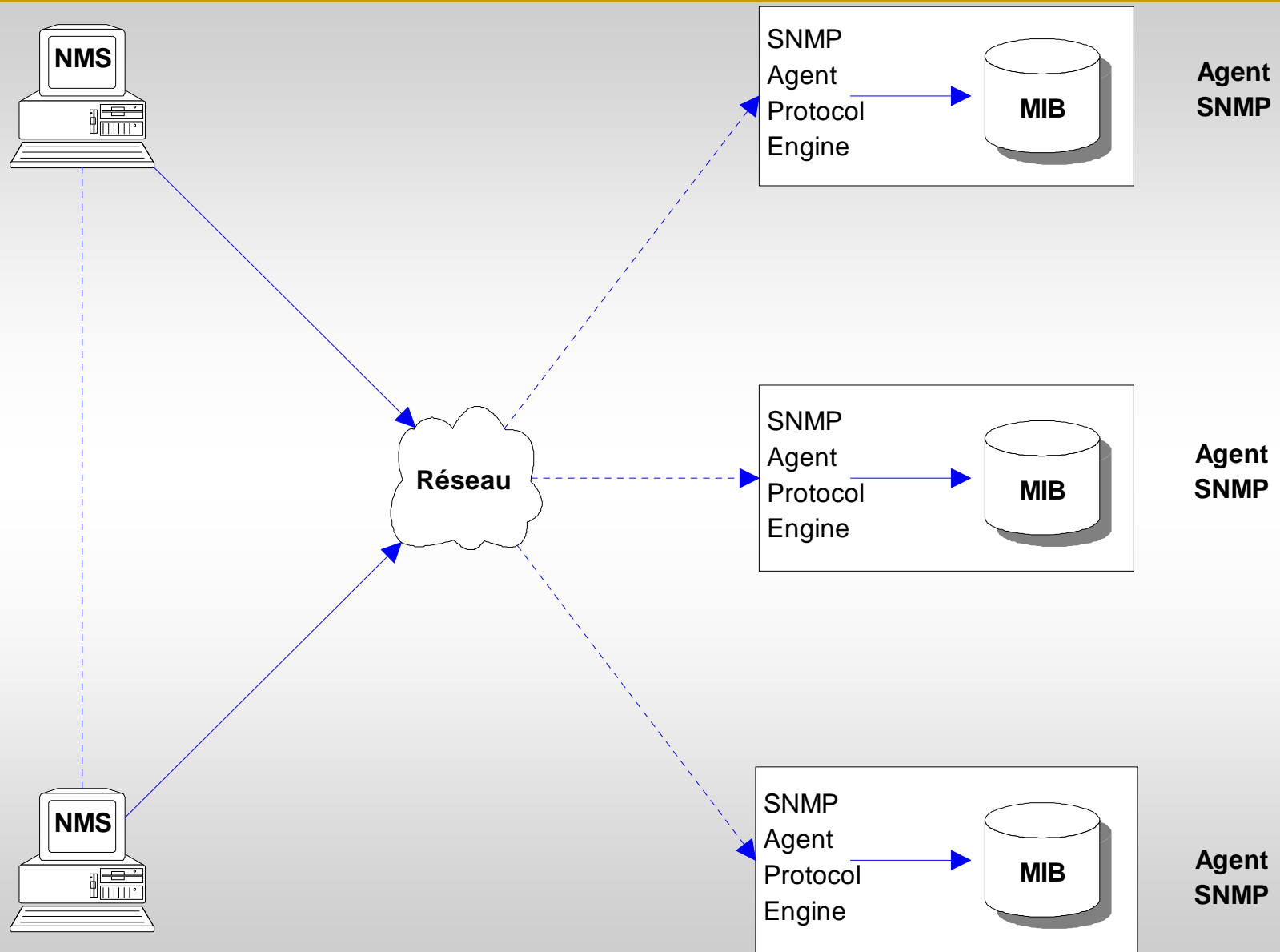
Le protocole SNMP s'appuie sur l'architecture réseau de la norme OSI.

Un modèle Client-Serveur

SNMP est un protocole de gestion de réseau. Il part du principe qu'un système d'administration réseau se compose :

- de nœuds administrés (*MN = Managed Node*) chacun contenant un **agent**. Les agents sont les **serveurs**.
- d'au moins une **station d'administration**. (*NMS = Network Management Station*). Cette station d'administration est le **Client**
- d'un protocole réseau utilisé par la NMS et les agents pour échanger des informations d'administration.(ici **SNMP**).

Architecture du protocole SNMP



Administrateur

La station d'administration doit posséder des ressources importantes.

Il existe plusieurs logiciels superviseurs de réseau à interface graphique. TIVOLI (IBM), SNMPc (PC), XSNMP (UNIX), HP-OpenView (PC et UNIX).

➤ Agent

Les nœuds *administrables* (MN) se décomposent en trois catégories:

- les **hosts** (station de travail, serveur de terminaux, imprimantes réseau, ...),
- les **équipements de raccordement** (commutateurs, répéteurs, ponts, HUB, routeurs, passerelles...)
- les **média** (coaxiaux, paire torsadée, fibre optique, liaison spécialisée,).

Représentation des données dans SNMP

Pour la gestion de la commuté Internet, l'**IETF** (Internet Engineering Task Force) a défini 3 **RFCs** (Requests For Comment)

↪ Documents définissant le standard:

- ✓ **RFC 1155:** décrit la structure et le nommage de l'information de gestion
- ✓ **RFC 1157:** définit le protocole SNMPv1 utilisé pour accéder via le réseau aux objets gérés
- ✓ **RFC 1213:** décrit la base d'information de gestion MIB

Représentation des données dans SNMP

SMI (Structure of Management Information) définit comment chacun des éléments d'information, qui concerne les équipement gérés, est représenté dans la MIB (Management Information Base)

Les objets dans la MIB sont définis en utilisant le langage ASN.1 (Abstract Syntax Notation One)

Chaque type d'objet a son nom, sa syntaxe et son encodage

Représentation des données dans SNMP

↳ Chaque groupe d'objet a un **nom**, une **syntaxe** et un **codage**

➤ **Nom** est représenté comme **Identificateur** d'objet qui est un nom administratif

Un identificateur d'objet est une suite de valeurs entières, positives ou nulles, qui parcourt une arborescence.

Représentation des données dans SNMP

↪ Chaque groupe d'objet a un **nom**, une **syntaxe** et un **codage**

➤ **Syntaxe** utilisée ASN.1 (Abstract Syntax Notation 1)

L'ASN1 est un langage formel, défini sous forme de grammaire. Il permet de définir les structures de données indépendamment de la représentation et de la limitation interne des machines. Dans l'environnement de gestion, l'ASN1 est utilisé pour définir

- ✓ Les structures des PDU échangées par le protocole de gestion
- ✓ Les objets gérés

➤ Quelques types

- ✓ Integer : valeurs entières
- ✓ Octet String : entre 0 et 255
- ✓ Object Identifier : type de donnée de type ASN.1
- ✓ Null

Représentation des données dans SNMP

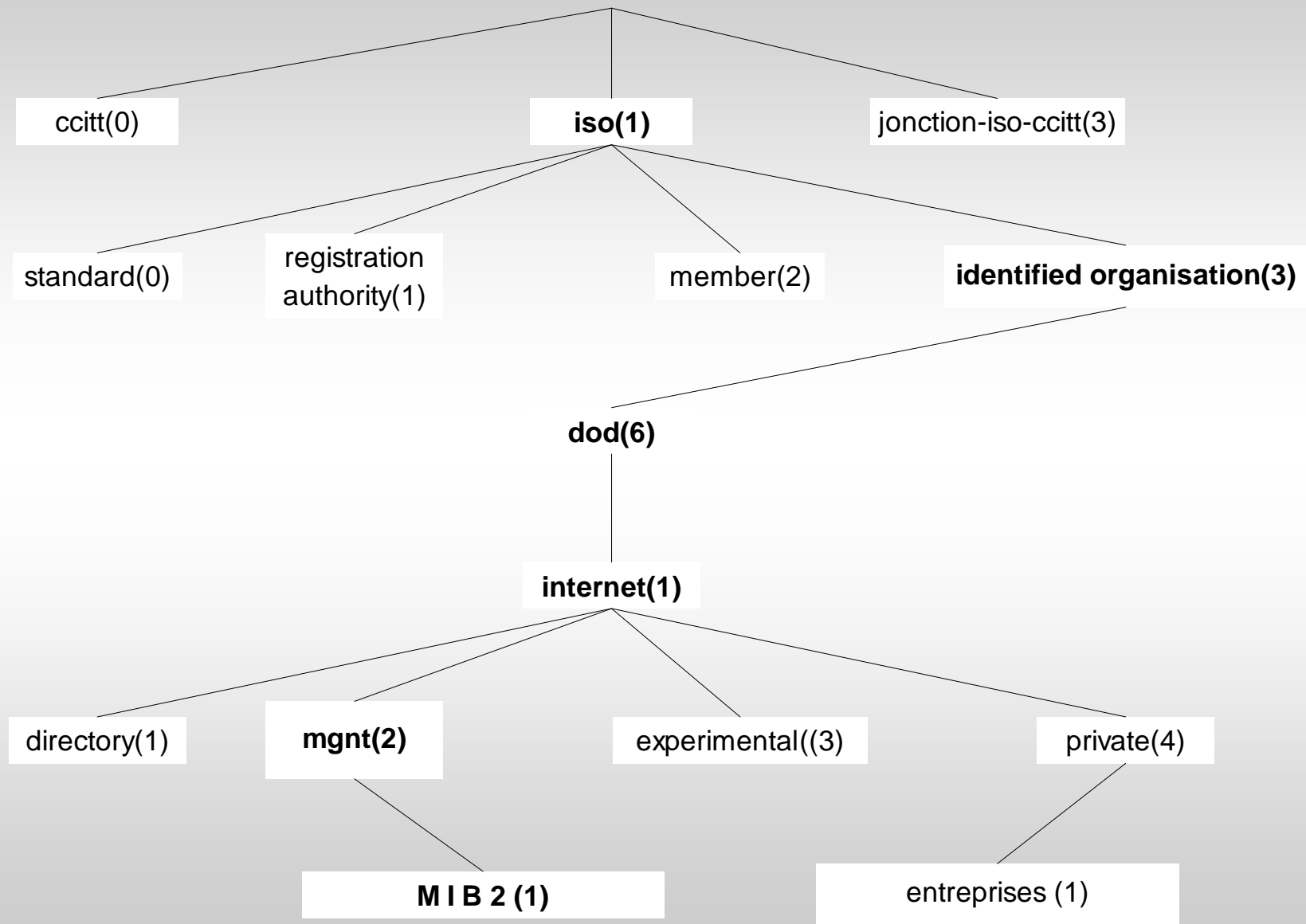
- ↳ Chaque groupe d'objet a un **nom**, une **syntaxe** et un **codage**
 - Le **codage** des structures de données, représentées en ASN1, est réalisé par une notation de syntaxe de transfert appelée BER (Basic Encoding Rules).

Représentation des données dans SNMP

SNMP procède de deux façons pour nommer les objets d'une MIB:

- la première est un nom unique par objet (ex: sysUpTime)
- la seconde utilise les notations d'ASN.1 (Abstract Syntax Notation)

M.I.B. (Management Information Base)



M.I.B. (Management Information Base)

- La classification des objets est arborescente. L'identificateur d'un objet est défini, en ASN.1 par le chemin qui conduit à l'objet.
- Par exemple, pour accéder à un objet d'administration, son identificateur autrement appelé **OID** commencera par *1.3.6.1.2 (iso.org.dod.internet.mgmt)*.
- Une MIB est donc simplement une collection de tous les objets que maintient un agent donné.
- Le premier standard utilisé pour la définition des objets d'administration de la MIB standard fut la MIB-I.

