MINISTERE DE L'ENSEIGNEMENT SUPERIEUR, DE LA RECHERCHE SCIENTIFIQUE ET DE LA TECHNOLOGIE
DIRECTION DES INSTITUTS SUPERIEURS DES ETUDES TECHNOLOGIQUES ISET MAHDIA

**DEPARTEMENT TECHNOLOGIES DE L'INFORMATIQUE**
**POUR LES ETUDIANTS INFORMATIQUE 4EME NIVEAU:**
**SPECIALITE SYSTEMES ET RESEAUX**

# Support de cours
# Administration des Réseaux

## Partie 1 : Le protocole SNMP

Elaboré par :

Noureddine GRASSA

Année 2007- 2008

# SOMMAIRE

# PLAN DU COURS ADMINISTRATION D'UN PARC INFORMATIQUE

## 1. SITUATION DU COURS DANS LE PROGRAMME

Le cours *Administration systèmes et réseaux* est destiné aux étudiants de l'ISET niveau 4, profil Réseaux Informatiques. Ce cours nécessite un pré-requis en **introduction réseaux**, **Unix** et le **langage Shell** et a comme volume horaire 22,5 heures de cours intégrés et 45 heures de travaux pratiques.

Le plan de ce cours englobe les objectifs généraux à réaliser et les objectifs spécifiques à atteindre à travers ce cours, ainsi que les méthodologies adoptées pour le présenter et les moyens à utiliser pour évaluer les performances.

Dans ce qui suit, on présentera, le but, les objectifs généraux, et une décomposition des objectifs généraux en objectifs spécifiques.

## 2. BUT DU COURS ET OBJECTIFS GÉNÉRAUX

Buts du cours

**Partie 1 :**

- Comprendre le protocole SNMP et son fonctionnement
- Mettre en place un outil de supervision utilisant le protocole SNMP sous Windows et sous Linux
- Savoir modifier dans les fichiers de configuration
- Automatiser des tâches avec des scripts Shell et batch

**Partie 2 :**

- Gestion et analyse du service DHCP
- Gestion et analyse du système DNS
- Gestion et analyse du service d'authentification : NIS et SAMBA
- Gestion des annuaires : LDAP et Active Directory

| Objectifs généraux | Conditions de réalisation de la performance | Critères d'évaluation de la performance |
|---|---|---|
| Comprendre le protocole SNMP ainsi que son fonctionnement | A partir des notes de cours et des références bibliographiques, l'étudiant devrait comprendre ce que fait le protocole SNMP et son emplacement dans le modèle TCP/IP et maitriser au moins un outil de supervision | Aucun concept ne doit être omis. |
| Mise en place d'un service DHCP | A partir des notes de cours, des références bibliographiques, TD et TP l'étudiant doit être en mesure d'installer et configurer correctement un serveur DHCP sous Linux et Windows | Aucune erreur n'est permise. |
| Gestion et analyse du système DNS. | A partir des notes de cours, des références bibliographiques, TD et TP l'étudiant doit être en mesure d'installer un serveur DNS sous Linux (packages +dépendances). Comprendre le fonctionnement du DNS et configurer correctement les différents fichiers associés. | Aucune erreur n'est permise. |
| Mise en place d'un serveur NIS. | A partir des notes de cours, des références bibliographiques, TD et TP l'étudiant doit être en mesure d'installer et configurer un serveur NIS. Il doit faire appel à ses connaissances sous Linux et avoir les pré requis nécessaire pour réussir cette tâche | Aucune erreur n'est permise. |
| Mise en place d'un serveur SAMBA | A partir des notes de cours, des références bibliographiques, TD et TP, l'étudiant doit être en mesure d'installer et configurer un serveur SAMBA pour partager les fichiers, dossiers et imprimante entre Linux et Windows. | Aucune erreur n'est permise. |
| Mise en place d'un annuaire LDAP | A partir des notes de cours, des références bibliographiques, TD et TP, l'étudiant doit être en mesure d'installer et configurer un serveur openLDAP, de connaitre la structure d'un serveur LDAP et d'ajouter des utilisateurs et groupes dans l'annuaire | 90% des exercices devront être réussis. |
| Mise en place d'un contrôleur de domaine Active Directory | A partir des notes de cours, des références bibliographiques, TD et TP, l'étudiant doit être en mesure d'installer et configurer un serveur Active Directory, créer des utilisateurs et groupes et d'ajouter les politiques de sécurité pour les utilisateurs | 90% des exercices devront être réussis. |

# 3. DÉCOMPOSITION DES OBJECTIFS GÉNÉRAUX EN OBJECTIFS SPÉCIFIQUES

**Objectif général 1 :** Comprendre le protocole SNMP ainsi que son fonctionnement

| Objectifs spécifiques | Éléments de contenu | Méthodologies et moyens | Durée |
|---|---|---|---|
| Connaître les besoins et les raisons d'apparition du protocole SNMP. | ✓ Définition du protocole SNMP<br>✓ Fonctionnement SNMP<br>✓ La table MIB<br>✓ Autres protocoles de supervision | Exposé informel (tableau, data show, transparents) | 4h CI<br>9h TP |
| Maitriser un ou plusieurs outils de supervision sur les deux environnements Windows et Linux | ✓ MRTG<br>✓ TKINED<br>✓ Les communautés<br>✓ UCD-SNMP | Exposé informel (tableau, data show, transparents) | |

**Objectif général 2 :** Mise en place d'un service DHCP

| Objectifs spécifiques | Éléments de contenu | Méthodologies et moyens | Durée |
|---|---|---|---|
| Connaître les différents fichiers de configuration et les packages installés | ✓ Connaitre le protocole DHCP et son rôle<br>✓ Installation d'un service DHCP dans les 2 environnements Windows et Linux | Exposé informel (tableau, data show, transparents) | 3h CI<br>6h TP |
| Savoir modifier dans ces fichiers. | ✓ Savoir configurer correctement un service DHCP<br>✓ Exemples | Exposé informel (tableau, data show, transparents) | |

**Objectif général 3 :** Gestion et analyse du système DNS.

| Objectifs spécifiques | Éléments de contenu | Méthodologies et moyens | Durée |
|---|---|---|---|
| Installation d'un serveur DNS | ✓ Introduction DNS<br>✓ BIND<br>✓ Configuration de resolver | Exposé informel (tableau, data show, transparents) | 3h CI<br>6h TP |
| Maitriser les fichiers de configuration | ✓ Configuration des fichiers named<br>✓ Nslookup | Exposé informel (tableau, data show, transparents) | |

**Objectif général 4 :** Mise en place d'un serveur NIS.

| Objectifs spécifiques | Éléments de contenu | Méthodologies et moyens | Durée |
|---|---|---|---|
| Centraliser les comptes utilisateur | ✓ Installation d'un serveur NFS<br>✓ Mise en place de NIS | Exposé informel (tableau, data show, transparents) | 3h CI<br>6h TP |
| Configuration les fichers de configuration coté serveur et coté client | ✓ Yppasswd<br>✓ Ypcat<br>✓ Ypmatch ..etc. | Exposé informel (tableau, data show, transparents) | |

**Objectif général 5 :** Mise en place d'un serveur SAMBA

| Objectifs spécifiques | Éléments de contenu | Méthodologies et moyens | Durée |
|---|---|---|---|
| Partage des fichiers entre plusieurs environnements | ✓ Configuration générale<br>✓ Les sections globales, homes et printers<br>✓ Exemple de fonctionnement | Exposé informel (tableau, data show, transparents) | 1,5h CI<br>6h TP |

| | | | |
|---|---|---|---|
| Samba en tant que contrôleur de domaine | ✓ Création des utilisateurs et groupes sous Linux<br>✓ Corréler les groupes Linux avec les groupes Windows<br>✓ Ajouter les utilisateurs dans le domaine | Exposé informel (tableau, data show, transparents) | |

**Objectif général 6 :** Mise en place d'un annuaire LDAP

| Objectifs spécifiques | Éléments de contenu | Méthodologies et moyens | Durée |
|---|---|---|---|
| LDAP contrôleur de domaine PDC | ✓ Présentation et caractéristiques de LDAP<br>✓ Structure de LDAP | Exposé informel (tableau, data show, transparents) | 4h CI<br>9h TP |
| Maitriser la configuration des fichiers | ✓ Installation openLDAP<br>✓ Les attributs<br>✓ Création des utilisateurs et groupes dans l'annuaire | Exposé informel (tableau, data show, transparents) | |

**Objectif général 7 :** Mise en place d'un contrôleur de domaine Active Directory

| Objectifs spécifiques | Éléments de contenu | Méthodologies et moyens | Durée |
|---|---|---|---|
| Active Directory contrôleur de domaine PDC | ✓ Présentation de l'AD<br>✓ Caractéristiques de l'AD<br>✓ Structure de l'AD | Exposé informel (tableau, data show, transparents) | 4h CI |
| Maitriser l'environnement Windows | ✓ Installation de Windows 2003/2008 server<br>✓ Ajout d'un contrôleur de domaine AD<br>✓ Création des utilisateurs et groupes | Exposé informel (tableau, data show, transparents) | 6h TP |

Chapitre 1

# LE PROTOCOLE SNMP

**Objectif :**

*S'initier aux notions de base de l'administration des réseaux SNMP.*

**Éléments de contenu :**

- *Généralités*
- *Niveau d'action du protocole SNMP*
- *Fonctionnement du protocole SNMP*
- *Représentation des tables MIB*
  - *Lire les informations : avec GET*
  - *Modifier les informations : avec SET*
  - *Les alarmes TRAP*
- *Autre protocole de supervision : Cmip*

## 1 - Introduction :

En plus des protocoles qui fournissent des services de niveau réseau et des programmes d'applications qui utilisent ces services, les administrateurs ont besoin de logiciels qui, dans un réseau, permettent de traiter les problèmes de fonctionnement, de contrôler le routage et de signaler les machines qui ont des comportements anormaux. L'ensemble de ces activités correspond à l'administration de réseaux.

## 2 - Niveau d'action des protocoles d'administration de réseaux

Contrairement à un réseau LAN/WAN homogène, un grand réseau TCP/IP ouvert sur Internet n'utilise pas qu'un seul et unique protocole de communication de bas niveau puisqu'il est constitué de plusieurs sous-réseaux reliés par des routeurs IP. L'administration d'un grand réseau est donc sensiblement différente de celle d'un réseau homogène. Tout d'abord, parce qu'un administrateur peut gérer des routeurs hétérogènes. Ensuite, parce que les entités gérées peuvent ne pas utiliser un même protocole de bas niveau. Enfin, l'ensemble des machines gérées par un administrateur peuvent être situées n'importe où dans l'internet. Un administrateur peut alors se trouver dans l'impossibilité de communiquer avec des ordinateurs dont il a la responsabilité, à moins que le logiciel d'administration qu'il utilise offre une connectivité de bout en bout à travers Internet.

