Le protocole SNMP

1) Introduction

Parmi les différents protocoles de gestion de réseaux à distance, on trouve le protocole SNMP (Simple Network Management Protocole). Dans ce chapitre nous allons étudier en détail son fonctionnement.

2) Présentation [4]

SNMP (Simple Network Management Protocol) est le protocole de gestion de réseaux proposé par l'IETF. Il est actuellement le protocole le plus utilisé pour la gestion des équipements de réseaux. C'est un protocole relativement simple. Pourtant l'ensemble de ses fonctionnalités est suffisamment puissant pour permettre la gestion des réseaux hétérogènes complexes. Il est aussi utilisé pour la gestion à distance des applications : les bases de données, les serveurs, les logiciels, etc.

3) Les composants de SNMP [4]

L'environnement de gestion SNMP est constitué de plusieurs composants : la station de supervision, les éléments actifs du réseau, les variables MIB et un protocole.

Les éléments actifs du réseau sont les équipements ou les logiciels que l'on cherche à gérer. Cela va d'une station de travail à un concentrateur, un routeur, un pont, etc. Chaque élément du réseau dispose d'une entité dite agent qui répond aux requêtes de la station de supervision. Les agents sont des modules qui résident dans les éléments réseau. Ils vont chercher l'information de gestion comme par exemple le nombre de paquets reçus ou transmis.

- ✓ **La station de supervision** (appelée aussi manager) exécute les applications de gestion qui contrôlent les éléments réseaux. Physiquement, la station est un poste de travail.
- ✓ **La MIB :** (Management Information Base) est une collection d'objets résidant dans une base d'information virtuelle. Ces collections d'objets sont définies dans des modules MIB spécifiques.
- ✓ **Le protocole** : qui permet à la station de supervision d'aller chercher les informations sur les éléments de réseaux et de recevoir des alertes provenant de ces mêmes éléments.

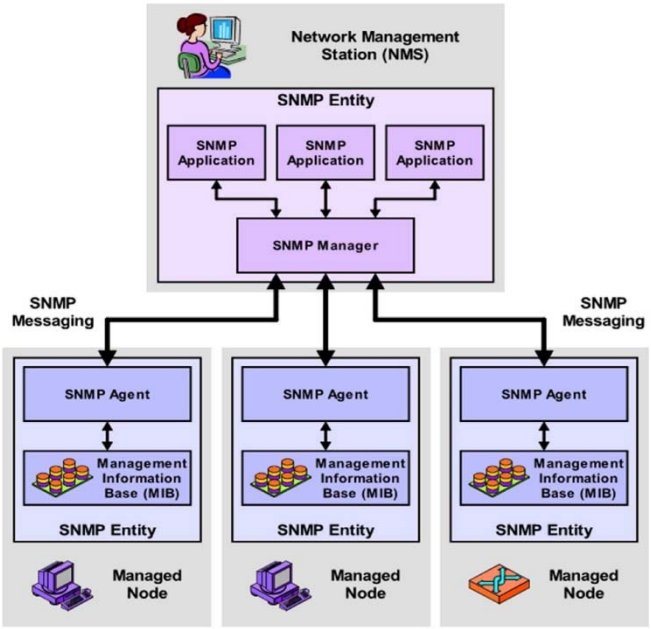


Figure 6 : Architecture SNMP (Kozierok, 2005)

4) Fonctionnement

Le protocole SNMP est basé sur un fonctionnement asymétrique. Il est constitué d'un ensemble de requêtes, de réponses et d'un nombre limité d'alertes. Le manager envoie des requêtes à l'agent, lequel retourne des réponses. Lorsqu'un événement anormal surgit sur l'élément réseau, l'agent envoie une alerte (trap) ou notifications au manager.
SNMP utilise le protocole UDP [RFC 768]. Le port 161 est utilisé par l'agent pour recevoir les requêtes de la station de gestion. Le port 162 est réservé pour la station de gestion pour recevoir les alertes des agents.