M.I.B. (Management Information Base)

L'OID de la MIB I est : 1.3.6.1.2.1 et sa définition est la suivante:

Numéro	Objet	Nombre de sous-objets
1	system	3
2	interfaces	23
3	at	3
4	ip	33
5	icmp	26
6	tcp	17
7	udp	4
8		

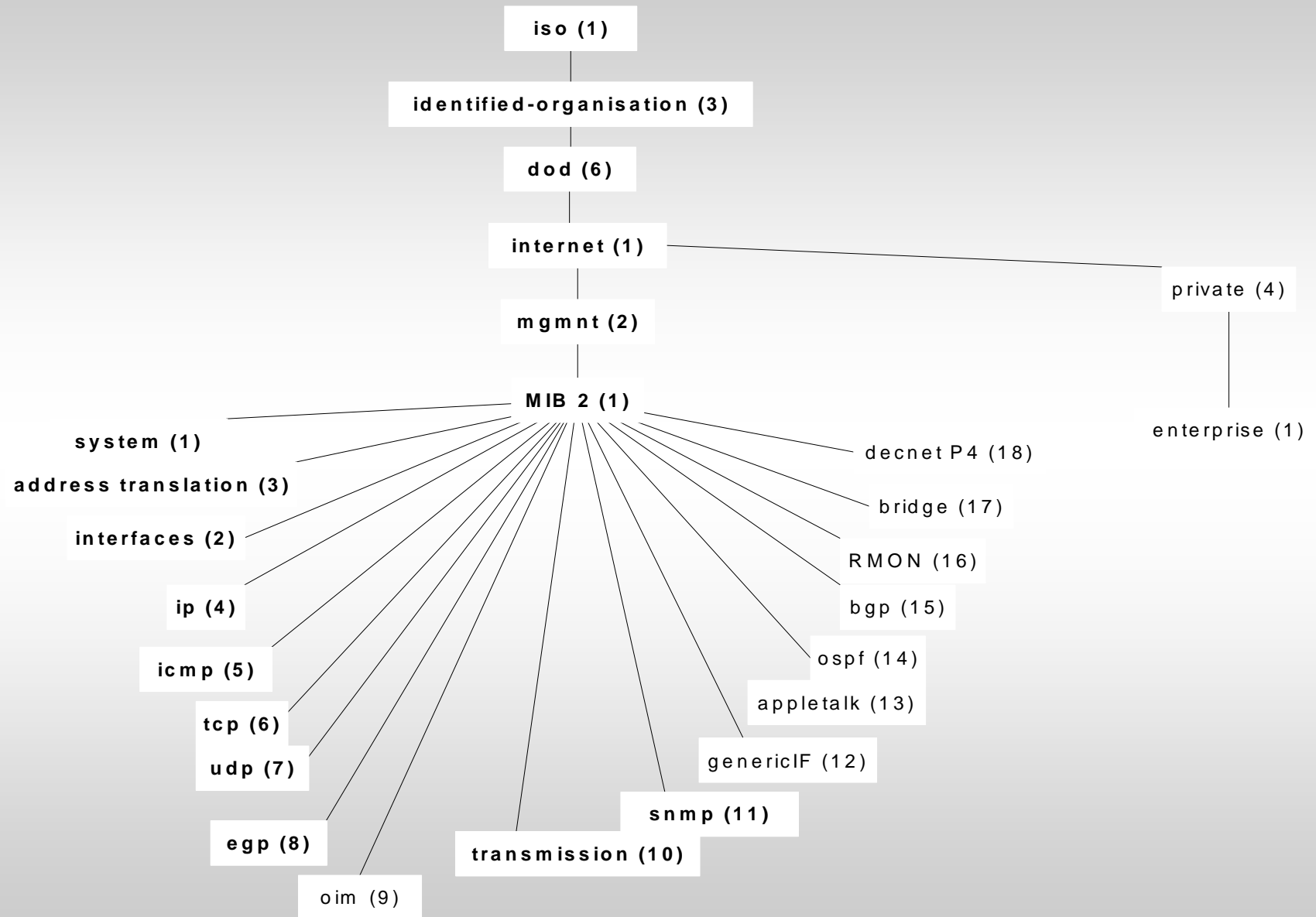
Contient une centaine d'objets rangés par groupes fonctionnels qui permettent de gérer uniquement un réseau TCP/IP;
Les 5 premiers groupes sont obligatoires,
les autres en fonction du protocole utilisé.

La MIB II

Un second standard fut défini pour rajouter des objets dans quelques unes des catégories de la MIB standard.

La MIB-II fort de ses 172 éléments, a remplacé actuellement la MIB I. Son OID est donc aussi : **1.3.6.1.2. 1**

M.I.B. (Management Information Base)



M.I.B. (Management Information Base)

Ajout de sysContact et sysLocation

Numéro	Objet	Nombre de sous-objets	Description
1	System	7	Informations générales concernant l'objet à travers le système.
2	Interfaces	23	Informations concernant chaque interface IP de l'agent
3	Address Translation	3	La table de translation d'adresses qui réalise la correspondance entre l'adresse MAC et l'adresse IP
4	IP	38	Compteurs IP
5	ICMP	26	Compteurs ICMP
6	TCP	19	Compteurs TCP
7	UDP	7	Compteurs UDP
8	EGP	18	Compteurs EGP
9	CMOT	0	Compteurs pour CMOT (protocole OSI équivalent à SNMP)
10	Transmission	0	Modes de transmission et protocoles d'accès de chaque interface. Remplacera <i>at</i>
11	SNMP	30	Statistiques du trafic SNMP.

M.I.B. (Management Information Base)

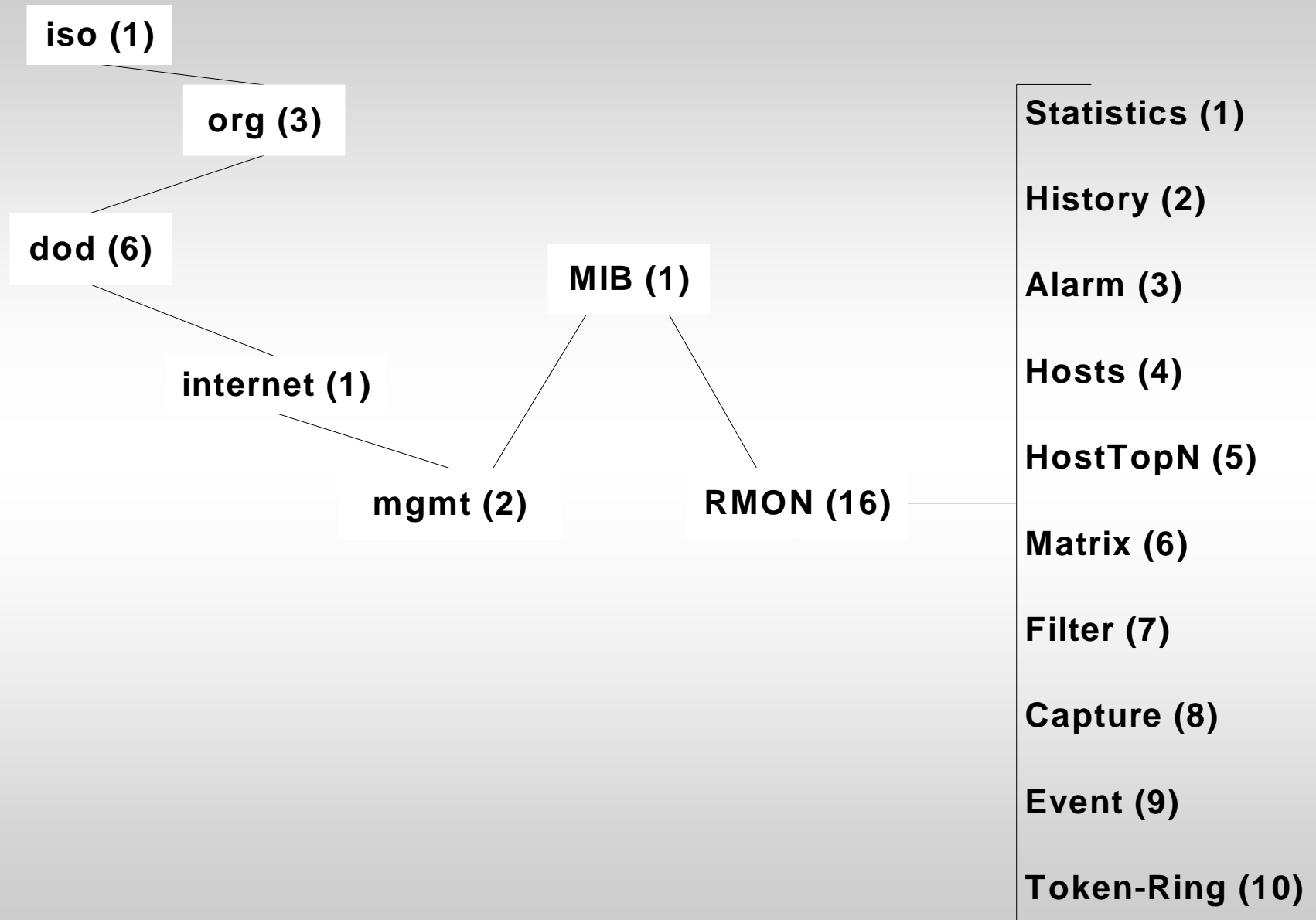
Les objets du groupe interface de la MIB II:

ifDescr:	Description de l'interface
ifType:	Identification du type d'interface.
ifMtu:	Taille maximale en octets des datagrammes IP qui peuvent être émis ou reçus sur l'interface.
ifSpeed:	Largeur de bande de la ligne en bits par seconde.
ifPhysAddress:	Adresse physique de l'interface si elle existe (internet)
ifAdminStatus:	Etat administratif de l'interface.
ifOperStatus:	Etat opérationnel de l'interface
ifLastChange:	Date du dernier passage de l'interface à l'état opérationnel
ifInOctets:	Nombre d'octets reçu sur l'interface.
ifInUcastPkts:	Nombre de paquets unicast de sous réseau transmis au protocole de niveau supérieur.
ifInNUcast...	
ifInDiscards:	Nombre de paquets reçu et volontairement détruits
ifInError:	Nombre de paquets reçus contenant des erreurs.
ifInUnknownProtos:	Nombre de paquets reçus et détruits car contenant un type de protocole de niveau supérieur non identifié

La MIB RMON

La MIB RMON est une “ *extension* ” de la MIBII, on lui a attribué le ‘*subtree identifier*’ **16**, c'est à dire que la MIB RMON a comme OID **1.3.6.1.2.1.16**.

M.I.B. (Management Information Base)



La gestion des performances

La gestion des performances nécessite la mesure permanente de l'ensemble des indicateurs de santé du réseau considéré. Pour cela, la MIB RMON possède tous ces indicateurs sous forme d'objets de type *counter*. Ces objets sont contenus dans le groupe **Statistics** pour l'aspect *temps réel*. Pour l'aspect *temps différé* ces mêmes objets sont repris dans le groupe **History** qui permet de définir des collectes sur des périodes plus longues

La gestion des anomalies

La gestion des anomalies utilise la notion d'alarme (traps) qui doit être émise pour avertir le superviseur d'une situation anormale nécessitant une action plus ou moins rapide et corrective. Pour mettre en place ce type de fonctionnalité, la MIB RMON possède un groupe **Alarm** qui permet de définir une alarme sur n'importe quel *compteur* de la MIB II et de la MIB RMON. Associé à ce groupe **Alarm**, on trouve le groupe **Event** qui permet de définir le type d'action que l'on associe au déclenchement de l'alarme.