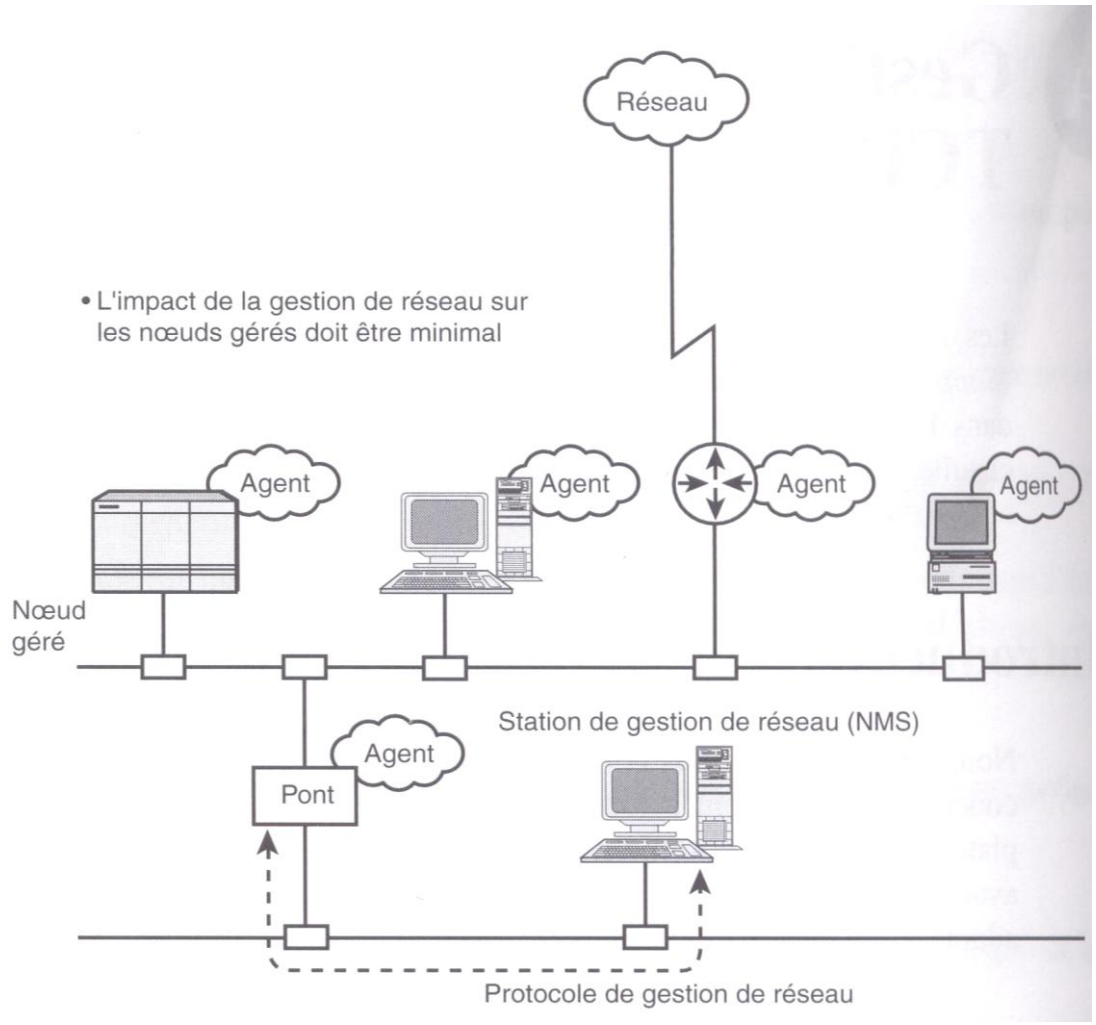
En conséquence, le protocole d'administration d'un réseau TCP/IP doit fonctionner au dessus du niveau transport :

*Dans un réseau TCP/IP, les routeurs IP sont des commutateurs actifs que les administrateurs doivent surveiller et contrôler. Comme les routeurs relient des réseaux hétérogènes, les protocoles d'administration opèrent au niveau application et communiquent en utilisant des protocoles TCP/IP de niveau transport.*

Concevoir le logiciel d'administration opérant au niveau application présente plusieurs avantages. Il peut ainsi être conçu indépendamment de l'architecture de la machine à gérer, sur n'importe quel ordinateur ou routeur. Du point de vue de l'administrateur, le fait d'utiliser un même ensemble de protocole garantit l'homogénéité : tous les ordinateurs ou routeurs répondent au même ensemble de commandes. De plus, comme le logiciel d'administration utilise IP pour ses communications, un administrateur peut contrôler des routeurs en tout point d'un réseau TCP/IP sans avoir besoin d'un accès direct.

Réaliser un logiciel d'administration de réseaux au niveau application présente toutefois quelques inconvénients. Un mauvais fonctionnement du système d'exploitation ou des protocoles TCP ou IP, peut interdire à l'administrateur d'entrer en contact avec un routeur ou un serveur.

## 3 - Modèle Architectural




Dans ce modèle, le réseau est constitué de plusieurs périphériques, chacun d'entre eux exécutant un agent de gestion. Cet agent connaît les paramètres du périphérique sur lequel il s'exécute. Certains de ces paramètres sont spécifiques au périphérique : un périphérique de routage disposera par exemple de sa table de routage. Tous les périphériques ont des paramètres en commun, par exemple un nom, une durée de fonctionnement sans incident, etc …

Les agents peuvent être gérés par un périphérique particulier appelé station de gestion de réseau (ou NMS, Network Management Station). La station NMS peut envoyer des requêtes à un périphérique afin d'obtenir des informations sur son paramétrage. L'agent du périphérique reçoit la requête et renvoie les informations demandées. Lorsqu'elle reçoit cette réponse, la station NMS peut utiliser les informations de configuration du périphérique afin de déterminer les opérations à entreprendre en fonction de son état.

Il est aussi important d'empêcher une station NMS non autorisée d'avoir accès aux informations de paramétrage des périphériques composant le réseau. Il faut donc disposer d'un mécanisme d'authentification afin d'empêcher les accès non autorisés.

Les mécanismes permettant de suivre et de contrôler le réseau doivent avoir le plus petit impact possible sur le réseau. En d'autres termes, les protocoles utilisés pour la collecte d'informations ne doivent pas amoindrir les performances du réseau et des périphériques qui le composent. Si le mécanisme de gestion de réseau utilise trop de bande passante, il pénalisera les utilisateurs. De la même façon, les agents s'exécutant sur les différents périphériques ne doivent pas consommer trop de ressources système, sans quoi les périphériques ne pourront pas effectuer correctement leurs tâches.

# 4 - Architecture des protocoles d'administration

## 4.1 – Protocole standard d'administration de réseaux : SNMP

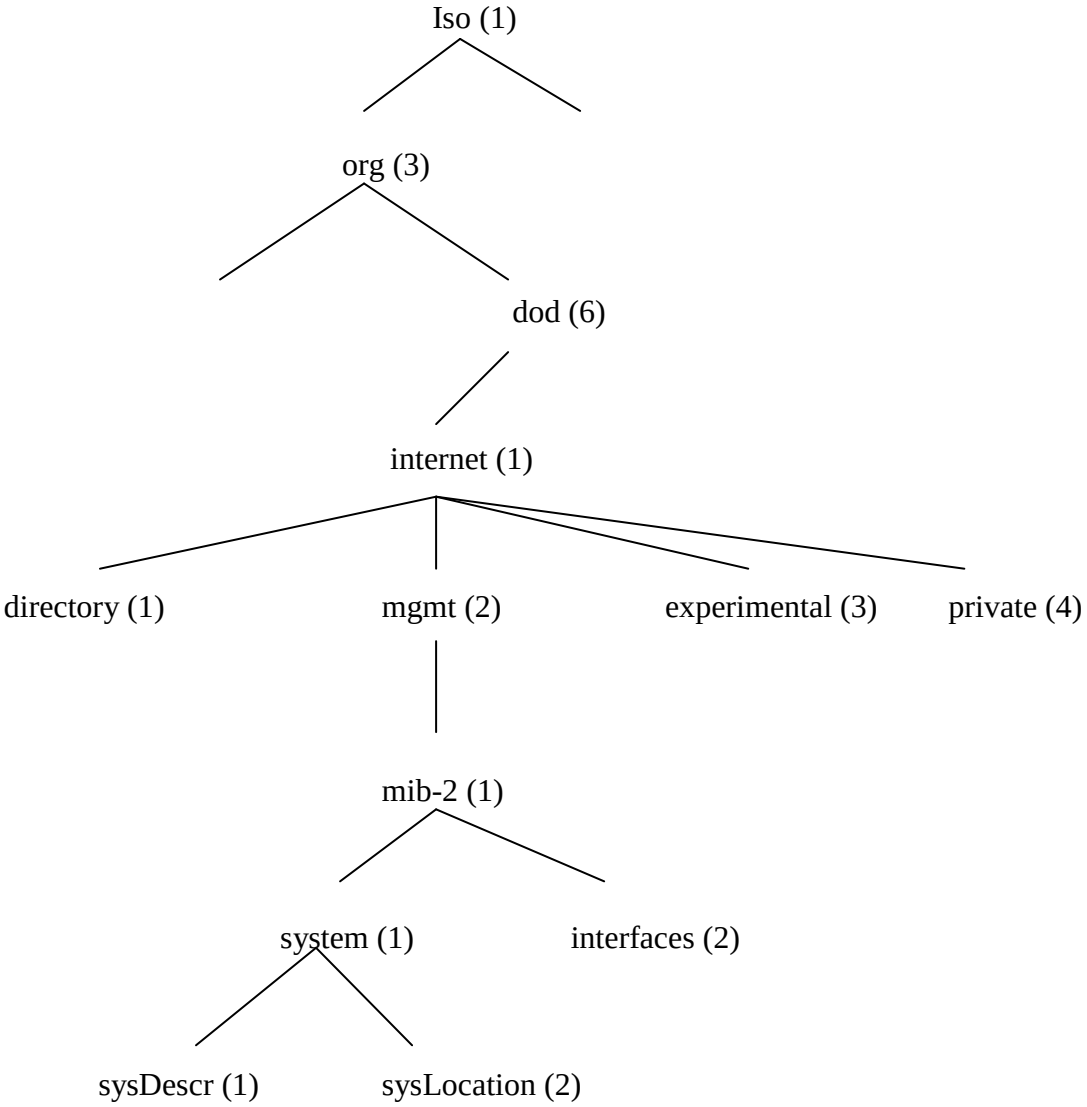
Le standard actuel d'administration de réseaux TCP/IP est le protocole SNMP (Simple Network Management Protocol). Actuellement c'est la version 3 de ce protocole qui est en cours de diffusion. Cette version comporte des fonctionnalités nouvelles, en particulier sur le plan de la sécurité.

## 4.2 – Définition standard des informations d'administration

Tout routeur administré doit tenir à jour des informations d'état accessibles à un administrateur. Par exemple, il doit fournir des statistiques relatives à l'état de ses interfaces réseau, au trafic entrant ou sortant, aux datagrammes détruits et aux messages d'erreurs émis. SNMP permet à un administrateur d'accéder aux statistiques des machines, mais il ne spécifie pas de détails sur les données associées. C'est le rôle de la base de données MIB (Management Information Base) de définir les données en détail. Par exemple, la MIB spécifie d'une part qu'un logiciel IP doit disposer d'un compteur d'octets comptabilisant tous les octets qui arrivent sur une interface réseau et, d'autre part que le logiciel d'administration peut uniquement lire ces compteurs. La MIB TCP/IP décompose les informations d'administration de réseaux en plusieurs catégories standard.

### 4.3 – Structure et représentation des noms d'objets MIB

Sous SNMP, les objets MIB reçoivent un identifiant unique composé de séquences de chiffres séparés par des points. Cette séquence se lit de gauche à droite et correspond à des nœuds dans l'arborescence des noms



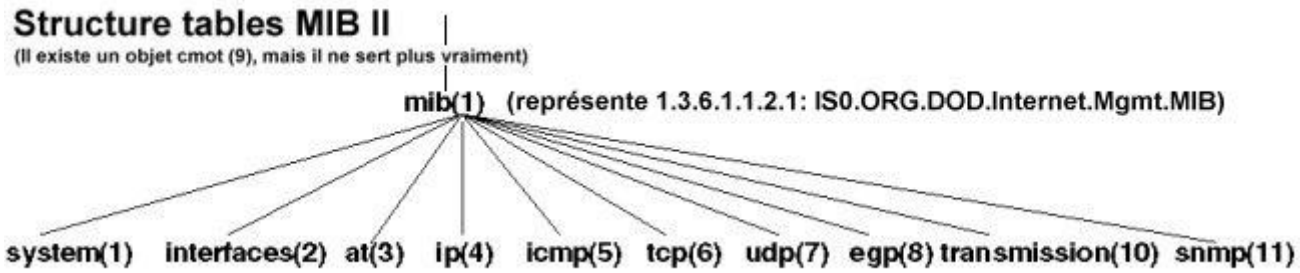

La gestion de réseau est définie par la branche iso(1). Dans cette branche on trouve un certain nombre de définitions d'organisations subordonnées. La gestion de réseau entre dans le nœud org(3).
Sous le nœud dod(6) se trouvent un certain nombre de réseaux subordonnés. La gestion de réseau entre dans le nœud internet(1).
Sous le nœud internet(1) se trouvent un certain nombre de nœuds subordonnés représentant différents services et tentatives de normalisation. La gestion de réseau standardisée se trouve dans le réseau mgnt(2).
Sous le nœud mgnt(2) se trouvent un certain nombre de nœuds subordonnés représentant différents services et tentatives de normalisation. La gestion de réseau standardisée se trouve dans le nœud MIB-2(1).