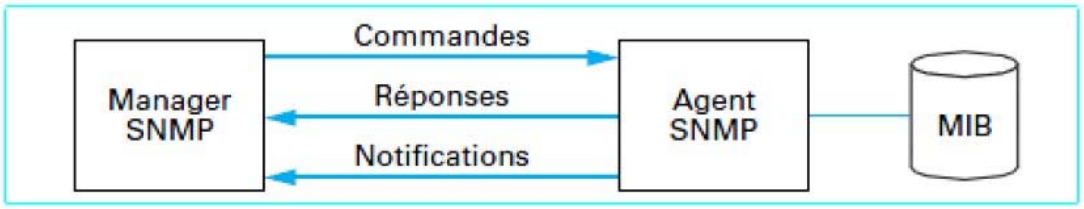


Figure 2.3 : Modèle Agent/Manager

5) Les commandes SNMP

5.1) Les requêtes SNMP

Il existe quatre grands types de requêtes : GetRequest, GetNextRequest, GetBulk, SetRequest.

- ✓ **GetRequest :** permet la recherche d'une variable sur un agent.
- ✓ **GetNextRequest :** permet la recherche de la variable suivante.
- ✓ **GetBulk :** permet la recherche d'un ensemble de variables regroupées.
- ✓ **SetRequest :** permet de changer la valeur d'une variable sur un agent.

5.2 Les réponses SNMP

À la suite de requêtes, l'agent répond toujours par GetResponse. Toutefois si la variable demandée n'est pas disponible, le GetResponse sera accompagné d'une erreur noSuchObject.

Get Response : est générée par une entité du protocole uniquement sur réception de la GetRequest-PDU, GetNextRequest-PDU.
Dès la réception de la GetResponse-PDU, l'entité du protocole de réception présente son contenu à son entité d'application SNMP.

5.3) Les alertes (Traps, Notifications)

Les alertes sont envoyées quand un événement non attendu se produit sur l'agent. Celui-ci en informe la station de supervision via une trap. Les alertes possibles sont : Cold Start, Warm Start, Link Down, Link Up, AuthentificationFailure.

- ✓ **Cold Start :** piège signifie que l'entité du protocole d'envoie se réinitialise tels que la configuration de l'Agent ou de l'entité mise en oeuvre peut être modifiée.
- ✓ **Warm Start :** piège signifie que l'entité du protocole d'envoie se réinitialise de telle sorte que ni la configuration de l'Agent ni l'entité mise en oeuvre est modifiée.

- ✓ **Link Down :** piège signifie que l'entité de protocole d'envoi reconnaît une défaillance dans l'un des liens de communication représentés dans la configuration de l'agent. Le piège-PDU de type Link Down contient, comme le premier élément de son variablebindings, le nom et la valeur de l'instance ifIndex pour l'Interface affectée.
- ✓ **Link Up :** piège signifie que l'entité du protocole d'envoi reconnaît que l'un des liens de communication représentés dans la configuration de l'agent est venu. Le piège-PDU de type Link Up contient, comme le premier élément de sa variablebindings, le nom et la valeur de l'instance ifIndex pour l'Interface affectée.
- ✓ **Authentification Failure** : piège signifie que le protocole d'envoi d'entité est le destinataire d'un message du protocole qui ne sont pas correctement authentifié. Alors que les mises en oeuvre du protocole SNMP doivent être capables de générer ce piège, ils doivent aussi être capables de supprimer l'émission de ces pièges par un mécanisme spécifique de mise en oeuvre.

6) Format du PDU (protocol data unit) de SNMP

6.1) Le format SNMPv1

La version 1 du protocole SNMP utilisée par les administrateurs réseaux ne comporte aucun processus d'authentification et de sécurité hormis le champ « Community Name » qui passe malheureusement en clair sur le réseau.