La gestion des stations

Lorsqu'un dysfonctionnement se produit (augmentation de l'activité, avalanche de broadcasts,...), il est nécessaire pour localiser et corriger le problème et de connaître le ou les équipements responsables.

Pour cela, la MIB RMON possède 3 groupes qui sont :

Host, HostTopN, Matrix

Le groupe Host

Ce groupe contient une table où chaque ligne correspond à une adresse MAC avec toutes les statistiques de trafic associées

(octets émis, reçus,
paquets émis, reçus,
erreurs émises,
broadcasts émis,
multicasts émis).

Le groupe HostTopN

Ce groupe nécessite la définition d'une étude pour fonctionner. Il permet de définir des études en classant un certain nombre d'équipements suivant des indicateurs d'activité choisis.

Une étude nécessite :

Le type d'indicateurs; Le nombre d'équipements

La durée de la période de mesure

Les indicateurs sont :

Le nombre d'octets émis

Le nombre d'octets reçus

Le nombre de paquets émis

Le nombre de paquets reçus

Le nombre de broadcasts émis

Le nombre de multicasts reçus

Le groupe Matrix

Ce groupe contient une table où chaque ligne correspond à un couple d'adresses MAC qui ont au moins échangé un paquet. Les objets associés à chaque couple sont le nombre d'octets, le nombre de paquets et le nombre d'erreurs.

L'entrée dans la table se fait par les adresses MAC et le type d'objet désiré.

Analyse du trafic :

Lorsque le problème est détecté et les équipements en cause identifiés, il est souvent intéressant de connaître le type d'application utilisée par ces équipements.

Les groupes **Packet Capture** et **Filter** permettent de définir les captures de trafic désirées.

La **MIB RMON** permet aussi un déclenchement automatique de capture de trafic lorsqu'une alarme programmée dans le groupe **Alarm** est validée.

Architecture du protocole SNMP

Groupe <i>Statistics</i>	Groupe <i>History</i>	Groupe <i>Alarm</i>
<p>Contient toutes les informations associées au fonctionnement d'un réseau local Ethernet. (performances temps réel)</p> <ul style="list-style-type: none"> - Nombre d'octets sur le réseau - Nombre de paquets - Répartition par taille de paquets - Multicasts - Broadcasts - CRC/Align - Jabbers - Fragments (Runts) - OversizePackets - UndersizePackets - Collisions 	<p>Définition de campagnes de collectes permettant d'avoir des informations sur des indicateurs réseau. (performances temps différé)</p> <ul style="list-style-type: none"> - Nombre d'octets - Nombre de paquets - Broadcasts - Multicasts - CRC/AlignErrors - UndersizePackets - OversizePackets - Fragments (Runts) - Jabbers - Collisions - Estimation de l'utilisation en % du réseau pendant la collecte 	<p>Définition des alarmes</p> <p>ex: Activité réseau > 40% pendant 1 minute.</p> <ul style="list-style-type: none"> - Objet concerné - Variation ou valeur absolue - Intervalle de mesure - mode de déclenchement (seuil en montée, descente) - Valeur du seuil en montée - Valeur du seuil en descente - Pointeur vers la table d'actions (Groupe EVENT)

Architecture du protocole SNMP

Groupe Host	Groupe HostTopN	Groupe Matrix
<p>Contient les informations de trafics associées à chaque nœud Ethernet découvert.</p> <ul style="list-style-type: none"> - paquets émis - paquets reçus - octets émis - octets reçus - paquets erreurs émis - paquets broadcasts émis - paquets Multicasts émis 	<p>Définition d'études permettant d'avoir une liste d'équipements classée suivant un indicateur de trafic.</p> <p>ex : Les 5 équipements qui ont émis le plus de paquets broadcasts pendant 1 min.</p> <ul style="list-style-type: none"> - Paquets reçus - Paquets émis - Octets reçus - Octets émis - Paquets erreur émis - Broadcasts émis - Multicasts émis - Nombre d'équipements désirés - durée de la mesure 	<p>Confient les Informations de trafic entre deux équipements Ethernet</p> <ul style="list-style-type: none"> - Flux échangé en octets - Flux échangé en paquets - Flux d'erreurs

Architecture du protocole SNMP

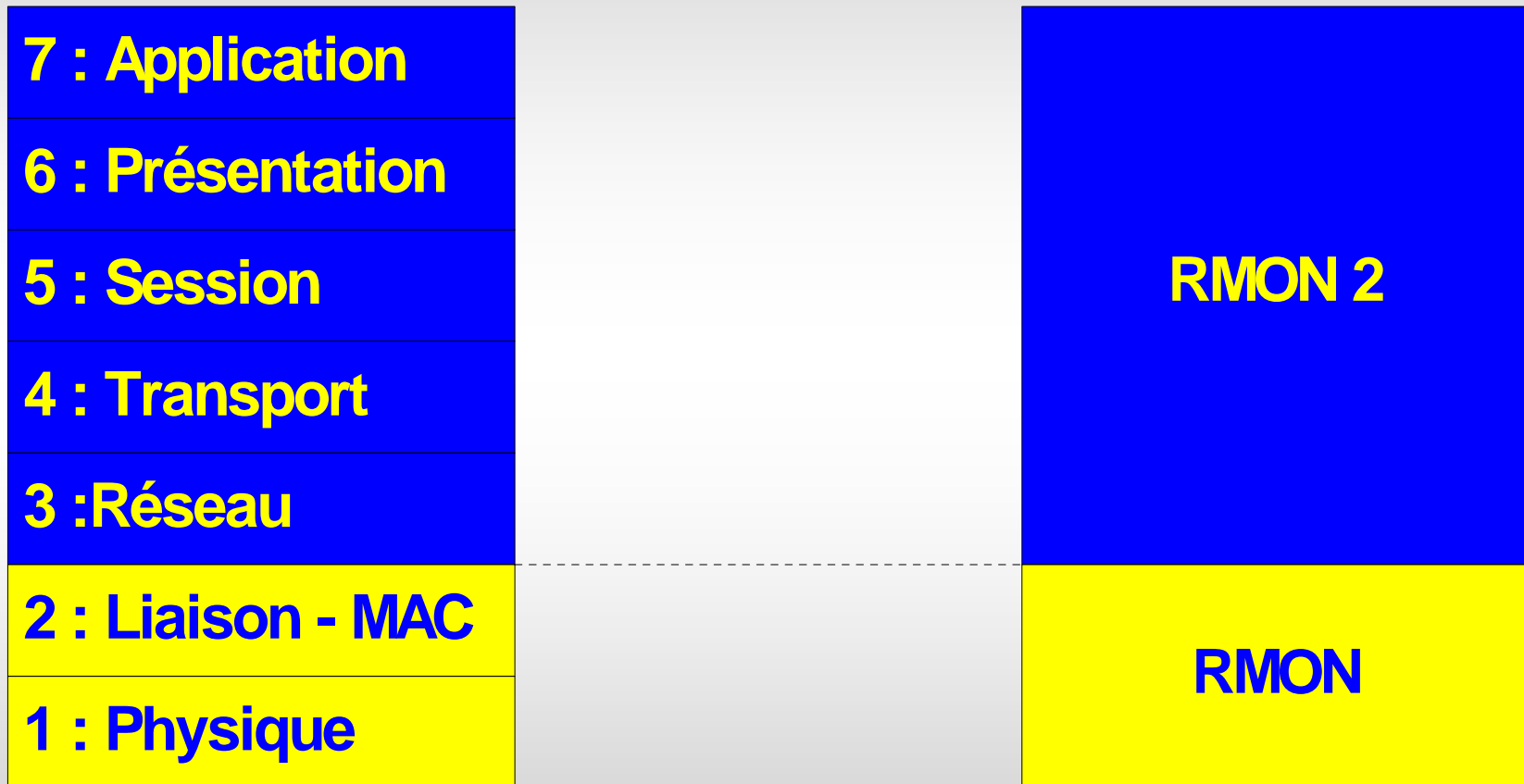
Groupe <i>Filter</i>	Groupe <i>Packet Capture</i>	Groupe <i>Event</i>
<p><i>Définition des filtres sur les captures de paquets</i> ex : filtrage du trafic SNMP</p> <ul style="list-style-type: none"> - Position du filtre dans le paquet - Valeur du filtre - Masque associé au filtre - Masque complémentaire - Masque associé à l'état du paquet - Masque complémentaire - Mode de capture (paquets correspondant au filtre ou paquets complémentaires) - Evénement déclenchant l'ouverture du canal - Evènement déclenchant la fermeture du canal - Nombre de paquets capturés - Evènement généré quand un paquet est capturé 	<p><i>Gestion de l'enregistrement des paquets capturés par le Groupe Filter</i></p> <ul style="list-style-type: none"> - No de canal utilisé - Etat du Buffer (disponible ou plein) - Action quand le Buffer est plein - Nombre d'octets enregistrés pour chaque paquet - Nombre d'octets remontés par SNMPGET - Offset sur les paquets remontés - Taille désirée pour le Buffer - Nombre de Paquets capturés 	<p><i>Définition des actions associées aux alarmes générées.</i></p> <ul style="list-style-type: none"> - communauté des Traps SNMP - Aucune action - Emission d'un Trap SNMP - Enregistrement dans la table Log - Table Log + Emission d'un Trap

La MIB RMON 2

La MIB RMON ne s'adresse qu'aux deux premières couches du modèle ISO (Physique et Ligne), ce qui a pour conséquence qu'une RMON 1 n'analysera que le segment où il se trouve, et cette analyse se fera au niveau MAC (Ligne).

La reconnaissance des protocoles et de son adressage ne pourra se faire dans RMON que si nous lui adjoignons des groupes de MIB qui s'adressent aux couches supérieures du modèle ISO.

Architecture du protocole SNMP



Architecture du protocole SNMP

Création de neuf nouveaux groupes pour dépasser la couche MAC :

- **Protocol Directory** définit les protocoles que la sonde peut analyser.
- **Protocol Distribution** prend les statistiques suivant le protocole.
- **Address Mapping** fait la relation entre l'adresse MAC et l'adresse du protocole (ex: MAC→IP).
- **Network layer Host** mesure globale des trames suivant le protocole (on n'est plus cantonné au segment).
- **Network layer Matrix** mesure entre deux hosts (pas forcément dans le même segment).
- **Application layer Host** nous montons vers les couches hautes de l'OSI pour faire notre analyse.
- **Application layer Matrix** mesure entre deux hosts (suivant le protocole applicatif).
- **History** mémorise les statistiques de niveau 3 en local.
- **Probe Configuration** normalisation de la configuration d'une sonde à partir du Manager.