Sous le nœud mib-2(1) se trouvent un certain nombre de nœud subordonnés représentant différents groupement de variables MIB.
Sous le nœud system(1), on trouve 2 variables MIB sysDescr(1) et sysLocation(2). Les identifiants d'objets de ces variables s'obtiennent en ecrivant de gauche à droite les différents nœuds, séparés par des points :

- sysDescr : 1.3.6.1.2.1.1.1
- sysLocation : 1.3.6.1.2.1.1.2

### 4.4 - Les tables MIB

Ce sont des tables contenant les informations de l'élément du réseau. Ces informations sont hiérarchisées sous forme d'arbre :




**at (adress translation)** : Table d'adresses IP pour les correspondances d'adresses MAC

**ip** : Statistiques du protocole IP, adresse cache et table de routage

**icmp** : Statistiques du protocoles ICMP

**tcp** : Paramètres TCP, statistiques et table de connexion

**udp** : Statistiques UDP

**egp** : Statistiques EGP, table d'accessibilité

**snmp** : Statistiques du protocole SNMP

En plus du standard MIB de TCP/IP, qui s'appelle maintenant MIB-II, un nombre important de RFC détaillent des variables MIB pour divers type de périphériques. Examinons quelques éléments de données de la MIB pour en clarifier le contenu.

| Variables MIB | Catégorie | Signification |
|---|---|---|
| sysUpTime | système | Durée écoulé depuis dernier démarrage |
| ifNumber | interfaces | Nombre d'interfaces réseau |
| ifMtu | interfaces | MTU d'une interface particulière |
| ipDefaultTTL | ip | Valeur utilisée dans le champ TTL |
| ipInReceives | ip | Nbre de datagrammes reçus |
| ipForwDatagrams | ip | Nbre de datagrammes acheminés |
| ipOutNoRoutes | ip | Nbre d'erreurs de routage |
| ipReasmOKs | ip | Nbre de datagrammes réassemblés |
| ipFragOKs | ip | Nbre de datagrammes fragmentés |
| ipRoutingTable | ip | Table de routage IP |
| icmpInEchos | icmp | Nbre de demandes d'echo ICMP reçues |
| tcpMaxConn | tcp | Nbre maxi de connexions TCP autorisées |
| tcpInSegs | tcp | Nbre de segments reçus par TCP |

| | | |
|---|---|---|
| udpInDatagrams | udp | Nbre de datagrammes UDP reçus |

Les valeurs des éléments de chacune des variables ci-dessus peuvent être enregistrées au moyen d'un seul entier. Toutefois, la MIB permet également de définir des valeurs plus complexes, comme par exemple la variable ipRoutingTable qui fait référence à la table de routage d'un routeur. Des variables MIB supplémentaires sont définies pour le contenu de la table et pour permettre aux protocoles d'administration de réseaux de référencer les données correspondant à chaque entrée.

# 5 - Structure des informations d'administration de réseaux

En complément du standard MIB qui définit les informations spécifiques d'administration réseaux et leur signification, un standard séparé spécifie l'ensemble des règles utilisées pour définir et identifier les variables MIB. Ce sont les règles de gestion des informations d'administration, SMI (Structure of Management Information). Pour que le protocole d'administration de réseaux reste simple, SMI pose des restrictions sur les types de variables autorisées dans la MIB, spécifie les règles de nommage de ces variables et crée les règles de définition des types de variables. Par exemple SMI comprend des définitions de termes comme *IpAddress* (défini comme une chaîne de 4 octets) et *Counter* (entier appartenant à l'intervalle $[0,2^e32-1]$ et indique que ce sont les termes utilisés pour définir les variables MIB. De plus, SMI décrit la façon dont la MIB référence les tables de valeurs (les tables de routage IP, par exemple).

## 5.1 - Définitions formelles utilisant ASN.1

Le standard SMI indique que toutes les variables MIB doivent être définies et référencées à l'aide de la notation ISO de syntaxe abstraite ASN.1 (Abstract Syntax Notation 1). ASN.1 est un langage formel qui présente 2 caractéristiques principales : une notation utilisée dans les documents manipulés par les humains et une représentation codée et concise de la même information, utilisée dans les protocoles de communication. Dans les 2 cas, la notation formelle élimine toutes les ambiguïtés possibles, tant du point de vue de la représentation que de la signification. Au lieu de dire par exemple, qu'une variable contient une valeur entière, un concepteur qui utilise ASN.1 doit définir la forme exacte et le domaine des valeurs prises par cet entier.

### 5.2 - Les messages SNMP

Le format et la longueur des messages SNMP sont variables et relativement complexes. On utilise ASN.1 pour décrire la structure des messages SNMP.
Voici un exemple d'utilisation d'ASN.1 décrivant la structure d'une trame Ethernet :

```
Ethernet-Frame ::= SEQUENCE {

        destAddr    OCTET STRING (SIZE(6)),
        srcAddr     OCTET STRING (SIZE(6)),
        etherType   INTEGER (1501..65535),
        data        ANY (SIZE(46-1500)),
        crc         OCTET STRING (SIZE(4))

        }
```

Remarquez que le ::= sert à attribuer à la variable de gauche la définition du membre de droite. SEQUENCE représente une liste ordonnée d'éléments. Cette liste ordonnée contient les champs d'une trame Ethernet, notamment l'adresse de destination, l'adresse source, le type d'Ethernet, les données et le CRC.

Le type d'un champ est spécifié à la suite du nom. Par exemple, destAddr et srcAddr sont déclarés comme étant de type OCTET STRING, définissant une variable de 0 ou plusieurs octets.
La valeur de chaque octet est comprise entre 0 et 255. La taille de la chaîne obtenue est placée après la déclaration de la chaîne OCTET STRING.

Ethertype est défini en tant que INTEGER, à savoir une valeur entière de taille et de précision arbitraire. Le (1501..65535) situé juste après permet de définir sa plage de valeur.

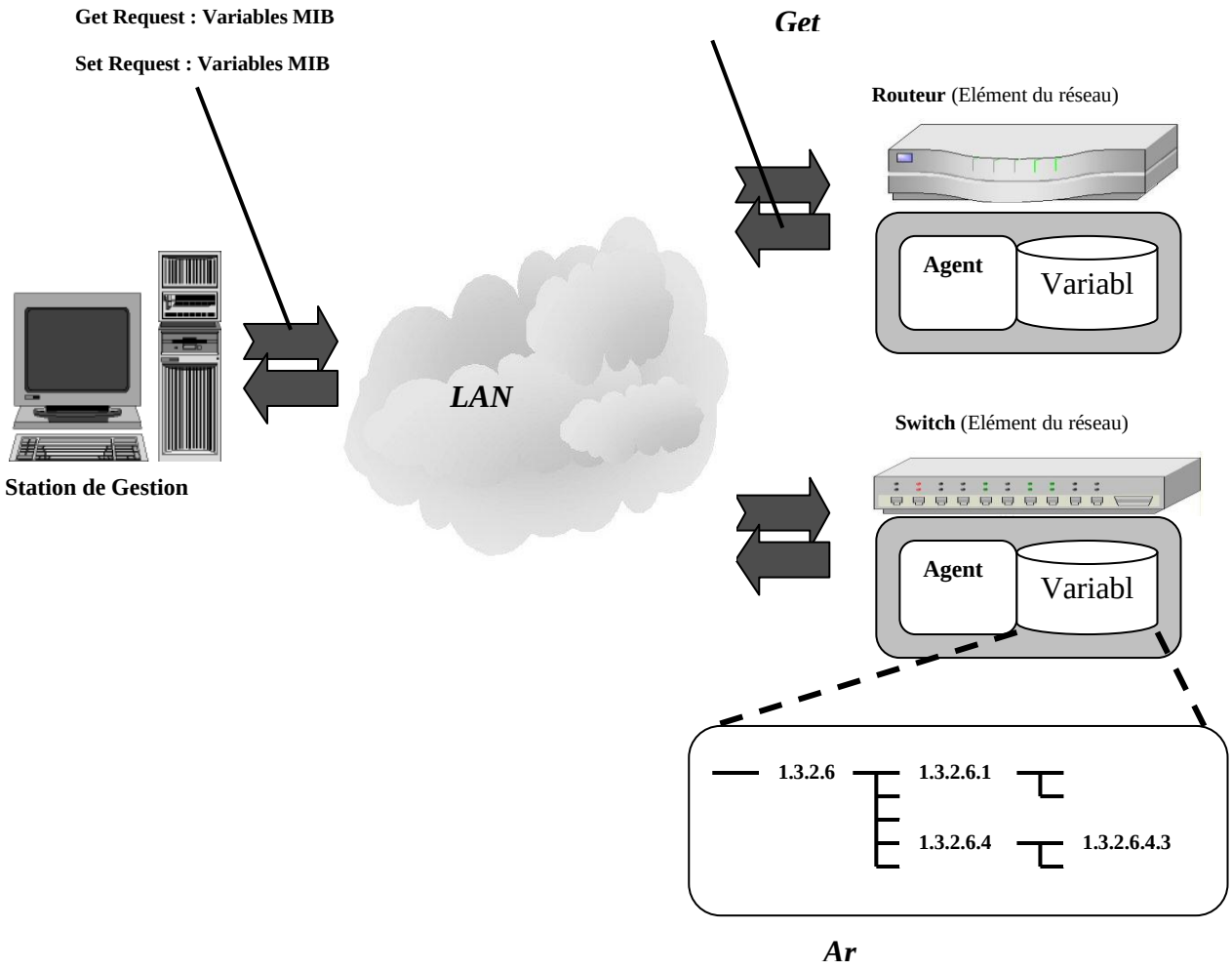
Le champ de donnée est de type ANY et sa taille varie de 46 à 1500 octets.

En utilisant la notation ASN.1, les messages SNMP prennent le format suivant :

```
SNMP-message ::= SEQUENCE {

        version     INTEGER {version-1(1)},
        community   OCTET STRING,
        data        ANY }
```

## 6 - SNMP : fonctionnement

SNMP est un protocole utilisé dans des programmes de gestion de réseaux informatiques. Son utilisation est simple; un ordinateur central interroge tous les nœuds de son réseau (terminaux, PC, routeurs, ponts, etc.) sur leur état. C'est l'agent qui s'exécute sur chacun de ces nœuds qui traite les demandes en modifiant ou prenant de l'information dans sa table MIB. La « Station de Gestion » (voir **Figure 1**) peut donc diagnostiquer des problèmes et tenir des statistiques sur l'état du réseau.




**- Figure 1 : L'environnement SNMP -**

## 6.1 - SNMP, les spécifications

Le fonctionnement de SNMP est asymétrique; il est constitué d'un ensemble de requêtes, de réponses et d'un nombre limité d'alertes. La « Station de Gestion » (**Figure1**) envoie des requêtes à l'agent, lequel retourne des réponses. 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**. Le schéma ci-dessus résume bien le fonctionnement du protocole SNMP.

Le gestionnaire SNMP doit être à même de lire et de modifier les valeurs des variables MIB des périphériques qu'il gère.

Lorsqu'un événement inattendu se produit sur un des périphériques gérés, par exemple un échec de transmission et une modification d'état, le périphérique envoie un message de *trap SNMP* au gestionnaire SNMP. Ce message contient une indication de l'événement ayant provoqué la génération du message. Le gestionnaire SNMP doit alors prendre les mesures qui s'imposent. Il peut se contenter d'enregistrer le message dans un fichier de suivi, ou prendre des mesures plus immédiates, par exemple demander des informations complémentaires au périphérique lui ayant envoyé le message. Ces informations supplémentaires sont obtenues grâce à des requêtes de lecture des variables MIB. Si le gestionnaire SNMP est programmé pour contrôler le périphérique, il peut lui demander de modifier la valeur de ses variables MIB.