Description des champs :

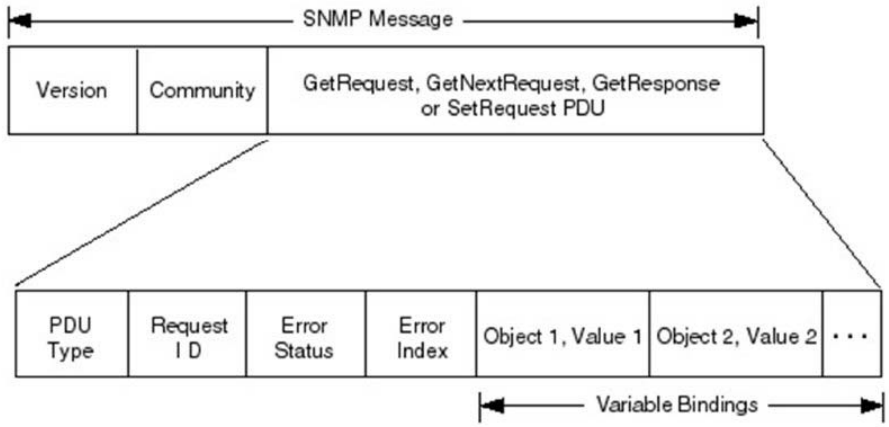


Figure 15 : Format des messages SNMPv1 (MILLER, 1997)

- ✓ **Version** : Version de SNMP.
- ✓ **Community** : 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)
- ✓ **Request-id** : Utilisé pour différencier les messages.
- ✓ **Error-status** : Utilisé pour signaler une erreur (0 si pas d'erreur).

- ✓ **Error-index** : Indique la sous-catégorie d'erreur.
- ✓ **Variable bindings** : représente la liste des variables, avec pour chacune un OID et une valeur (ou un code d'erreur dans le cas de SNMPv2).

6.2) Le format d'un trap

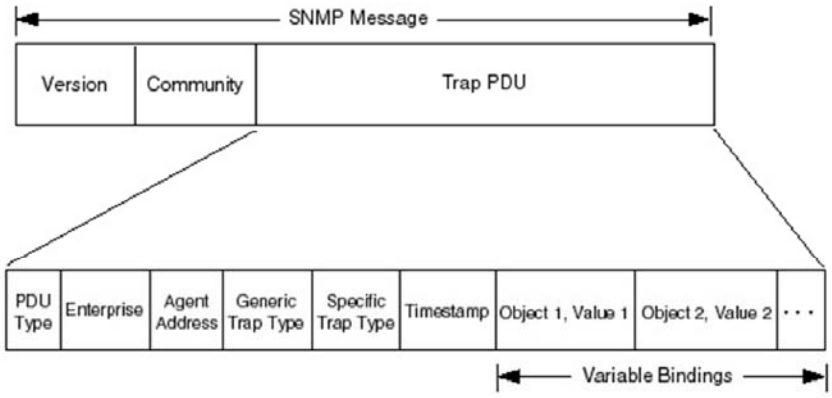


Figure 16 : Format des messages SNMPv1 de type Trap (MILLER, 1997)

- ✓ **Variablebindings** : Nom des variables avec leurs valeurs. Rq : Lors d'une opération **Get**, les valeurs sont NULL.
- ✓ **Enterprise** : Type de l'objet générant l'alarme.
- ✓ **Agent-addr** : Adresse de l'émetteur de l'alarme.
- ✓ **Generic-trap** : Identificateur de l'alarme.
- ✓ **Specific-trap** : Identificateur d'alarme spécifique.
- ✓ **Time-stamp** : Temps écoulé depuis la dernière réinitialisation de l'entité.

Format du Specific-trap

Nom	Numero 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 non opérationnelle (la première variable dans la liste variable bindings identifie l'interface).
linkUp	3	Une des interfaces de l'agent est à nouveau opérationnelle (la première variable dans la liste variable bindings identifie l'interfaces).
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 sont accordés.
egpNeighborLoss	5	Un EGP peer (EGP=Exterior Gatway Protocol) est tombé (la première variable dans la liste variable-bindings contient l'adresse IP).
entrepriseSpecific	6	certains évènements dépendant de l'agent et ne sont donc pas standardisés, Dans ce cas le numéro du trap est donné dans le champ specific-trap.

Tableau IV.3 : la liste des « Trap » envoyé par l'agent.

6.2) Le format SNMPv2

Il existe plusieurs variantes de la version 2, mais aucune d'entre elles n'a été officialisée. Cette version reste principalement expérimentale et rajoute notamment de la sécurité avec SNMPsec. Cette version a laissé place à la version 3 de SNMP.