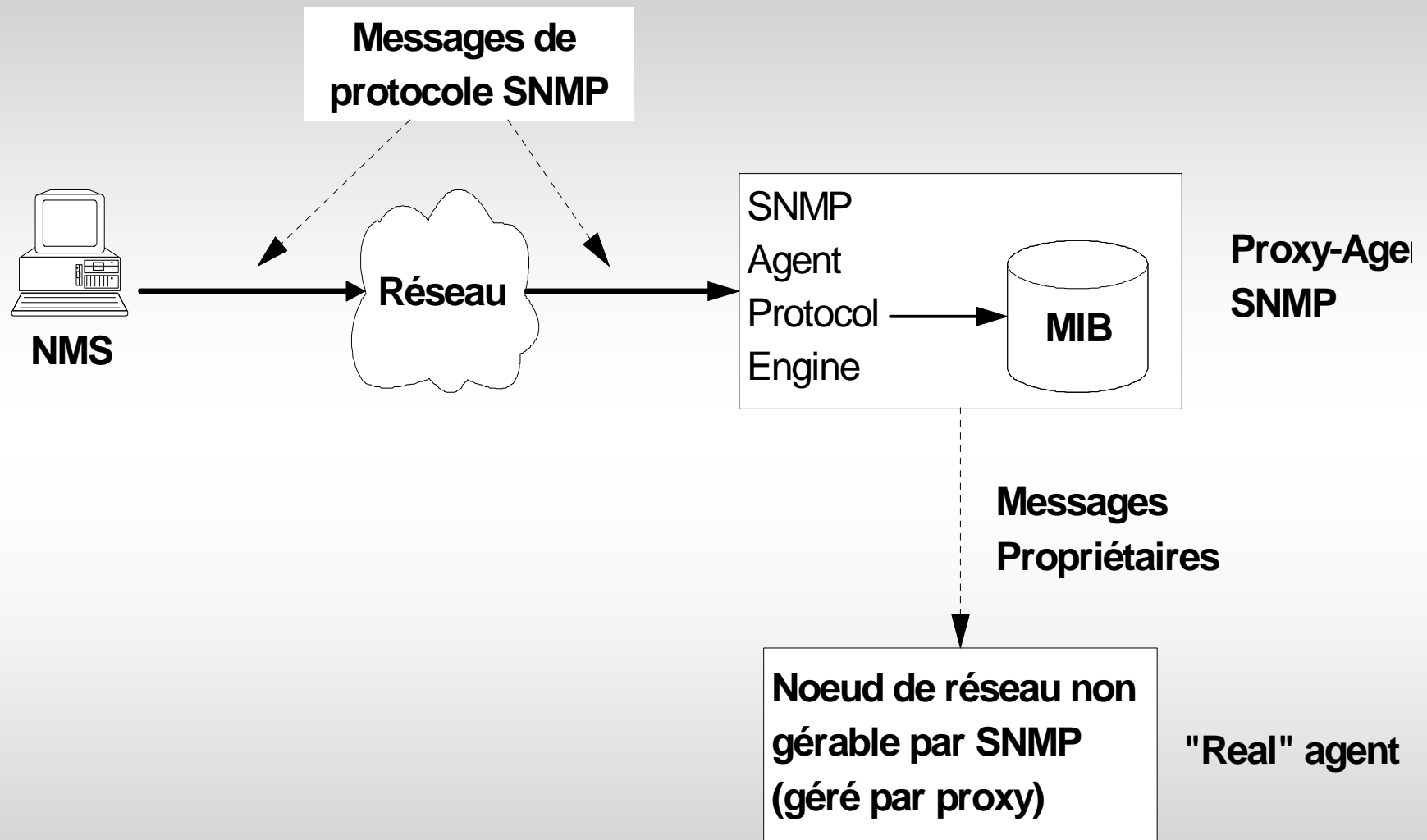
Les PROXIES

Le principe important à retenir de cet Agent RMON évolutif est que l'on donne de l'intelligence à un agent SNMP réputé instrumental à la base. Mais désormais cet agent peut agir éventuellement sans l'aide de son manager et faire une collecte d'informations et une réaction sur cette collecte d'une manière autonome.

Cette solution a donné naissance au principe de proxy-agent ou sous-agent SNMP qui travaillera dans une station de travail sous l'agent SNMP. En fait cela permet de faire de la délégation d'administration

L'agent proxy sert également de passerelle entre une station d'administration SNMP et un agent “ non-SNMP ” qui utilise un protocole propriétaire. Il occupe alors un rôle de traducteur.

Architecture du protocole SNMP

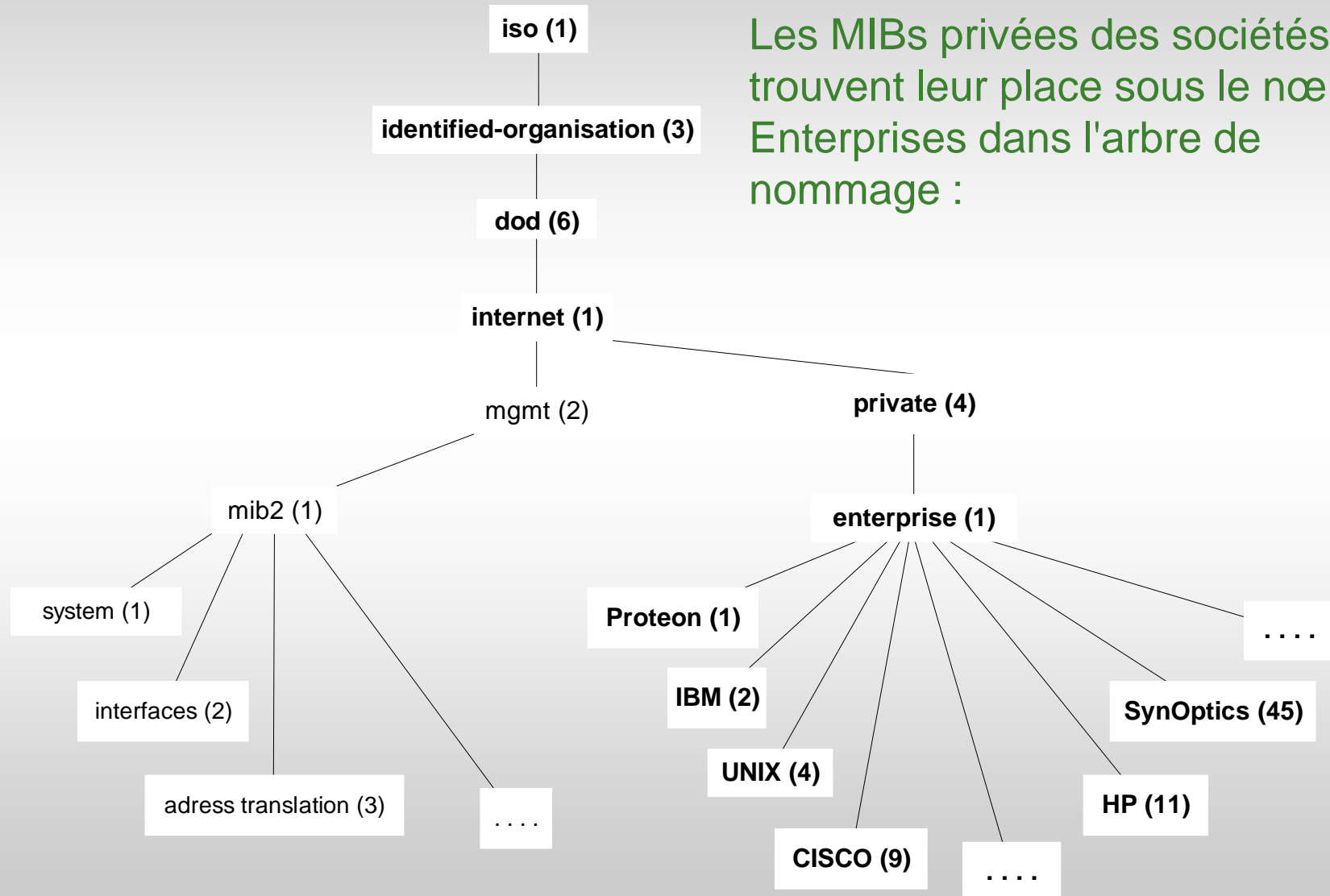


Proxy Management of Agent

Architecture du protocole SNMP

Private MIB

Les MIBs privées des sociétés trouvent leur place sous le nœud Enterprises dans l'arbre de nommage :



Introduction (échange de données)

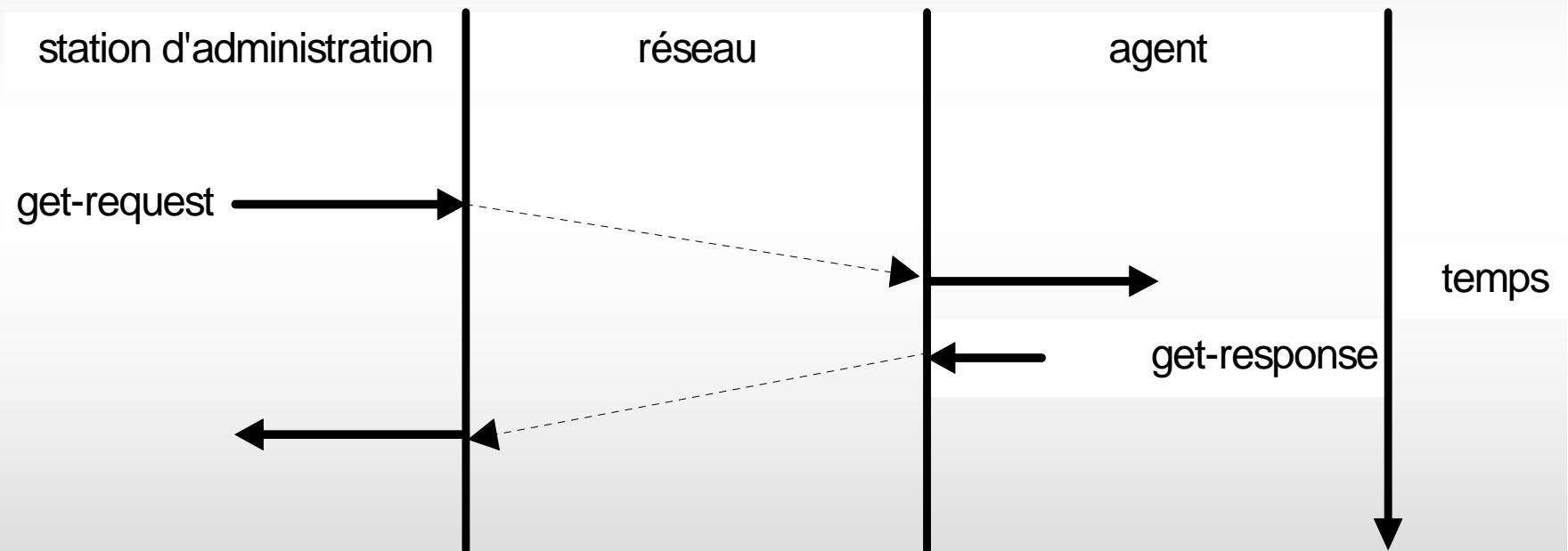
SNMP est un protocole **asynchrone** de requêtes / réponses.

Par conséquent, une entité SNMP n'a pas besoin d'attendre une réponse après avoir envoyé un message.