Lorsque le gestionnaire décide de modifier l'état du périphérique, il le fait en modifiant les variables MIB du périphérique. Par exemple, le gestionnaire modifiera la variable MIB *ipPowerOff* d'un périphérique afin de l'éteindre à distance.

Comme les variables MIB sont ordonnées selon leur identificateur d'objet, le gestionnaire SNMP peut parcourir toutes les variables du périphérique en utilisant la commande SNMP *GetNext*.

## 6.2 - Les différentes requêtes d'un message SNMP

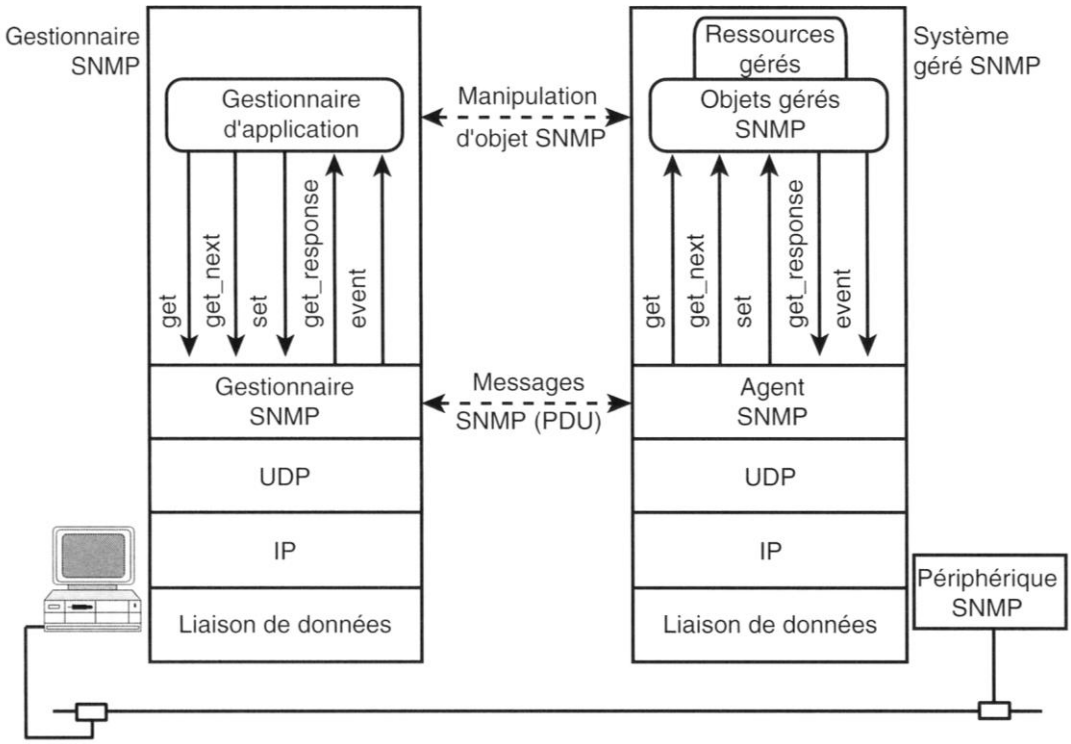
Un système SNMP supporte trois types de requêtes : GET, SET et TRAP. SNMP utilise alors les opérations suivantes pour la gestion du réseau :

### 6.2.1 - Lire les informations GET

- GetRequest : Cette requête permet aux stations de gestion (manager) d'interroger les objets gérés et les variables de la MIB des agents. La valeur de l'entrée de la MIB (nom) est passée en paramètre. Elle permet d'accéder à une variable précise.
- GetNextRequest : Cette requête permet aux stations de gestion de recevoir le contenu de l'instance qui suit l'objet nommé (passé en paramètre) dans la MIB. Cette commande permet en particulier aux stations de gestion de balayer les tables des MIB. Elle permet d'accéder à plusieurs variables simultanément.
- GetResponse: À des requêtes, l'agent répond toujours par *GetResponse*. Toutefois si la variable demandée n'est pas disponible, le *GetResponse* sera accompagné d'une erreur *noSuchObject*.

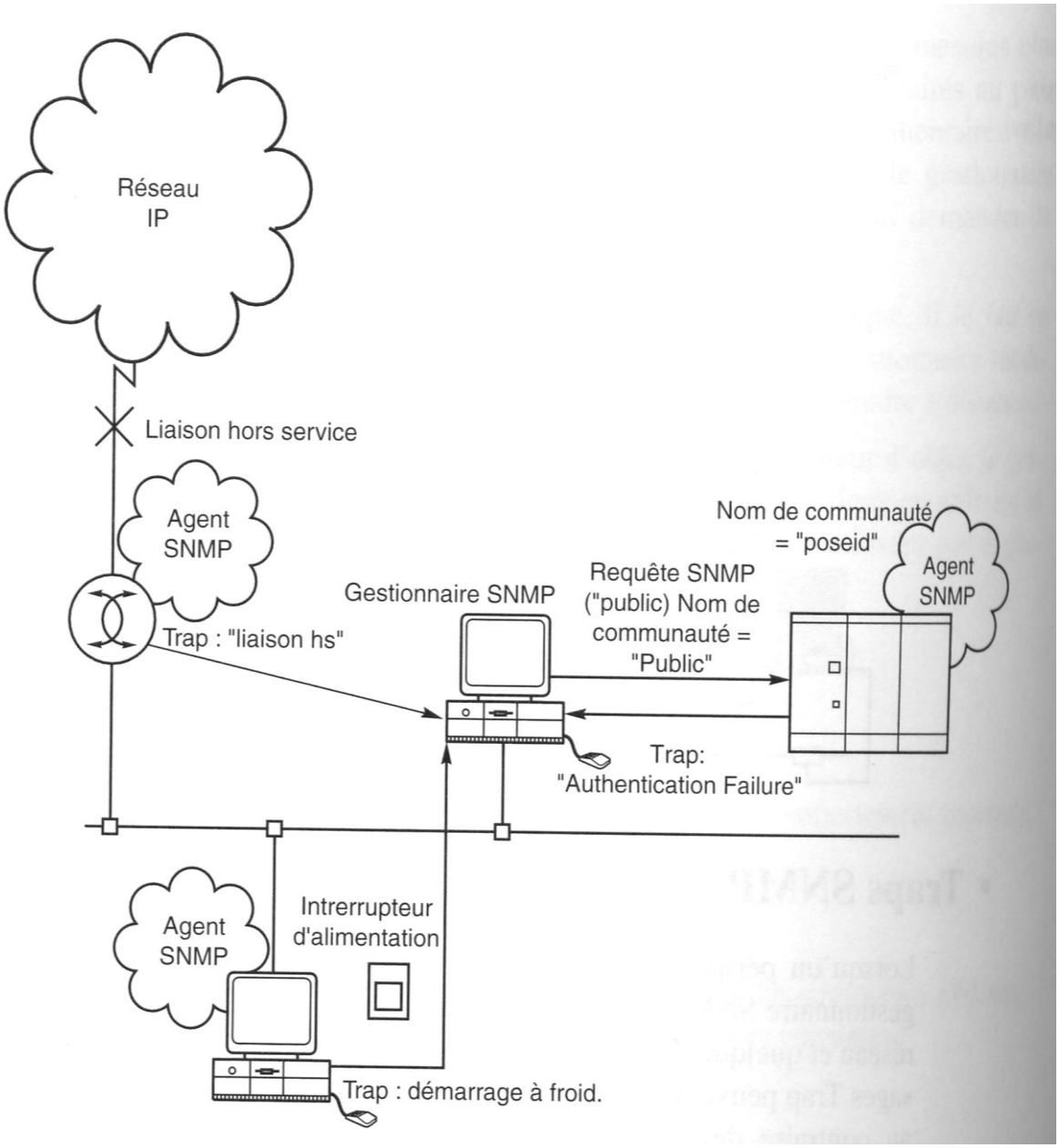
### 6.2.2 - Modifier les informations SET

- SetRequest : Cette requête permet aux stations de gestion de modifier une valeur de la MIB ou d'une variable et de lancer des périphériques. Elle permet par exemple à un manager de mettre à jour une table de routage. SetRequest provoque aussi le retour de GetResponse.



### 6.2.3 - Les alarmes TRAP

- Lorsqu'un périphérique entre dans un état anormal, l'agent SNMP prévient le gestionnaire SNMP par le biais d'un Trap SNMP. Les messages Trap peuvent être Link Up ou Link Down (lorsque l'interface devient active ou au contraire passive), cold start (démarrage à froid), warm start (démarrage à chaud), réinitialisation de l'agent SNMP, authentication failure (échec d'authentification, lorsqu'un nom de communauté incorrect est spécifié dans une requête), loss of EGP neighbour (perte de voisin EGP)



### 6.3 - Les différentes versions snmp

*Il existe 3 versions du protocole SNMP : SNMPv1, SNMPv2 et SNMPv3. Comme précédemment mentionné, la version 1 est encore la plus utilisée.*

- **SNMPv1 (complet)** : Ceci est la première version du protocole, telle que définie dans le RFC 1157. La sécurité de cette version est triviale, car la seul vérification qui est faite est basée sur la chaîne de caractères « community ».

- **SNMPsec (historique)** : Cette version ajoute de la sécurité au protocole SNMPv1 et est définie par RFC 1351, RFC 1352 et RFC 1353. La sécurité est basée sur des groupes. Cette version n'est que rarement utilisée.

- **SNMPv2p (historique)** : Cette version apporte des mises à jour des opérations du protocole, de nouvelles opérations et de nouveaux types de données. La sécurité est basée sur les groupes de SNMPsec. Cette version est décrite par les RFC 1441, RFC 1445, RFC 1446, RFC 1448 et RFC 1449.

- **SNMPv2c (expérimental)** : Cette version est une amélioration des opérations de protocole et des types d'opérations de SNMPv2p et utilise la sécurité par chaîne de caractères « community » de SNMPv1. Cette version est définie par RFC 1901, RFC 1905 et RFC 1906.

- **SNMPv2u (expérimental)** : Cette version du protocole utilise les opérations, les types de données de SNMPv2c et la sécurité basée sur les usagers. Cette version est décrite par RFC 1905, RFC 1906, RFC 1909 et RFC 1910.

- **SNMPv2* (expérimental) :** Cette version combine les meilleures parties de SNMPv2p et SNMPv2u. Les documents qui décrivent cette version n'ont jamais été publiés dans les RFC.

- **SNMPv3 (proposé)** : Cette version comprend une combinaison de la sécurité basée sur les usagers et les types et les opérations de SNMPv2p, avec en plus la capacité pour les « proxies ». La sécurité est basée sur ce qui se trouve dans SNMPv2u et SNMPv2*.

## 7 - SNMP, la sécurité

Les deux premières versions du protocole SNMP offre de nombreuses failles en matière de sécurité. Un pirate pourrait exécuter des commandes systèmes sur un serveur, un switch ou un routeur, si un accès etait laissé à l'extérieur.

Ceci serait dû à un mauvais fonctionnement de la pile d'exécution des managers SNMP. Exploitée, cette faille pourrait selon le Cert mettre en péril des milliers de machines à travers le monde et menacer l'équilibre d'Internet.

## 8 - Cmip

Le CMIP (*Common Management Information Protocol*) est un protocole de gestion développé par ISO pouvant fonctionner sur des réseaux hétérogènes. Nous pouvons le comparer au SNMP sur le fait que les deux protocoles se servent de tables MIB pour effectuer leur travail. D'ailleurs, le CMIP a été construit à partir du SNMP.

Par contre, leur fonctionnement est plutôt différent puisque dans le protocole CMIP, la station s'occupant de la gestion ne va pas chercher elle-même les informations; elle attend que les stations rapportent leur état. Le CMIP est un protocole très évolué comparativement au SNMP; les stations doivent pouvoir exécuter les tests elles-mêmes. Par exemple, une station, qui a eu plusieurs problèmes consécutifs à accéder à un serveur de fichiers, doit en avertir l'ordinateur de gestion.