6.3) Le format SNMPV3

Cette nouvelle version du protocole SNMP vise essentiellement à inclure la sécurité des transactions. La sécurité comprend l'identification des parties qui communiquent et l'assurance que la conversation soit privée, même si elle passe par un réseau public.

L'intégrité a pour rôle d'assurer que le paquet reste inchangé pendant la transmission tandis que l'authentification garantit l'identité de l'émetteur.
Pour construire ce mécanisme, on doit avoir connaissance des fonctions de hachage à une seule direction. Des exemples de ces fonctions sont : MD5 et SHA.

Afin d'effectuer des communications sécurisées, SNMPv3 se base sur DES (Data Encryption Standard) ou AES puisqu'il permet de crypter les messages SNMP pour assurer la privatisation des opérations.
Le cryptage a pour but d'empêcher qu'un intrus n'obtienne les informations de gestion contenues dans un paquet SNMPv3 en écoutant les requêtes et les réponses qui circulent sur le réseau.
Avec SNMPv3, le cryptage de base se fait sur un mot de passe « partagé » entre le manager et l'agent. Ce mot de passe ne doit être connu par personne d'autre. Pour des raisons de sécurité, **deux mots de passe sont utilisés : un pour l'authentification et un autre pour le cryptage (privatisation).**

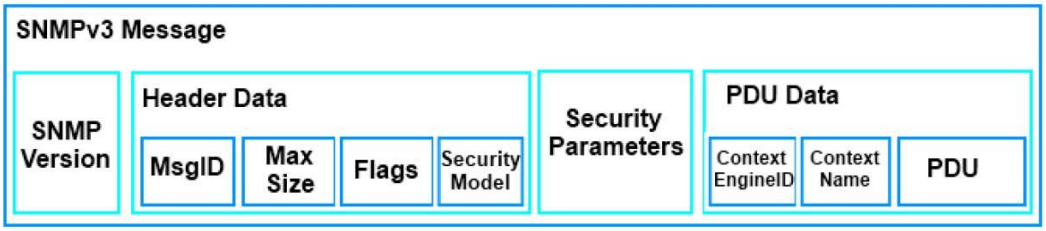
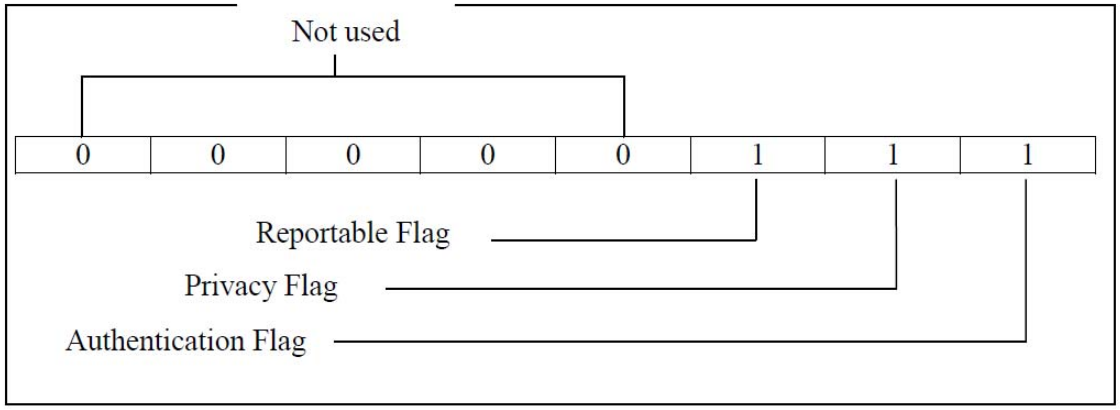


Figure 2.12 : Description du paquet SNMPv3

- ✓ **Version SNMP :** Pour SNMPv3, on place la valeur 3 dans ce champ.
- ✓ **Identificateur de message (MsgId):** Utilisé pour associer les requêtes et les réponses, on retrouve souvent des algorithmes, où le premier message de requête est envoyé avec un nombre aléatoire et les suivants avec les incréments de 1. Les paquets qui sont émis en réponse à une requête portent la même identification que le paquet de la requête.
- ✓ **Grandeur maximale du paquet réponse (Max Size):** Le moteur choisit la taille maximale d'une réponse à une requête selon ses capacités en mémoire tampon et ses limites à décoder de longs paquets. Quand on envoie une réponse à une requête, on doit veiller à ne pas dépasser la taille maximale.
- ✓ **Drapeaux(Flags) :** indiquant si une réponse est attendue et si un modèle de sécurité a été utilisé.