On distingue quatre types d'opérations au niveau SNMP:

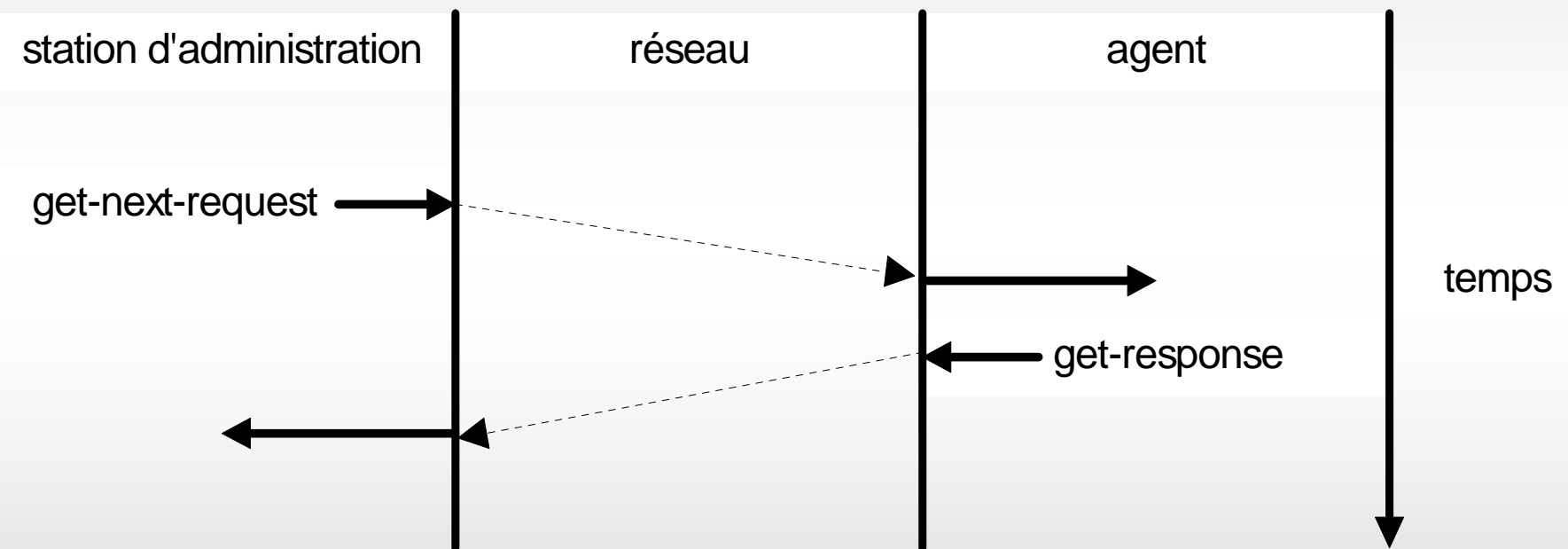
get-request/get-response

La station d'administration interroge l'agent



get-next-request/get-response

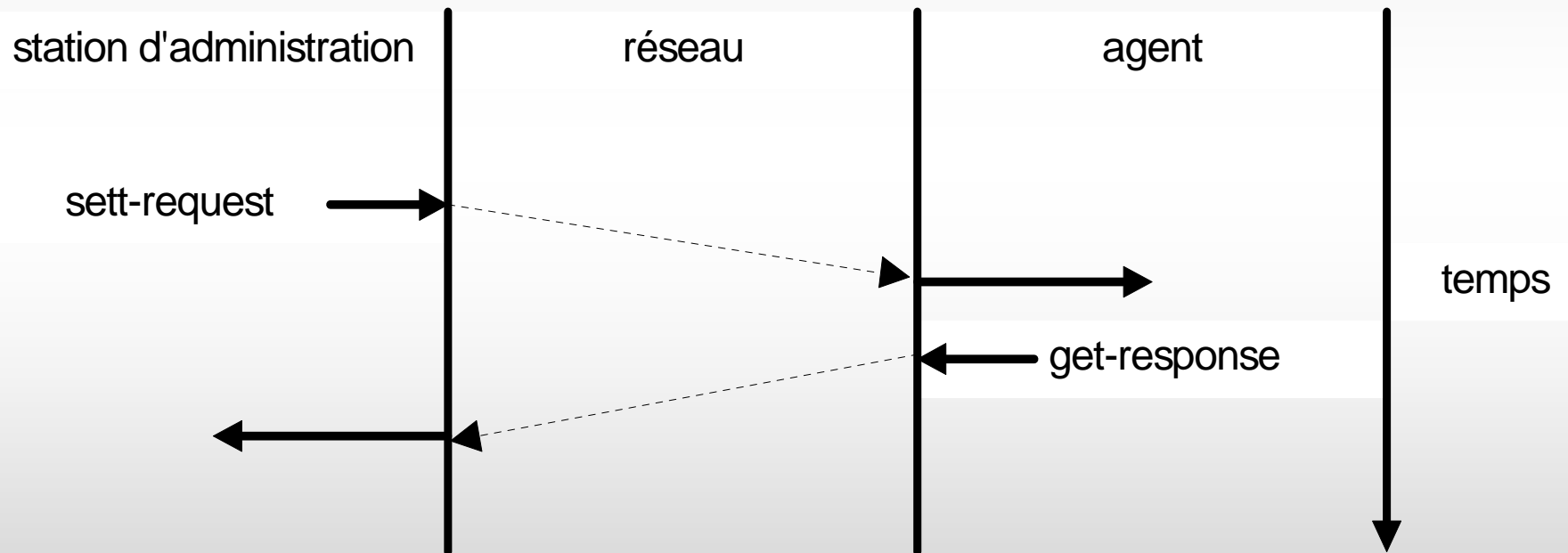
La station d'administration interroge une table de l'agent



Remarque : Cette commande est très lourde car elle implique l'échange de plusieurs **get-request/get-response** successifs.

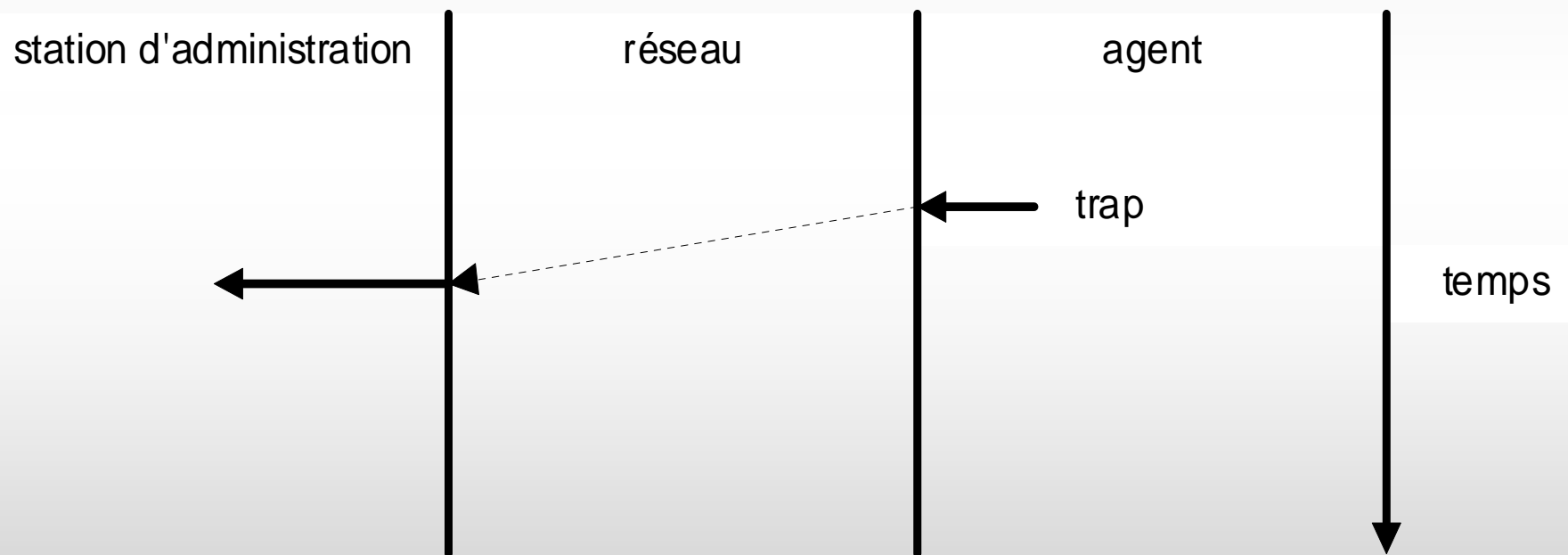
set-request/get-response

La station d'administration enregistre des données au niveau d'un agent.



trap

L'agent envoie un évènement "**extraordinaire**" vers la station d'administration.



Ports

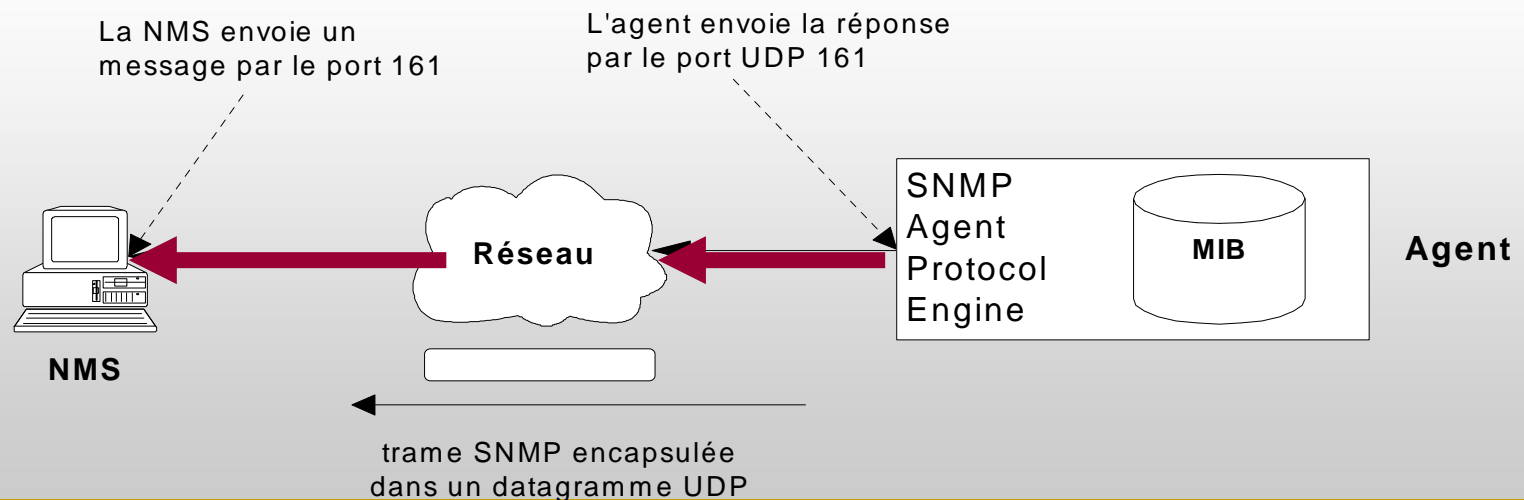
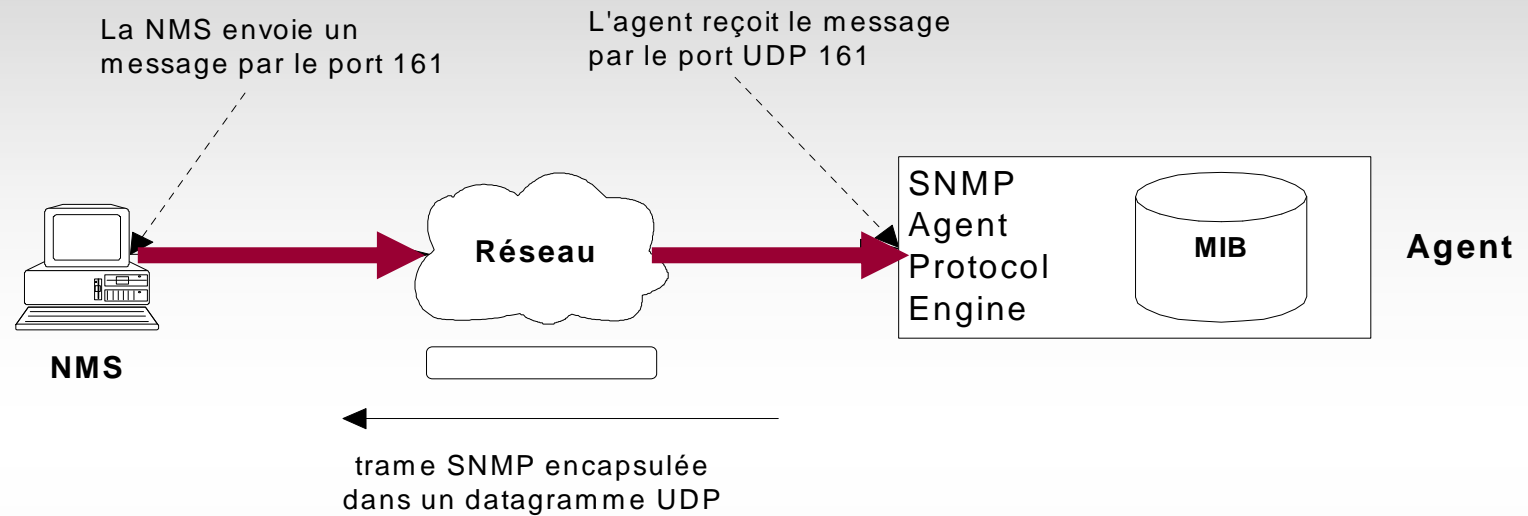
SNMP est un service qui fonctionne au-dessus du protocole de transport UDP, il a donc besoin d'un numéro de port pour communiquer. En fait, on constate que 2 ports lui sont réservés : **161** et **162**.