Cela engendre de grande différences de performance par rapport au SNMP. Le trafic sur le réseau est diminué, car il y a des transferts d'information que lorsqu'il y a quelque chose d'important à rapporter. Par contre, les ressources des appareils sont énormément plus utilisées avec le CMIP, dans un rapport approximatif de 10 fois.

Le CMIP est un protocole plus sécurisé que le protocole SNMP, mais sa complexité exige un réseau dont les appareils doivent êtres assez performants.

## 9 - Choix des produits : Tkined & Mrtg

MRTG est un produit qui permet de voir très facilement les trafics entrants et sortants des routeurs et des switchs. Il permet aussi d'avoir une vue hebdomadaire, mensuel et même annuel du trafic des éléments ciblés. Ce produit était déjà utilisée sans le protocole SNMP pour le routeur. Son utilisation était déjà très apprécié des employés du centre serveur.

Tkined permet d'accéder très facilement à n'importe quelles variables SNMP de la table MIB. Il ne permet néanmoins pas de faire des statistiques. Il peut offrir une bonne vue de l'architecture du réseau.

Ces deux produits sont complémentaires : les deux permettent d'avoir une bonne exploitation des remontées SNMP des switchs.

Chapitre 2

# SUPERVISION MRTG

**Objectif :**

*Maitriser un outil de supervision qui se base sur le protocole SNMP, dans ce chapitre on traite le cas de MRTG.*

**Éléments de contenu :**

- *Présentation de MRTG*
- *Statistiques sur la journalisation*
- *Génération des pages html par MRTG*
- *Configuration*
  - *Création des répértoires*
  - *Création des pages HTML*
  - *Evolution des graphiques : CRON*

# 1 - Présentation

Le logiciel "Multi Router Traffic Grapher" (MRTG) est un outil permettant de superviser la charge du trafic sur des éléments réseaux. MRTG génère des pages HTML contenant des images graphiques permettant de visualiser le trafic.

Le fonctionnement de MRTG est basé sur un ensemble de scripts Shell ou Perl. Ce sont ces scripts qui récupère via un des deux modes de communication (voir ci-dessous, communication *via SNMP* et *sans SNMP*), les informations contenues dans les pages html générées.

MRTG peut interroger suivant deux modes de communication :

- Via le protocole SNMP (et les communautés) : pour l'utilisation des fonctions de base de mrtg ; c'est à dire la visualisation du trafic des switchs ou des routeurs. MRTG possède déjà dans son répertoire d'installation les outils SNMP permettant de le faire. Cependant, si l'on veut utiliser MRTG pour récupérer la charge CPU, l'espace disque, … d'un serveur, il faut installer d'autres outils SNMP contenus dans le package **ucd-snmp**.
- Sans le protocole SNMP ; en faisant un accès distant et en téléchargeant les informations.

## 1.1 - Des résultats quasiment immédiats

Les graphiques sont actualisés autant de fois que mrtg est lancé. Ainsi, sous Linux en utilisant la crontab les informations peuvent être recueillies toutes les 5 minutes (voir la configuration). Ce qui permet d'avoir une visualisation du trafic quasiment immédiate.

## 1.2 - Des Statistiques

Les graphiques sont de quatre types :

- **journalier** : graphique visualisant le trafic sur 1 journée.
- **hebdomadaire**: graphique visualisant le trafic sur 1 semaine.
- **mensuel** : graphique visualisant le trafic sur 1 mois.
- **annuel** : graphique visualisant le trafic sur 1 année.

Pour chaque interface de la machine ciblé, un graphique de chaque page est généré.

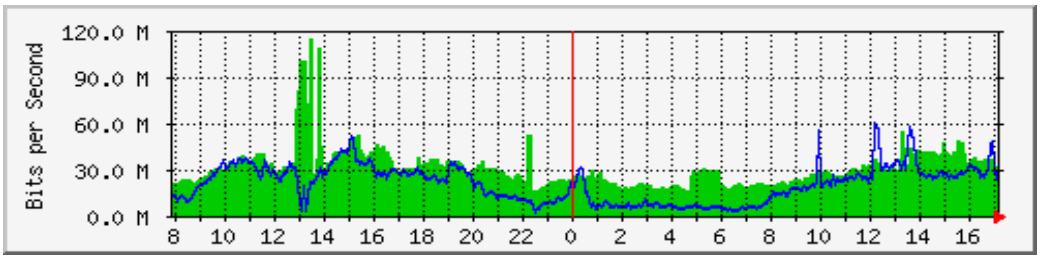
Ces types de graphiques fournissent un certain nombre de statistiques permettant d'apprécier plus finement l'efficacité de chaque liaison réseau.

## 1.3 - Exemples de pages html générées par mrtg

**MRTG génère dans une page Index des graphiques sous forme d'images. Chaque graphique représente une interface. Ces graphiques présentent le trafic journalier.**

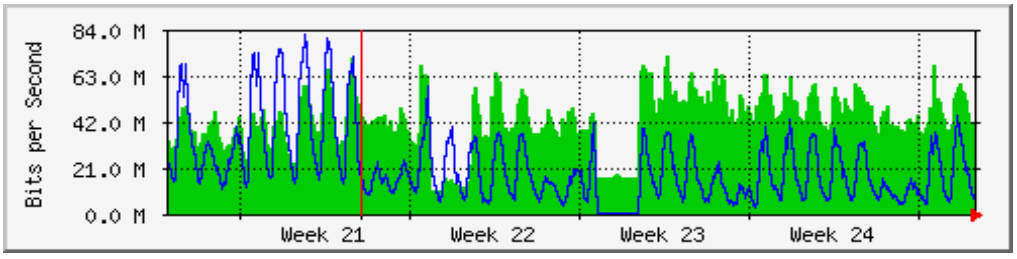
Le **Graphique 1** est un exemple de ce type de graphique.

La courbe en **bleue** représente le flux sortant et la courbe en **vert** représente le flux entrant.

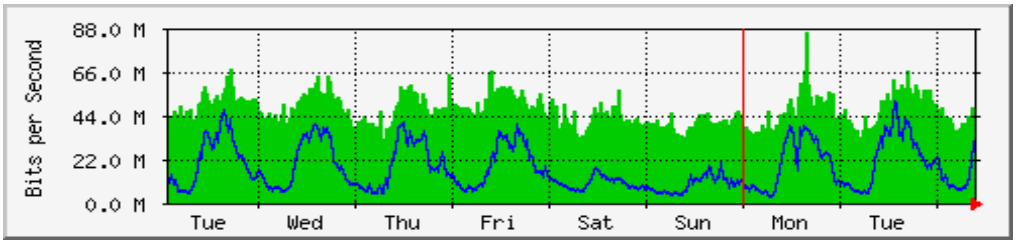



**- Graphique 1 -**

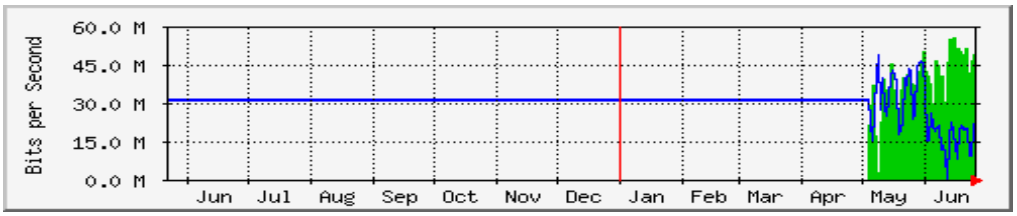
Si l'on clique sur un des graphiques de la page Index sur **Graphique 1**, par exemple, on obtient les
4 types de graphiques présentés précédemment : **journalier** (c'est le même que **Graphique 1**), **hebdomadaire**, **mensuel**, **annuel**. Les graphiques **Graphique 2, Graphique 3** et **Graphique 4** sont respectivement des graphiques de type hebdomadaire, mensuel et annuel.




**- Graphique 2 -**




**- Graphique 3 -**




**- Graphique 4 -**

## 2 - Configuration

Pour des raisons de commodité et de sécurité, nous allons crée un utilisateur mrtg.

```
# adduser mrtg
# mkdir mrtg
# chown mrtg:users /home/mrtg
```

Pour plus de sécurité encore, nous empêcherons à cet utilisateur de pouvoir se loguer : dans /etc/password, on mettra à la ligne qui correspondant à l'utilisateur mrtg un **!** à la place d'un **x** et **/bin/false** à la place de **/bin/bash** :

```
…
…
mrtg:!:xxxxxxx:/bin/false
```

De là, toutes les commandes qui suivent ainsi que l'utilisation des outils mrtg seront faits en tant qu'utilisateur mrtg

```
#   chown   mrtg:users   /usr/local/mrtg-
2.9.18 -R
# su mrtg
$
```

## 2.1 - Création des répertoires

- **Création du répertoire contenant les fichiers de conf**

  Un fichier de configuration est utilisé pour chaque machine supervisé. Dans notre configuration, le répertoire ***/home/mrtg/*** contiendra tous les fichiers de configuration.

```
$ mkdir /home/mrtg
```

- **Création du répertoire contenant les pages html**

  Sur la machine où j'ai installé mrtg, les pages web sont stockées dans le répertoire **/usr/local/httpd/htdocs**, je crée le répertoire mrtg qui contiendra les pages html.

```
$ mkdir /usr/local/httpd/htdocs/mrtg
```

## 2.2 - Création du fichier de configuration associé au switch xxx.xxx.xxx.253

J'ai pris comme convention de prendre comme référence le dernier octet de l'adresse ip des switchs sur lesquels je travaillais (içi **253**)

- **Création du répertoire contenant les pages web du switch 253**

```
$ mkdir /usr/local/httpd/htdocs/mrtg/253
```

- **cfgmaker : création du fichier de conf**

Dans le répertoire **bin** du répertoire d'installation de mrtg, se trouve l'utilitaire **cfgmaker** qui permet de un fichier de configuration standard.

Ou /usr/bin/cfgmaker

```
$ /usr/local/mrtg-2.9.18/bin/cfgmaker --global 'WorkDir:   \
/usr/local/httpd/htdocs/mrtg/253'   \
--global 'Options[_]: bits,growright'   \
--output /home/mrtg/253.cfg   \ nom_communaute@xxx.xxx.xxx.253
```

Par défaut, le nom de la communauté nom_communaute est **public**.

**Remarque** : le fichier de configuration du switch contient, en commentaire, la présentation de chaque interface du switch au format .html. Ce sont ces balises qui serviront de structure à la création des fichiers .html.

## 2.3 - Création des pages .html associés aux interfaces

Dans le répertoire **bin** du répertoire d'installation de mrtg, se trouve l'utilitaire **mrtg** qui permet de créer les pages .html contenant les graphiques (sous forme d'images .png) de chaque interface du switch *xxx.xxx.xxx*.253.

```
$  /usr/local/mrtg-2.9.18/bin/mrtg /home/mrtg/253.cfg
```

## 2.4 - Création de la pages .html regroupant toutes les interfaces

Dans le répertoire **bin** du répertoire d'installation de mrtg, l'utilitaire **indexmaker** génère en sortie une page .html qui regroupe toutes les interfaces générées par mrtg.

```
$ /usr/local/mrtg-2.9.18/bin/indexmaker /home/mrtg/253.cfg > \
/usr/local/httpd/htdocs/mrtg/253/index.html
```

## 2.5 - Evolution des graphiques générés : cron

Pour que les graphiques évoluent (c.a.d. que les images soient recrées en tenant compte des infos récupérées précédemment), il faut relancer mrtg plusieurs fois ...
On peut par exemple, faire une requête sur le switch toutes les 5 minutes. Pour se faire, il faut faire une entrée dans la crontab qui exécutera la commande mrtg toutes les 5 minutes.

```
$ crontab -e
```

Sous l'éditeur, on tape :

```
*/5 * * * * /usr/local/mrtg-2.9.18/bin/mrtg /home/mrtg/253.cfg 2> \
/dev/null
```

La redirection vers /dev/null permet d'éviter les messages d'erreurs sur la sortie standard.