Trois bits sont utilisés (sur les 8) pour indiquer :

- Si une réponse est attendue à la réception de ce paquet. (Reportable Flag).
- Si un modèle de privatisation a été utilisé (Privacy Flag).
- Si un modèle d'authentification a été utilisé (Authentication Flag).

7) L'exploitation des données le MIB (Management Information Base)

C'est une collection d'objets décrits formellement, dont chacun représente un type particulier d'information. Objets MIB accessibles et peuvent être gérés avec le Simple Network Management Protocol (SNMP) grâce à un système de gestion de réseau. Cette collection d'objets contient des informations requises par un système de gestion et l'information est stockée sous la forme d'un ensemble de variables MIB. La MIB est une base d'informations de gestion. Elle comprend des informations à consulter, des paramètres à modifier, ainsi que des alarmes à émettre. La MIB est structurée sous une forme arborescente. Chaque objet de la MIB est identifié par un nom et un **OID**. Le chemin suivi pour aller de la racine à l'objet constitue l'OID de celui-ci. Les OIDs permettent de parcourir la MIB jusqu'à atteindre la variable souhaitée pour en lire ou modifier les attributs.
Ce sont des identifiants universels permettant d'assurer l'interopérabilité entre différents logiciels indépendamment du matériel.
Les objets administrés (aussi bien les objets de type scalaire que tabulaire) sont composés d'une ou plusieurs instances d'objets, lesquelles sont essentiellement des variables. Les objets scalaires définissent une seule instance d'objets alors que les objets tabulaires définissent plusieurs instances liées d'objets, et celles-ci sont regroupées dans des tables MIB. Chaque niveau de l'arborescence est repéré par un index numérique.

8) Structure du MIB

Chaque MIB est propre à l'agent. Il y a donc une MIB pour chaque équipement supervisé. Cette MIB contient des :

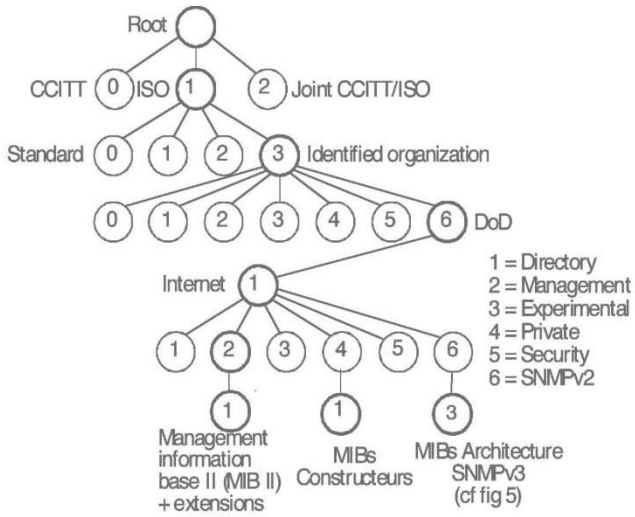
• informations à consulter.
• paramètres à modifier.
• alarmes à émettre.

La MIB est organisé hiérarchiquement avec une structure arborescente, de la même façon que les domaines d'Internet.

Chaque nœud d'un arbre représente un objet. Cet objet est défini avec un OID (Object Identifier). Cet identifiant est constitué d'une suite de chiffre séparé par des points.
Pour qu'un client accède à ces objets, il faut qu'il en connaisse l'existence. Une MIB contient un ensemble d'informations standards ; c'est la MIB standard.
Or pour la plupart des éléments réseaux, on rajoute un certain nombre d'objets propre à un agent pour en exploiter les possibilités : c'est la MIB privée. Par exemple, dans la MIB standard, nous retrouvons certains compteurs, notamment ceux mesurant la performance réseau.