La station d'administration **émet** (set, get, getnext) par le port 161 en direction de l'agent qui reçoit aussi par le port 161.

L'agent **répond** (response) par le port 161 à la station d'administration qui reçoit cette réponse également par le port 161.

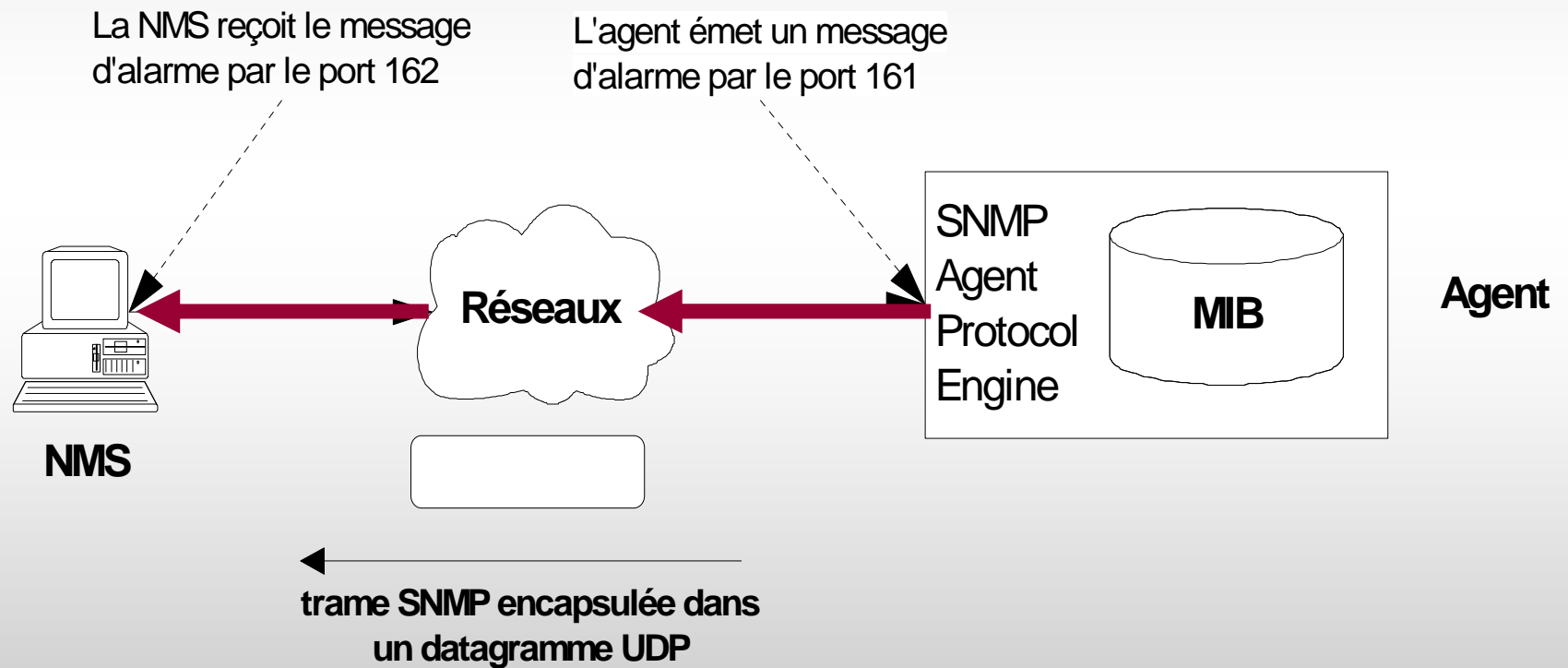
UDP Transport

Normal Send - Response Exchange



Lorsqu'il s'agit d'une alarme (trap), l'agent l'émet par le port 161 mais la station le reçoit par le port 162 :

UDP Transport



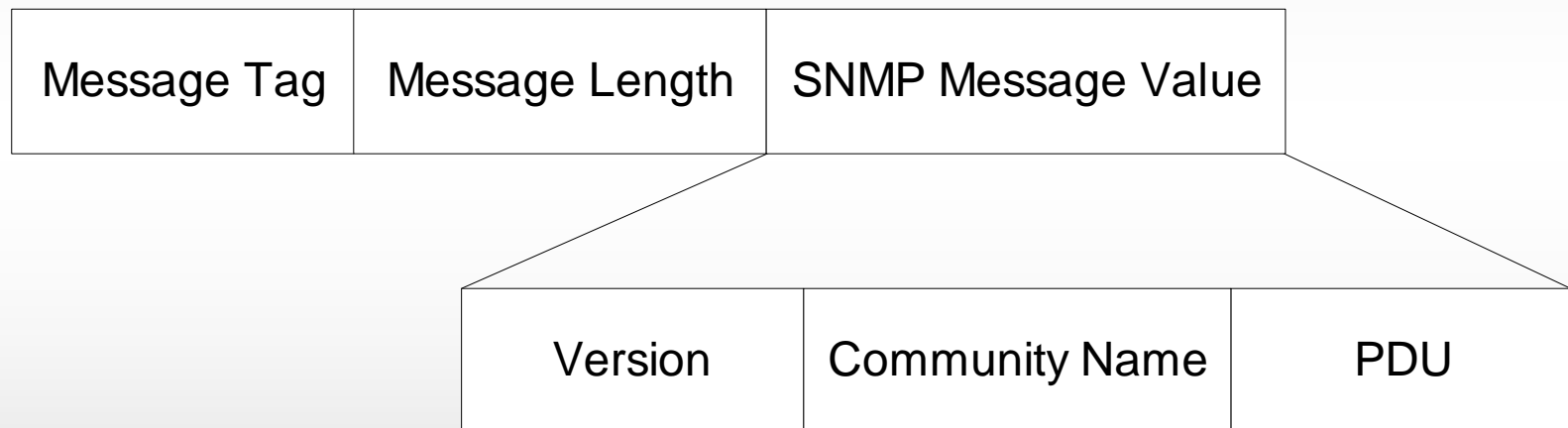
Commandes

La syntaxe abstraite ASN.1 permet de décrire les objets manipulés.
Pour transférer l'information entre deux machines et ce indépendamment de l'architecture des machines, on utilise actuellement un codage : le BER (Basic Encoding Rules).

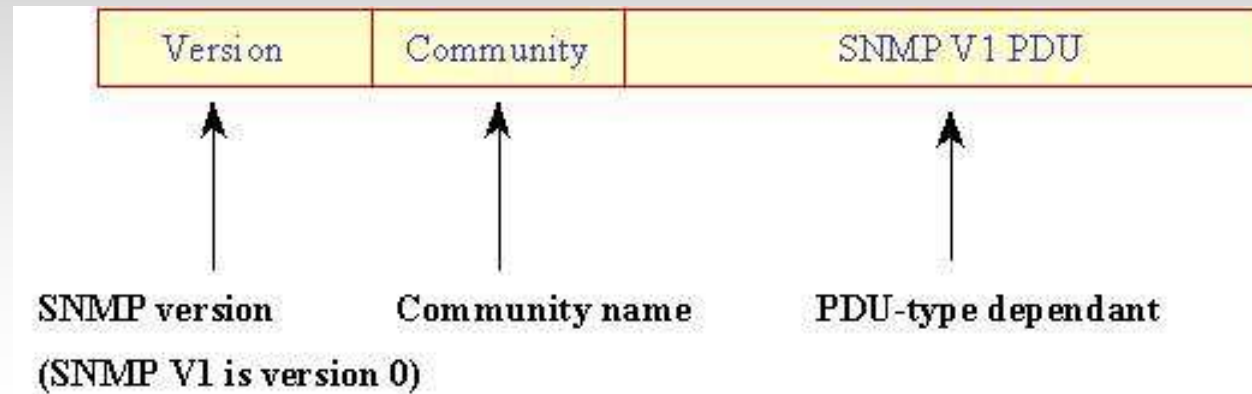
Le BER poursuit le même objectif que le protocole XDR de Sun (eXternal Data Representation), mais il est plus complexe, donc plus gourmand en CPU et n'offre pas de fonction de compactage de l'information.