**Remarque** : La configuration de mrtg pour d'autres switchs ou routeurs est équivalente.

Chapitre 3

# SUPERVISION TKINED

**Objectif :**

*Maitriser un outil de supervision qui se base sur le protocole SNMP, dans ce chapitre on traite le cas de TKINED.*

**Éléments de contenu :**

- *Présentation de TKINED*
- *Principes de fonctionnement*
- *Administration avec TKINED*
- *Utilisation de SNMP dans TKINED*
- *Exemples*
  - *Visualisation d'une MAP avec TKINED*
  - *Visualisation des variables MIB*

# 1 - Presentation

Tkined a été conçu dans le but de faciliter le contrôle et la gestion des réseaux basés sur le protocole IP. Tkined permet de récupérer les diverses informations produites par les équipements du réseau de façon à connaître à un instant donné l'état de l'ensemble des équipements du réseau.

## 1.1 - Caractéristiques techniques

Tkined est un outil qui fait parti du package logiciel "Scotty". Ce package comprend notamment l'outil "Tnm", une extension du langage Tcl permettant d'accéder aux informations générales d'administration de réseau (qui supporte notamment les protocoles SNMP, ICMP, DNS, HTTP,...). Tkined est basé sur le langage Tcl et son extension "Tnm", ce qui lui procure une flexibilité importante, l'utilisateur pouvant facilement implémenter, selon ses besoins, des fonctionnalités nouvelles en Tcl, en s'appuyant sur l'extension "Tnm".

## 1.2 - Principes de fonctionnement

L'administration d'un réseau sous Tkined s'effectue par l'intermédiaire d'une interface graphique permettant de modéliser facilement et rapidement le réseau dont on veut effectuer l'administration. La création des équipements à prendre en considération dans l'administration s'effectue simplement en cliquant sur l'icone prévue à cet effet dans la barre située sur la gauche de l'écran , puis en cliquant à l'emplacement souhaité pour l'équipement. Dès que l'icone de l'équipement est en place, l'utilisateur doit spécifier les paramètres de l'équipement (sélection de l'équipement, puis clic droit). Une fois les équipements paramétrés, la création du réseau reliant ces équipements s'effectue simplement en utilisant le trait épais pour symboliser le réseau et le trait fin pour relier les équipements au réseau (les traits sont situés dans la partie gauche de l'écran). Ceci représente les fonctionnalités principales et indispensables pour débuter. Tkined offre aussi d'autres possibilités tel que regrouper plusieurs éléments en un seul (possibilité de modéliser des réseaux importants), modifier les icones des équipements, ...

## 1.3 - Administration avec tkined

Pour administrer les divers équipements du réseau, Tkined s'appuie essentiellement sur deux classes d'outils permettant d'obtenir des informations :

- les informations directement issues du protocole IP
- les informations fournies par le protocole SNMP

Un exemple d'utilisation des informations IP avec l'exploitation de la commande "ping" sur un des équipements administrés. Cherchant plus à exploiter SNMP, le but n'est pas de

s'étendre sur l'exploitation d'IP mais il peut être utile de savoir si un élément réseau ne répondant pas aux requêtes SNMP répond à un "ping".

## 1.4 - Utilisation de snmp dans tkined

Tkined implémente le protocole SNMP de façon à tirer tous les avantages offerts par ce protocole. Il permet notamment aux équipements de rendre compte de leurs états des deux façons envisagées dans SNMP :

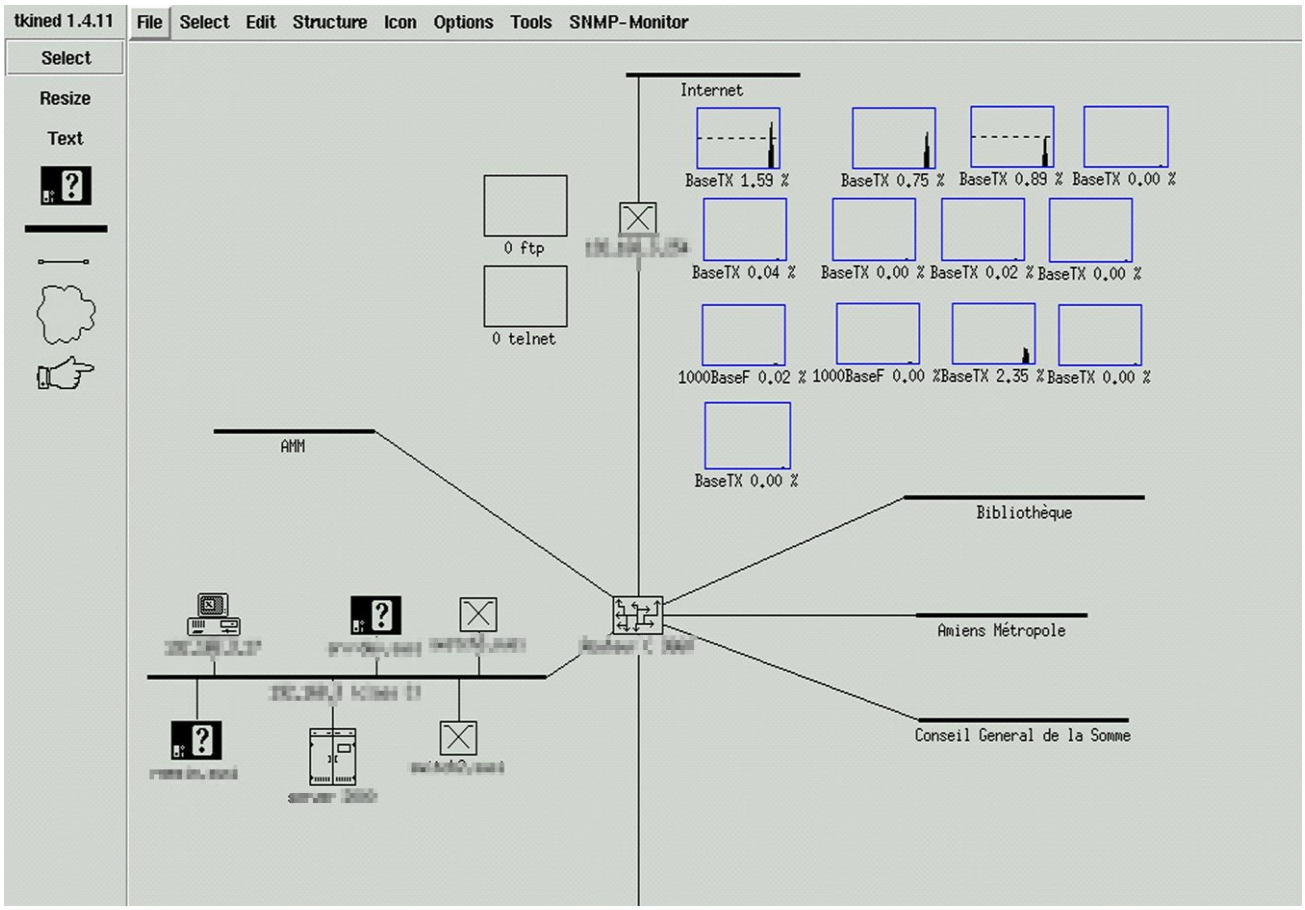
- Envoi de messages d'alerte ("trap message") en cas de détection d'un problème
- Envoi de messages suite à la réception d'une requête d'une station d'administration

L'utilisateur peut aussi visualiser la MIB pour sélectionner les variables de l'équipement qu'il souhaite observé. La possibilité de positionner des seuils pour les variables, avec envoi d'alerte en cas de dépassement, ainsi que la possibilité d'observer n'importe quelle variable de la MIB confère une grande puissance à Tkined.

# 2 - Exemples des possibilités de tkined

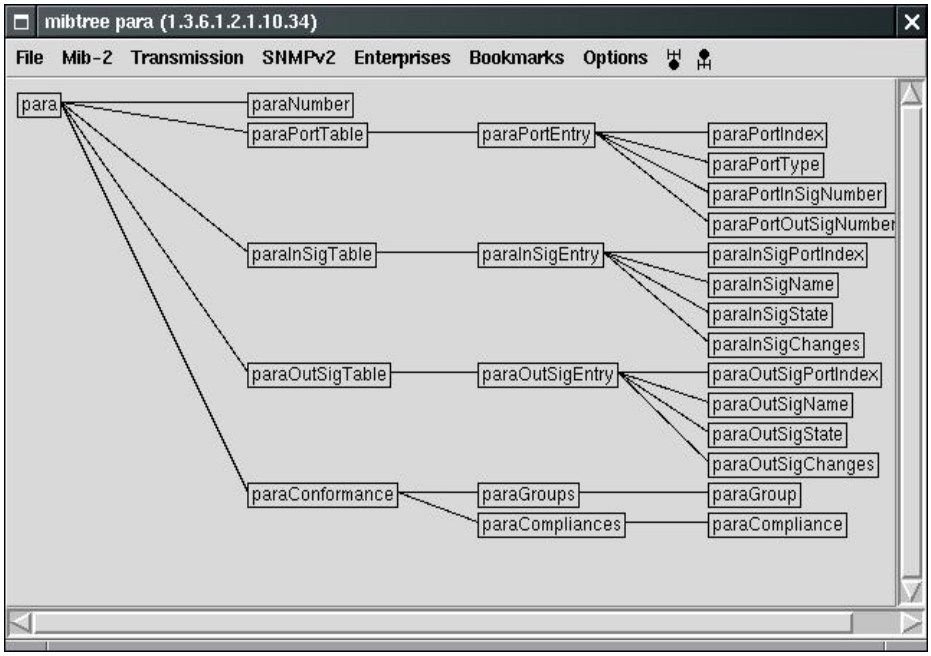
## 2.1 - Visualisation d'une map avec tkined

Une telle map réalisée avec tkined permet de visualiser très simplement le trafic de toutes les interfaces d'un switch ainsi que les connexions telnet et ftp en cours.



## 2.2 - Visualisation des variables mib

Cette fenêtre permet d'accéder très facilement à des variables **MIB**.



Chapitre 4

# UTILISATION AVANCEE DE SNMP

**Objectif :**

*Maitriser l'utilisation et l'administration des réseaux SNMP.*

**Éléments de contenu :**

- *Rappel*
- *Recherche de SNMP sur un Réseau*
- *Utilisation de ADMsnmp pour les communautés*
- *Description de la MIB*
- *UCD-SNMP pour accéder à la MIB sous Linux*

## 1 - Rappel sur SNMP

SNMP est avant tout une source d'informations sur le réseau en place et peut aussi être une source de problèmes pour un administrateur qui a mal configuré son matériel réseau... SNMP permet entre autre d'arrêter des interfaces (cartes) réseau ainsi que de tuer des connexions effectuées sur les équipements qui sont mal configurés.

***l'architecture:***

***- les agents*** : des équipements réseaux (switchs, routeurs, certains serveurs, ...) qui enregistrent des informations dans une base de données locale appelée Management Information Base (***MIB***).
Si les agents voient qu'un événement particulier se produit (interface réseau qui s'arrête ou qui démarre, erreur d'authentification, ...), ils émettent un datagramme appelé "trap" vers la station de Management pour la prévenir de l'évènement.

***- la station de Management*** : elle reçoit les traps envoyés, va chercher à intervalles réguliers l'information (polling) sur chacun des équipements gérés et permet à l'administrateur de les contrôler à distance (redémarrage, arrêt d'une interface, tuer des connexions TCP).

Et pour communiquer, ces deux ensembles utilisent le protocole SNMP.

L'administrateur organise les MIB par ***communautés*** ce qui lui permet de classer et protéger les informations. Tout ce qu'il pense pouvoir laisser accessible au plus grand nombre, il le met dans la communauté "**public**". Tout ce qu'il veut cacher, il le met dans la communauté "**private**" (et limite l'accès à cette base uniquement à la machine de Management) ou alors utilise un nom de communauté (comme "isri" par exemple).

Sur chacune des communautés qu'il définit, il attribue des droits (la communauté public ne devrait par exemple pas permettre la modification de données) et les adresses autorisées à y accéder (en général uniquement l'IP de la machine de supervision).