La MIB est un fichier texte écrit en langage ASN 1(Abstract Syntax Notation 1). Il contient la spécification de différents OID. Pour chaque OID, cette spécification comprend :

Un OID unique, dans l'arbre des OID ; par exemple 1.3.6.1.2.1.1.3.



Espace de nommage de l'ISO

Branchement réservé à SNMP : 1.3.6.1

Les chemins les plus utilisés sont :

1.3.6.1.2.1….Ensemble des MIB du standard (MIB-2, extensions…) développées par l'IETF

1.3.6.1.4.1…. espace des MIBs constructeurs

1.3.6.1.6.3…. snmpv2, en fait contient les MIBs associées aux sous systèmes de l'architecture SNMPv3 (framework, USM, Coex, VACM…)

Figure 2

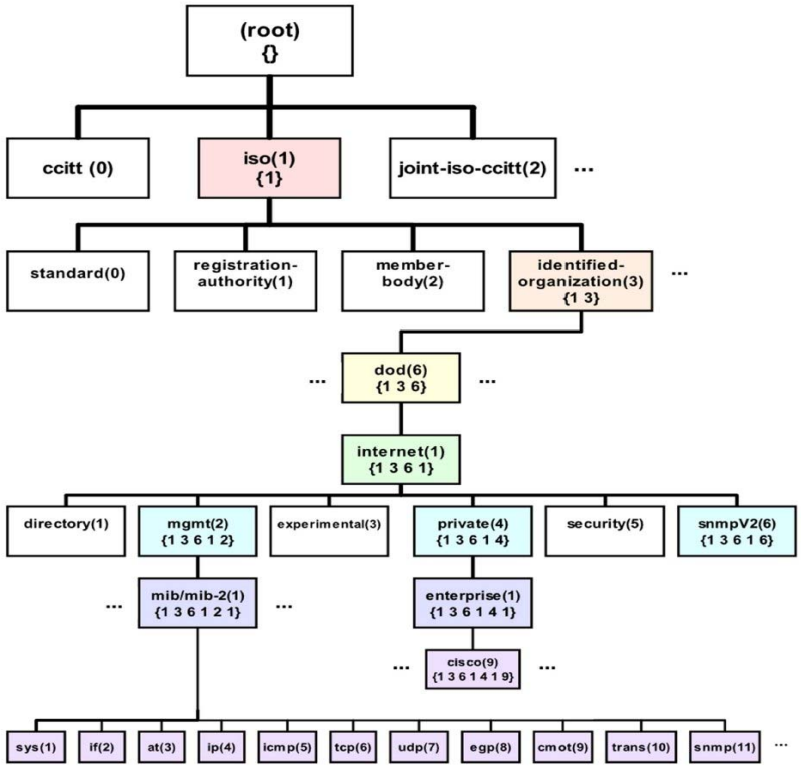


Figure 10 : Arborescence de la MIB II (Kozierok, 2005)

Les types simples : le tableau suivant illustre les types simples utilisés par SNMP.

TYPE	DESCRIPTION
INTEGER	Entier de 32 bits
OCTET STRING	Chaîne d'octets
NULL	Vide
OBJECT IDENTIFIER	Séquence d'entiers identifiant un objet à l'aide de son positionnement à partir de la racine de la MIB (ex. : 1.3.6.1)
ENUMERATION OF INTEGER	Permet de représenter une énumération de valeurs par des entiers non nuls (exemple : on(1), off(2))

Tableau 2.2 : Types simples d'ASN1

Type SMIv1	Type SMIv2	Description
IpAddress	IpAddress	Représente l'adresse IP 32 bits par un OCTET STRING de longueur 4.
Counter	Counter32	Représente un entier non nul dont la valeur croît jusqu'à sa valeur maximum puis repasse à zéro (ex : compteur de trames émises sur une interface). Représenté sur 32 bits.
Gauge	Gauge32	Jauge : entier positif dont la valeur peut croître ou diminuer (ex. : % de CPU).
TimeTicks		Permet de compter une durée en centièmes de seconde à partir d'un temps origine.
Opaque	(obsolète en v2)	Permettent de passer une syntaxe ASN.1 arbitraire sous forme d'un OCTET STRING.
	BITS	Ensemble de bits nommés.

Tableau 2. 4 : Types simples SMI pour les objets MIB