Trame

La trame SNMP suit le modèle suivant



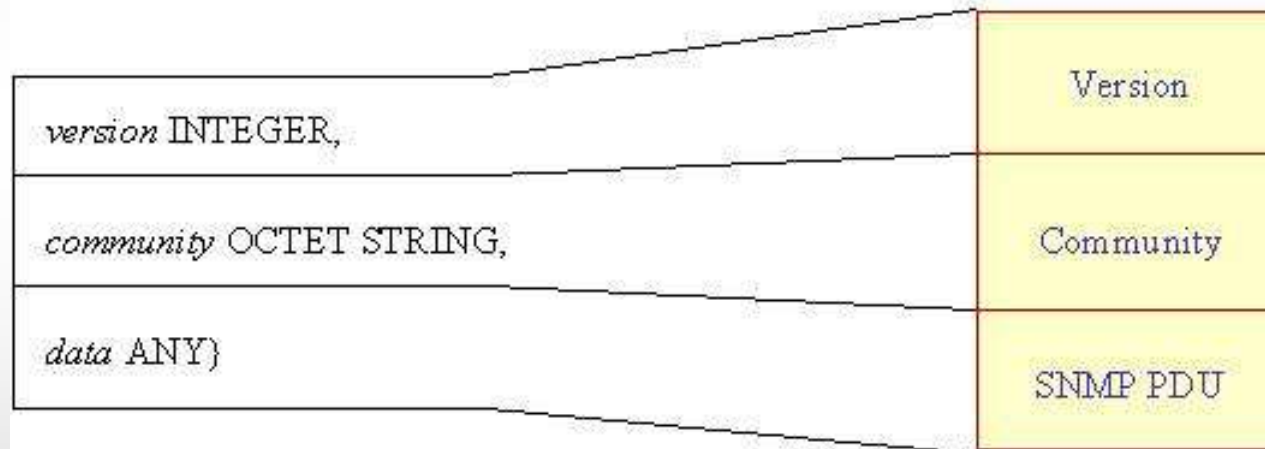
➤ Format des messages SNMP



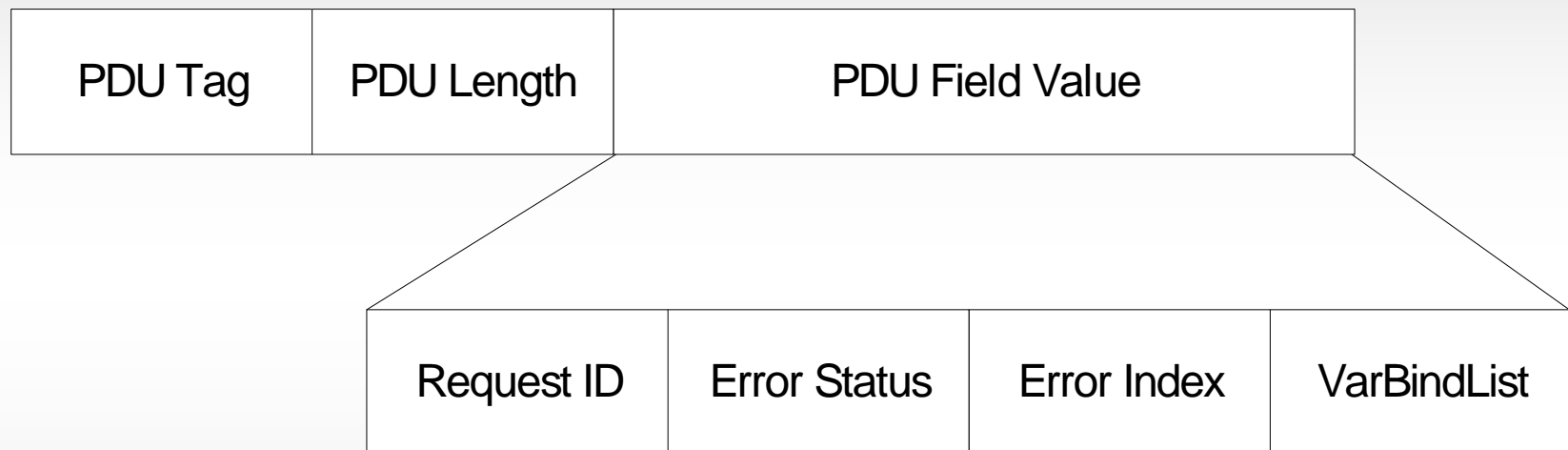
- ✓ Le concept « Community Name » est un concept local défini dans chaque Agent.
- ✓ SNMP community = ensemble de gestionnaires SNMP autorisés à accéder à l'agent
- ✓ Chaque communauté dispose d'un nom unique
- ✓ Chaque gestionnaire doit indiquer la communauté à laquelle il appartient

➤ Définition de messages ASN.1

```
RFC1157-SNMP DEFINITIONS ::= BEGIN  
IMPORTS ObjectName, ObjectSyntax, ... FROM RFC1155-SMI;  
Message ::= SEQUENCE {
```



Le PDU se décompose comme suit :



Description des champs :

Version indique la version de SNMP utilisée. 0 correspond à SNMPv1.

Community Name sert à faire de l'authentification. En fait on peut définir plusieurs groupes qui auront des droits différents sur les objets de la MIB (lecture seule, lecture/écriture). Cette authentification est la seule option de sécurité observée dans SNMPv1. Malheureusement la chaîne de caractère correspondante transite en clair sur le réseau !!!

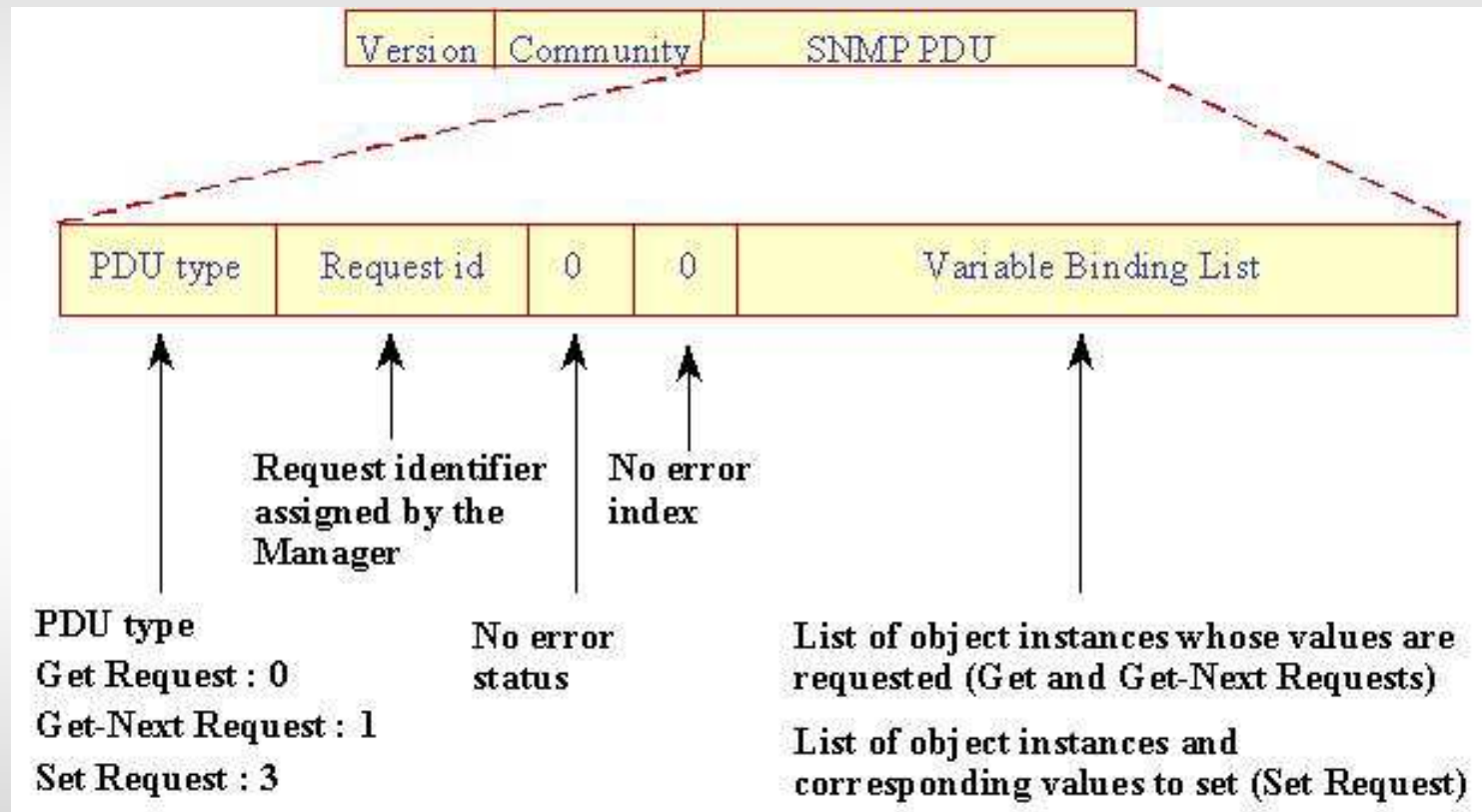
PDU Tag Type du PDU

Request ID Identité du message

Error Status Indication d'erreur

Error Index Pointeur sur l'erreur

➤ Formats des primitives Get, Get-Next et Set.



➤ Définitions des primitives Get, Get-Next et Set.

```
PDU ::= 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

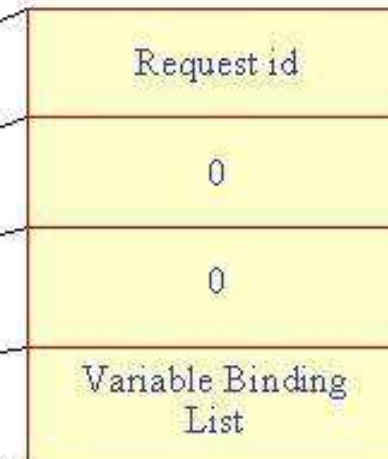
GetNextRequest-PDU ::= [1] IMPLICITE PDU

Response-PDU ::= [2] IMPLICITE PDU

SetRequest-PDU ::= [3] IMPLICITE PDU

PDU ::= SEQUENCE {

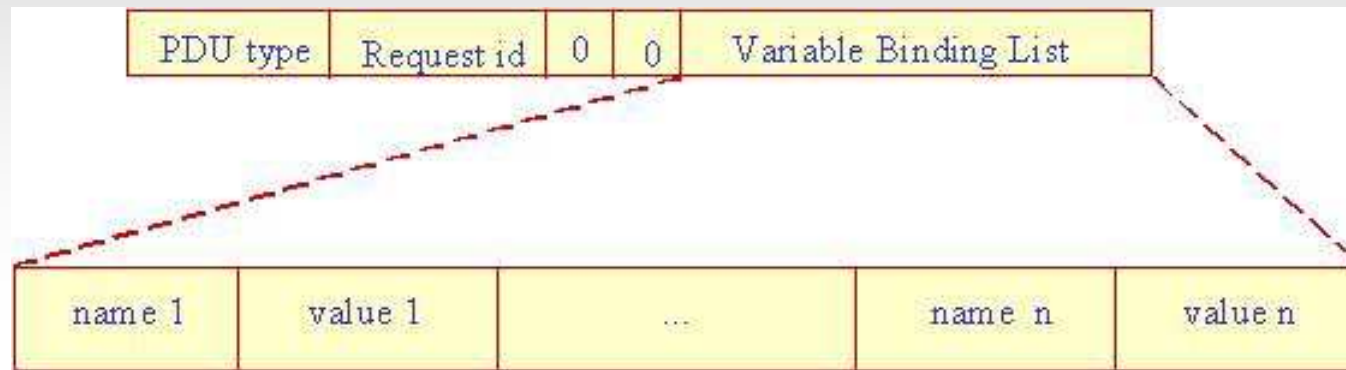
```
    request-id INTEGER,  
    error-status INTEGER,  
    error-index INTEGER,  
    variable-binding VarBindList }
```



L'objectif de la variable « **Binding List** » est

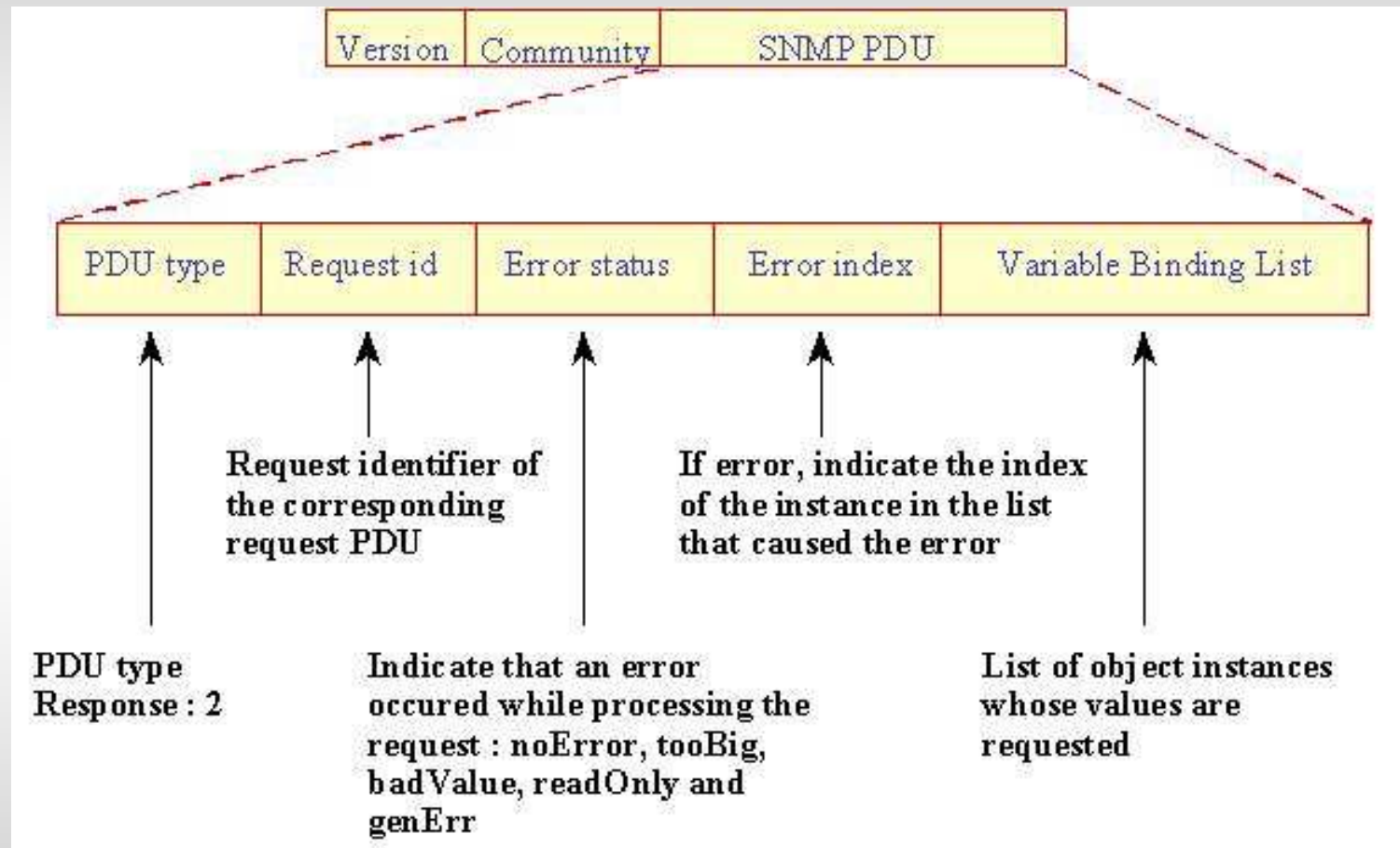
- ✓ Grouper plusieurs opérations de même type
- ✓ Réduction le flux d'information de gestion entre Gestionnaire-Agent

➤ Format de la variable « *Binding List* »