## 2 - Recherche de SNMP sur un réseau

Pour commencer un 'audit' de notre réseau il faut donc :

- ***Rechercher les agents***
- ***Identifier d'une part le nom de la communauté qui est accessible en écriture***
- ***Repérer l'adresse IP de la machine de Management.***

Le protocole SNMP utilise le protocole ***UDP*** pour toutes ses transmissions:
***les agents écoutent les requêtes sur le port 161 et le manager écoute les traps sur le port 162.***

Le scan se fait tout simplement grâce à **nmap** (http://nmap.org) : Considérons que nous voulons scanner toutes les machines du réseau 172.23.4.0/24

```
$ nmap -sU -p 161,162 172.23.4.*

Starting nmap

Interesting ports on (172.23.4.19):

Port                    State           Service

161/udp                 open            snmp

Interesting ports on (172.23.4.249):

Port                    State           Service

161/udp                 open            snmp

Interesting ports on (172.23.4.252):

Port                    State           Service

161/udp                 open            snmp

162/udp                 open            snmptrap
```

Apparemment nous avons 3 adresses qui font tourner les services SNMP et nous avons, à priori, repéré le Manager (172.23.4.252) puisqu'il a le port 162 ouvert. Les deux autres sont donc des agents.

## 3 - Les communautés

Nous allons donc voir si, parmi ces adresses, nous arrivons à trouver les noms de communautés.

Pour cela il existe un outil qui s'appelle **ADMsnmp**

Récupérez-le sur Internet : http://seclists.org/bugtraq/1999/Feb/329

Puis décompresser et compiler :

```
# tar xvzf ADMsnmp.0.1.tgz

ADMsnmp/

ADMsnmp/snmp.c

ADMsnmp/snmp.passwd

ADMsnmp/ADMsnmp.README

#cd ADMsnmp
#gcc -o ADMsnmp snmp.c
```

**ADMsnmp** permet de scanner une adresse IP pour les noms de MIB courant et teste si la communauté est accessible en écriture.

Rajoutez les noms que vous pensez plausibles dans le fichier snmp.passwd (comme "iset" par exemple).

Prenons deux Exemples d'exécutions de ADMsnmp :

```
# ./ADMsnmp 172.23.4.19
ADMsnmp vbeta 0.1 (c) The ADM crew
ftp://ADM.isp.at/ADM/
greets: !ADM, el8.org, ansia

...

>>>>>> get req name=private id = 11 >>>>>>

<<<<<<<<<<< recv snmpd paket id = 12 name =private ret =0 <<<<<<<<<<

>>>>>>>>>>>> send setrequest id = 12 name = private>>>>>>>>

<<<<<<<<<<< recv snmpd paket id = 13 name =private ret =0 <<<<<<<<<<

…

<!ADM!> snmp check on 172.23.4.19 <!ADM!>
sys.sysName.0:SWITCH1
name = private write access
```

Celui-ci permet l'écriture dans la zone private. On voit bien que c'est un switch

```
$ ./ADMsnmp 172.23.4.249
ADMsnmp vbeta 0.1 (c) The ADM crew
ftp://ADM.isp.at/ADM/
greets: !ADM, el8.org, ansia

...

>>>>>>>>>>> get req name=public id = 8 >>>>>>>>>>>

<<<<<<<<<<< recv snmpd paket id = 9 name = public ret =0 <<<<<<<<<<

>>>>>>>>>>>> send setrequest id = 9 name = public >>>>>>>>

<<<<<<<<<<< recv snmpd paket id = 10 name = public ret =0 <<<<<<<<<<

>>>>>>>>>>> get req name=private id = 11 >>>>>>>>>>>

<<<<<<<<<<< recv snmpd paket id = 12 name = private ret =0 <<<<<<<<<<

>>>>>>>>>>>> send setrequest id = 12 name = private >>>>>>>>

<<<<<<<<<<< recv snmpd paket id = 13 name = private ret =0 <<<<<<<<<<

...

<!ADM!> snmp check on 172.23.4.249 <!ADM!>
sys.sysName.0:SRVWEB
name = public readonly access
name = private readonly access
```

Celui-ci est un peu mieux paramétré ! Mais on peut quand même récupérer les informations présentes dans ses bases "public" et "private"

Apparemment, c'est un serveur web.

Passons maintenant à la station de Management.

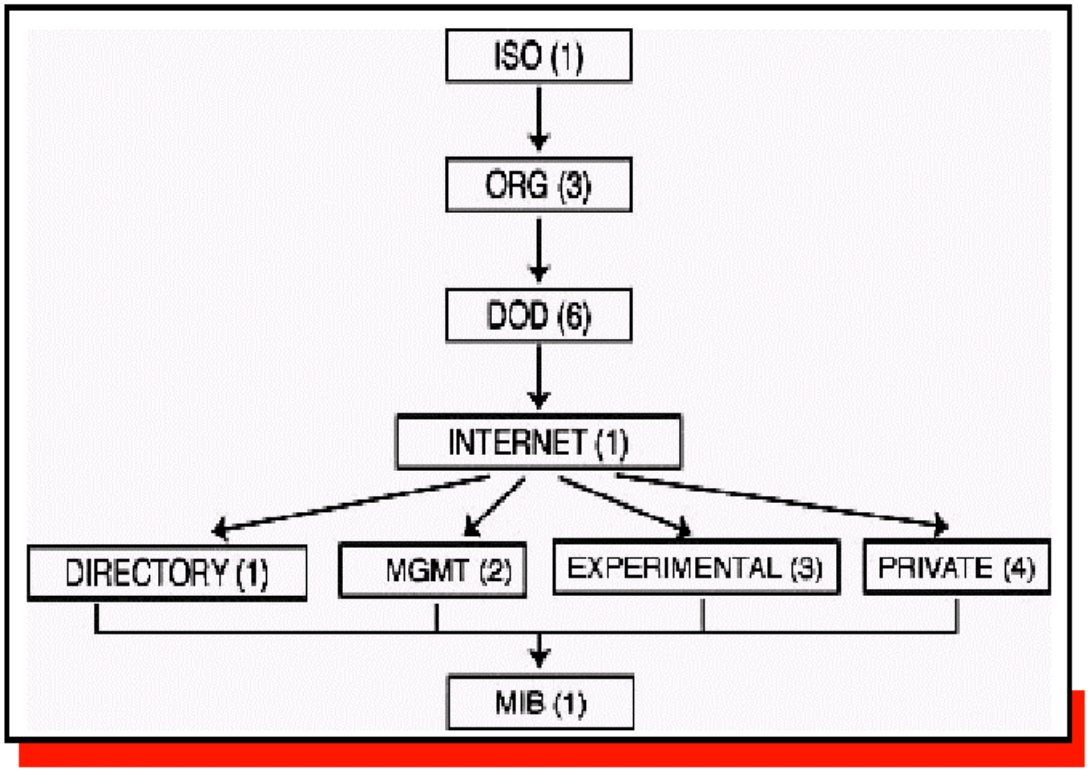
```
$ ./ADMsnmp 172.23.4.252
ADMsnmp vbeta 0.1 (c) The ADM crew
ftp://ADM.isp.at/ADM/
greets: !ADM, el8.org, ansia

..

>>>

>>>>>>>> get req name=public id = 8 >>>>>>>>>>>

>>>>>>>>>>> get req name=private id = 11 >>>>>>>>>>>

..

<!ADM!> snmp check on 172.23.4.252 <!ADM!>
```

Sur celui-ci aucune information n'est disponible.

Donc, nous avons trouvé deux noms de communauté: public et private.

## 4 – La MIB

Les données sont classées dans la base par un système d'arbre de classement :




Donc, pour arriver à accéder aux données de la MIB, il faut en théorie spécifier toute la chaîne partant de la racine de l'arbre jusqu'à la position de la donnée MIB que l'on souhaite accéder, soit 1.3.6.1.2.1 pour accéder à iso.org.dod.internet.mgmnt.MIB.

Il existe deux types de données : les objets de type variable simple et ceux de type table. En règle générale, les objets de type tableau s'appellent xxxxTable et ils sont composés d'éléments de type xxxxxEntry.

Sous la branche MIB, nous avons encore plusieurs catégories, et c'est ci que nous allons nous attarder :

***- System(1) :*** Contient l'identifiant de la machine (*sysObjectID(2)*), sa description (*sysDescr(1)*) qui permet de savoir à quel type d'équipement nous avons affaire, le nom du responsable(*sysContact(4)*), le nom de l'équipement(*sysName(5)*), sa localisation (*sysLocation(6)*).

***- Interfaces(2):*** Nous allons en savoir un peu plus sur la configuration matériel de l'équipement :

ifNumber(1) fournit le nombre d'interfaces réseau.
ifTable(2): un tableau d'interfaces (ifEntry(1)). Une ifEntry est constituée de :

ifIndex(1): numéro identifiant de cette interface dans la table des interfaces.

ifDescr(2): description de cet interface (ça peut permettre de comprendre le rôle de cet équipement, notamment pour les routeurs.

ifType(3): le type de support physique. Si c'est de l'ethernet, ca sera ethernet-csmacd(6).

ifMtu(4): la Maximum Transmit Unit (la taille max au delà de laquelle un paquet doit être fragmenté pour pouvoir passer sur le réseau).

ifSpeed(5): bande passante en bits/s.

ifPhysAddress(6): @Phys (MAC pour les réseaux ether-net).

ifAdminStatus(7): statut désiré de l'interface : on souhaite qu'elle fonctionne (up(1)) ou qu'elle soit arrêtée (down(2)) ?

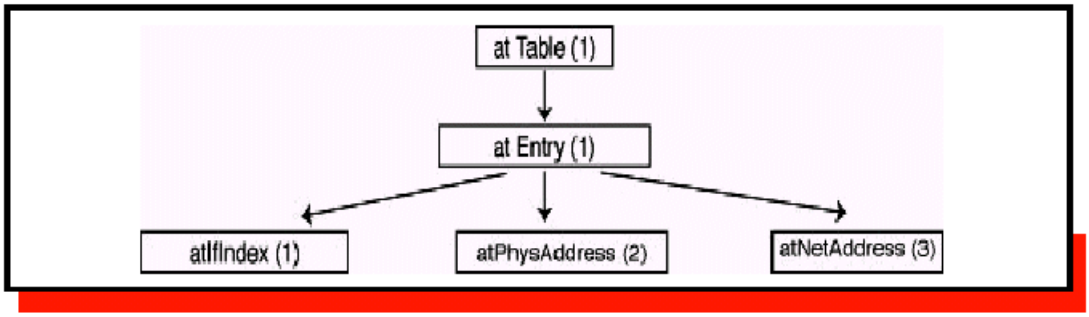
ifOperStatus(8): idem ifAdminStatus sauf que là c'est le statut réel de l'interface.

Lorsque vous souhaiterez arrêter une interface (pour "débrancher" toute une partie du réseau par exemple), vous devrez effectuer la requête sur la donnée ifAdminStatus et non pas sur ifOperStatus qui est en lecture seule.

## 5 - Autres variables statistiques

*- Address Translation(at(3)) :* contient les données liées à la résolution ARP.

atTable(1): table d'entrées de type atEntry(1) :




Ici nous trouverons la table ARP de l'équipement.

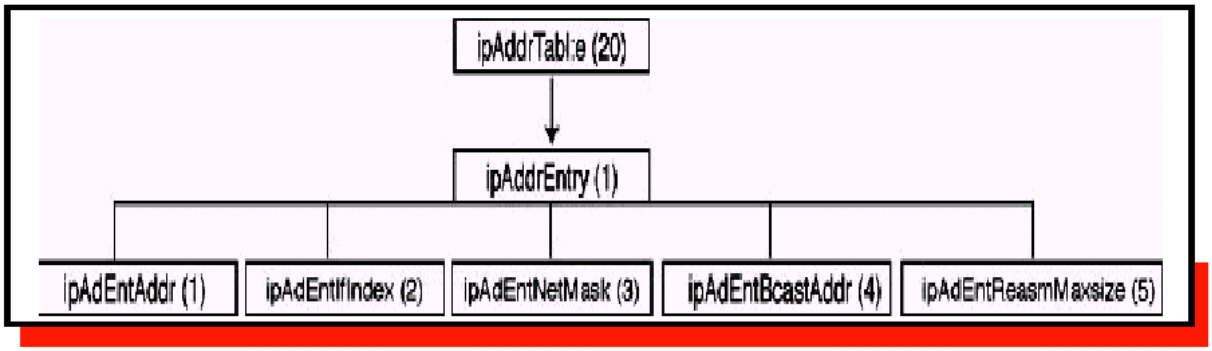
AtPhysAddress(2) est l'adresse Physique, atNetAddress l'adresse IP associée.

***- IP(4) :*** est parmi les plus intéressantes :

ipForwarding(1): l'équipement sert de passerelle si la valeur est à 1. Elle peut être à 2 dans le cas contraire.

ipDefaultTTL(2): permet de connaître le TTL par défaut. C'est le temps qu'un paquet fragmenté est conservé (en attente de réception de tous les fragments) avant destruction. Cette variable est read-only (on ne peut pas la modifier).

ipAddrTable(20): table des adresses IP de l'équipement.




Elle est constituée comme suit :
Grâce à ipAddrTable, on peut récupérer l'adresse IP de toutes les interfaces de l'équipement, le masque de sous-réseau associé, l'adresse de broadcast (pour écrire à toutes les machine du sous-réseau).

ipRouteTable(21): la table de routage de cette machine, constituée de ipRouteEntry(1)

ipRouteDest : renseigne sur la destination associée à cette route (0.0.0.0 pour la route par défaut).

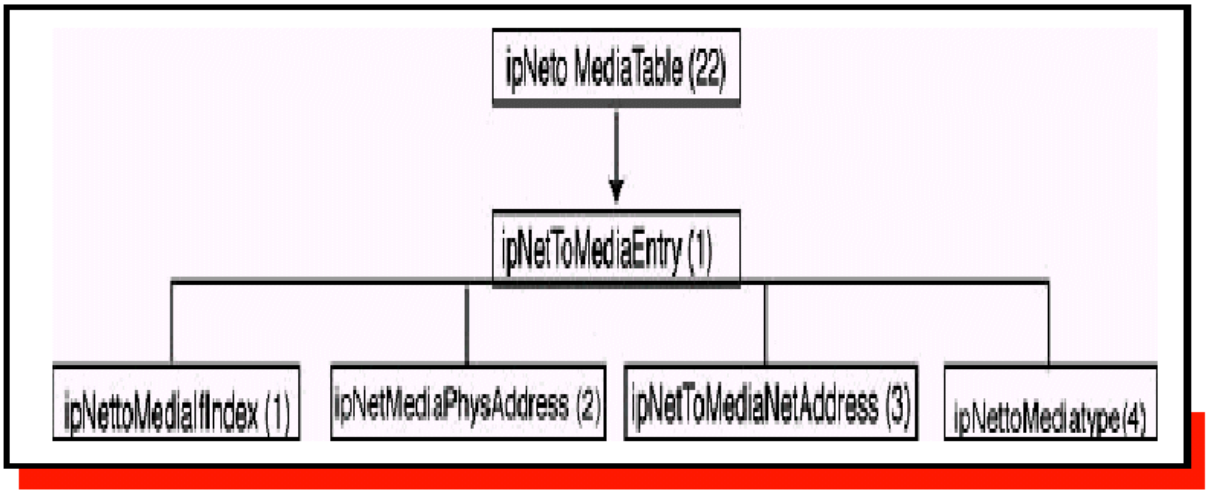
ipRouteIfIndex donne l'identifiant de interface qui sera utilisée pour retransmettre le paquet sur le bon réseau.

ipRouteNextHop donne l'adresse IP du prochain routeur pour cette route.

ipRouteType nous indique si le réseau de destination est connecté sur l'interface destination (direct(3)) ou sur un réseau non accessible localement(indirect(4)).

ipRouteMask(11) spécifie le masque réseau associé à cette route.

Il existe aussi ipRouteProto(9) qui est au même niveau que ipRouteNextHop et qui peut être utile pour savoir quels protocoles réseaux de configuration de route sont pris en compte sur ce réseau (valeurs notoires : local(2) (route mise en dur par l'administrateur), icmp(4) (icmp redirect !), egp(5), rip(8), ospf(13)).




ipNetToMediaTable(22): table des correspondances adresse IP/adresse physique (adresse MAC). Ici nous pouvons récupérer l'adresse physique des interfaces locales et des autres adresses IP connues, ainsi que le type d'attribution de l'adresse IP (dynamique si ipNetToMedia-Type==3 ; si égal à 4, l'attribution est statique).

***- IMCP(5) :*** Nous ne détaillerons pas, ce sont des statistiques (pas forcément inutiles d'ailleurs).

***- TCP(6) :*** Fournit une sorte de netstat (liste toutes les connexions ouvertes et les serveurs en écoute) :

tcpMaxConn(4) est le nombre maximum de connexions TCP que l'équipement peut accepter.
tcpConnTable(13) est une table contenant des tcpConnEntry(1) qui sont composées de :

tcpConnState(1): état de la connexion (1: fermée, 2: en écoute (serveur), 5: établie, 10: fermée, 12: fermeture de la connexion ordonnée par le manager (dele-teTCB))

tcpConnLocalAddress(2): adresse locale pour cette connexion (0.0.0.0 pour un serveur acceptant des connexions depuis n'importe quelle interface)

tcpConnLocalPort(3): port local

tcpConnRemAddress(4): @IP destinataire de cette connexion

tcpConnRemPort(5): port de destination.

Ces informations nous permettent de savoir si des serveurs écoutent sur cette machine, et avec quelles machines cet équipement est actuellement en relation.

***- UDP(7) :*** permet de récupérer le même style d'information pour le protocole UDP.

udpTable(5): contient la liste des ports (udpEntry(1)) sur lesquelles une application écoute et attends la réception de données en UDP.

une udpEntry contient les champs suivants :
udpLocalAddress(1): l'adresse de l'interface sur laquelle le serveur "écoute".
udpLocalPort(2): le port local sur lequel l'application "écoute".

De la même manière qu'avec TCP, on peut ainsi connaître les ports ouverts sur chaque interface sans avoir à scanner le réseau.

***- EGP(8) :*** idem ICMP

## 6 - UCD-SNMP sous Linux

Le paquetage **UCD-SNMP** fournit des outils permettant d'accéder en lecture/écriture aux données des agents.

Ce sont les commandes ***snmpget*** (pour récupérer une valeur), ***snmpwalk*** (pour récupérer toutes les valeurs des données situées à partir d'un certain point de l'arborescence), ***snmptable*** (pour récupérer des variables de type table) et ***snmpset*** (pour modifier la valeur de variable)

Voyons leur fonctionnement :

Nous allons interroger la valeur de la variable 1.5 (system.sysName) => il faut rajouter .0 à la fin car c'est une variable de type simple:

```
$ snmpget 172.23.4.19 private 1.5.0
system.sysName.0 = SWITCH1
```

Utilisons snmpwalk sur toute l'arborescence débutant à « sys »

```
$ snmpwalk 172.23.4.19 private 1
system.sysDescr.0 = BayStack 450-24T HW:RevL FW:V1.46 SW:v3.0.0.41 ISVN:0
system.sysObjectID.0 = OID: enterprises.45.3.35.1
system.sysUpTime.0 = Timeticks: (235697570) 27 days, 6:42:55.70
system.sysContact.0 = Administrateur
system.sysName.0 = SWITCH1
system.sysLocation.0 = Amiens
system.sysServices.0 = 3
system.sysORLastChange.0 = Timeticks: (0) 0:00:00.00
```

Pour la modification, voici les principaux types : ("man snmpset" pour tous les types)

« i » ENTIER
« s » CHAINE DE CARACTERES
« a » ADRESSE IP

On modifie cette valeur (le « s » est pour STRING car le champ en question est de type CHAINE DE CARACTERES) :

```
$ snmpset 172.23.4.19 private 1.5.0 s "SWITCH ISET"
system.sysName.0 = SWITCH_ISET
```

Récupération de table:

```
$ snmptable 172.23.4.19 private 4.20
SNMP table: ip.ipAddrTable
ipAdEntAddr ipAdEntIfIndex ipAdEntNetMask ipAdEntBcastAddr
ipAdEntReasmMaxSize
172.23.4.19 1 255.255.254.0 1 1512
```

Ici nous n'avons qu'une seule interface, mais nous pouvons voir que son adresse IP est 172.23.4.19 et que son masque de sous-réseau est 255.255.254.0.

Donc en explorant la MIB, vous pouvez recueillir quasiment toutes les informations que vous recherchez.

Rappelez-vous que lorsque vous tapez les identifiants 1.5.0, le système considère que vous vous placez déjà dans le chemin suivant : iso.org.dod.internet.mgmt.mib.

Vous n'avez plus qu'à taper l'identifiant de sys, ip, tcp, udp, interface, at et les champs nécessaires pour atteindre la variable recherchée.

Vous pouvez également taper l'ensemble des identifiants à partir de la racine en démarrant par un '.' (À ce moment là, il faut donner tous les champs .iso.org.dod....).

Quelques exemples de commandes :

Fermeture d'une connexion TCP (172.23.4.237:2524 =>172.23.4.19:23) : (tapez "snmpwalk 172.23.4.19 private tcp.tcpConnTable.tcpConnEntry" pour avoir une liste des connexions)

```
$ snmpset 172.23.4.19 private
tcp.tcpConnTable.tcpConnEntry.tcpConnState.172.23.4.19.23.172.23.4.237.2524 i
12
```

(La valeur 12 correspond à deleteTCP (fermeture de la connexion demandée par le manager)

Arrêt d'une interface réseau :

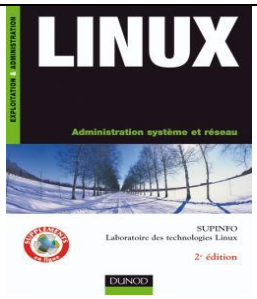
```
$ snmpset 172.23.4.19 private interfaces.ifTable.ifEntry.ifAdminStatus.4 i 2
interfaces.ifTable.ifEntry.ifAdminStatus.4 = down(2)
```

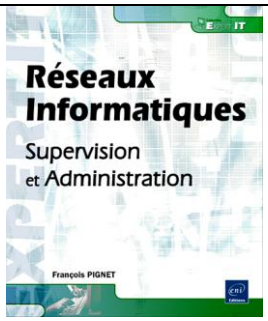
## DS et Examens

Consulter le site : http://www.isetinfo.com

## BIBLIOGRAPHIE ET WEBOGRAPHIE

### *OUVRAGES*





| | | |
|---|---|---|
| 🕮 | **Titre** | **: "Administration système et réseau"** |
| | **Maison d'édition** | **:** DUNOD |
| | **Auteur** | **:** Laboratoire SUPINFO des technologies LINUX |
| | **Année** | **:** 19/03/2008 |
| 🕮 | **Titre** | **: "Supervision et Administration"** |
| | **Maison d'édition** | **:** ENI |
| | **Auteur** | **:** François Pignet |
| | **Année** | **:** 10/12/2007 |
| 🕮 | **Titre** | **: "Essential SNMP"** |
| | **Maison d'édition** | **:** O'REILLY |
| | **Auteur** | **:** Douglas R. MAURO, Kevin J. SCHMIDT |
| | **Année** | **:** 23/09.2005 |

# SITES INTERNET

| | | |
|---|---|---|
| | **Site 1** | : **http://www.misfu.com/information-sur-le-fichier-196.html**<br>Ce site nous propose un très bon support de Cours SNMP |
| | **Site 2** | **: http://christian.caleca.free.fr/snmp/principe.htm**<br>Ce site a été fait par un expert dans le domaine systèmes et réseaux, offre une excellente alternative par rapport aux autres sites. |
| | **Site 3** | **: http://www.irisa.fr/prive/bcousin/Cours/**<br>Ce site propose plusieurs cours, TP, TD et examens en formats PDF dans le domaine des services réseaux. |