```
VarBind ::= SEQUENCE {  
    name ObjectName,  
    value ObjectSyntax }  
VarBindList ::= SEQUENCE OF VarBind
```


➤ Format de la réponse



Command	Error Status	Error Index	Meaning	Action
GetRequest	noError	0	Command successfully processed	
	noSuchName	Offset of first variable in error	Object does exist is aggregate type Wrong access code	Verify object name and type Verify object is readable
	tooBig	0	Response PDU too large	Shorten VarBindList
	genErr	Offset of first variable in error	Agent instrumentation routine failed	Eliminate variable from VarBindList
GetNextRequest	noError	0	Command successfully processed	None
	noSuchName	Offset of first variable in error	Next object does not exist Wrong access code	Verify object name Verify object is readable
	tooBig	0	Response PDU too large	Shorten VarBindList
	genErr	Offset of first variable in error	Agent instrumentation routine failed	Eliminate variable from VarBindList
SetRequest	noError	0	Command successfully processed	None
	noSuchName	Offset of first variable in error	Object does not exist Is aggregate type Wrong access code	Verify object name and type Verify object is readable
	badValue	Offset of first variable in error	Incorrect ASN.1 type, length or value	Correct ASN.1 encoding of variable
	genErr	Offset of first variable in error	Agent instrumentation routine failed	Eliminate variable from VarBindList
	tooBig	0	Response PDU too large	Shorten VarBindList

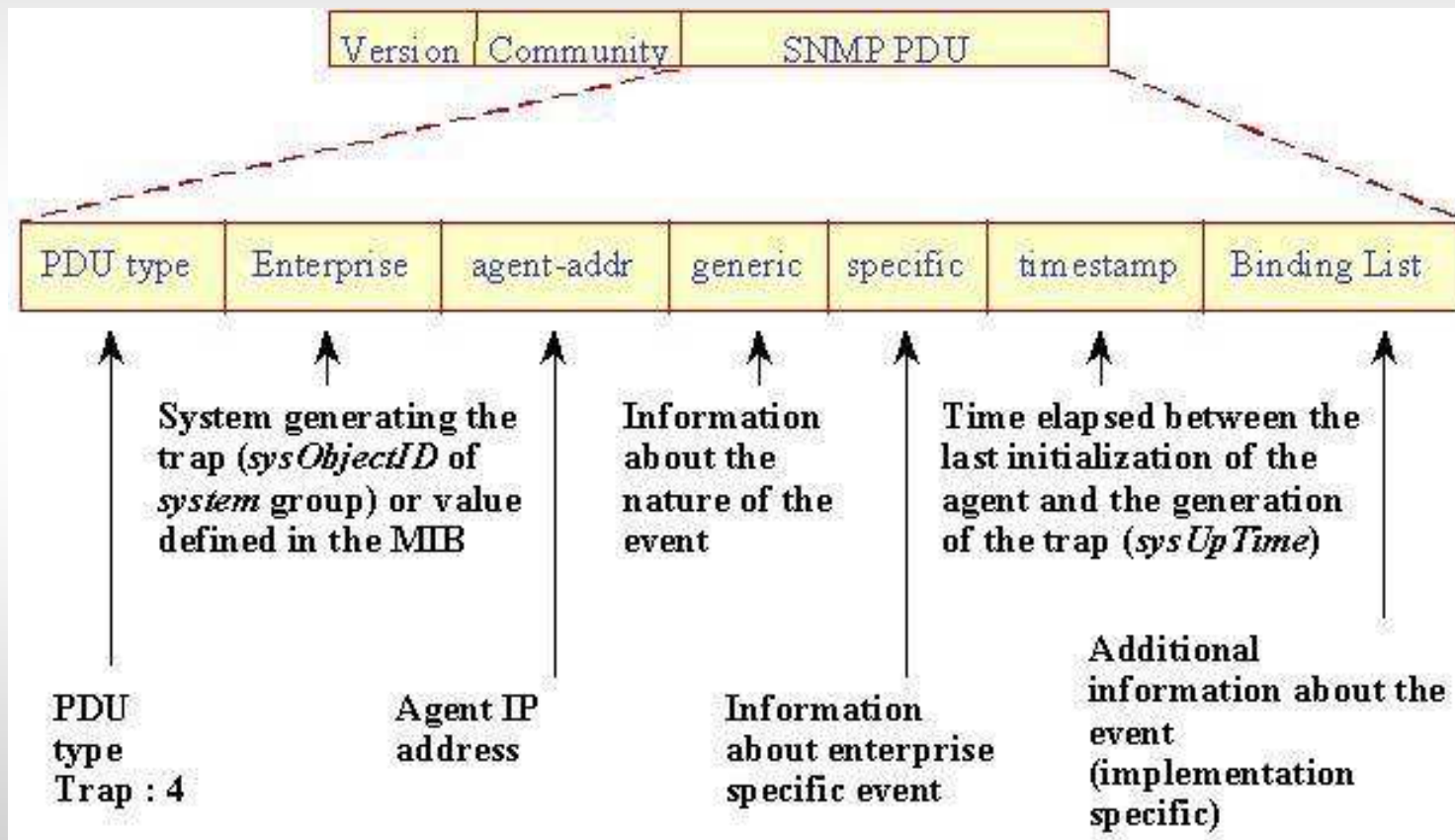
0 : noError,
1 : tooBig,
2 : noSuchName,
3 : badValue,
4 : readOnly,
5 : genError.

Trap

Le mécanisme des traps SNMP permet à l'agent d'envoyer à tout moment un message exceptionnel (une interface qui tombe en panne, etc...)

La trame d'un trap utilise la structure suivante :

➤ Format de la Trap



➤ Définitions d'une Trap ASN.1

```
PDU ::= 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 }
```

Enterprise

agent-addr

generic

specific

timestamp

Variable Binding
List

Entreprise	valeur de l'objet <i>sysObjectId</i> de la MIB de l'agent (notation ASN.1)
Agent Addr	valeur de l'objet <i>Network Address</i> de l'agent (adresse IP)
Generic Trap	cf. tableau des traps SNMP (ci-dessous)
Specific Trap	identifie l'entreprise Specific trap (trap spécifique à l'agent donc non standardisé)
Time Stamp	valeur de l'objet <i>sysUpTime</i> de la MIB de l'agent lorsque l'événement s'est produit
VarBindList	liste de variables contenant des informations sur le trap

Les *deux premiers* champs identifient le matériel qui envoie le message, tandis que les *trois suivants* procurent des informations quant à la nature du problème et l'heure à laquelle il s'est produit ...

Les valeurs du **Generic Trap**:

Nom	Numéros SNMP	Signification
coldStart	0	l'agent se réinitialise et les objets peuvent changer (changement de configuration)
warmStart	1	l'agent se réinitialise mais les objets ne sont pas modifiés (pas de changement de configuration)
linkDown	2	une des interfaces de l'agent est tombée (la première variable dans la liste <i>variable-bindings</i> identifie l'interface)
linkUp	3	une des interfaces de l'agent est à nouveau opérationnelle (la première variable dans la liste <i>variable-bindings</i> identifie l'interface)
authenticationFailure	4	un message SNMP a été reçu d'une entité SNMP et il y a eu un problème d'authentification en fonction du nom de communauté et des droits d'accès qui lui sont accordés.
egpNeighborLoss	5	un EGP peer (EGP = Exterior Gateway Protocol) est tombé (la première variable dans la liste <i>variable-bindings</i> contient l'adresse IP).
entrepriseSpecific	6	certaines événements dépendent de l'agent et ne sont donc pas standardisés, dans ce cas le numéro du trap est donné dans le champ <i>specific-trap</i> .

Avantages

L'avantage majeur dans le fait d'utiliser SNMP est qu'il est de conception **simple**. Le résultat flagrant de cette simplicité est une administration de réseau simple à implémenter et **rapide**.

Un autre avantage de SNMP est qu'il est vraiment **très répandu** aujourd'hui.

L'expansion est un autre avantage de SNMP. De par sa simplicité de conception, il est facile de mettre à jour le protocole pour qu'il réponde aux besoins des utilisateurs futurs. Il est également **modulable** : on n'a pas besoin d'installer les commandes qui nous semblent trop coûteuses.

Enfin, SNMP est basé sur le protocole de transport UDP ce qui nécessite **moins de ressources et de connexions** simultanées qu'avec TCP.

Et enfin, c'est une solution **peu chère**.

Inconvénients

Le premier défaut de SNMP est qu'il contient quelques gros **trous de sécurité** à travers lesquels des intrus peuvent accéder aux informations transitant sur le réseau. (implémenter des mécanismes de sécurité en ce qui concerne le caractère privé des données, l'authentification et le contrôle d'accès).

Puisque SNMP se trouve au dessus de UDP, il n'y a **pas de reprise sur erreur, ni de contrôle de flux**. La requête ou la réponse peut être égarée.

SNMP est un protocole **bavard**. Cette surcharge de trafic n'est pas trop gênante sur un réseau local mais devient embarrassante via le réseau public. (Ce qui rend CMIP plus adapté aux grands réseaux).