REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE D'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
Université Mouloud Mammeri Tizi-Ouzou
Faculté de Génie Electrique et Informatique
Département d'informatique

Mémoire de fin d'études

En vue de l'obtention du grade de Master II en Informatique
Option: Ingénierie des Systèmes Informatiques

Thème:

Conception et réalisation d'une application de surveillance réseau basée sur SNMP

Proposé et dirigé par:
Mr: RAMDANE MOHAMED

Réalisées par :
Mr: Santos Hugo Antonio da Costa Dias dos
Mlle: Jeremias Sara Caridade Maria

Présenté devant le Jury composé par:
Président:. . . . . .
Examinateur:. . . . . .
Examinateur:. . . . . .
Promoteur: Ramdane Mohamed

Année universitaire 2012/2013

Conception et réalisation d'une application de surveillance réseau basée sur snmp

Dédicace

Nous dédions ce travail

A tous ceux qui nous aiment et nous aimons...

A tous ceux à qui on compte pour eux et ils comptent pour nous...

A tous ceux qui se sentent participants dans notre réussite...

Remerciements

Nous tenons à remercier dieu le tout puissant qui nous a donné le courage et la patience qui a éclairé notre chemin pour achever ce travail.

Nous remercions très vivement notre Promoteur Mr: Ramdane pour nous avoir encadré, suivi et soutenu dans ce parcours, pour nous avoir conseillé, encouragé et fait partager son expérience et ses connaissances.

Nous n'omettrons jamais d'exprimer toute notre gratitudes à tout le staff de la faculté de génie électrique et informatique, que ce soit mes collègues, enseignants ou cadres administratifs, qui de prés ou de loin n'ont épargné aucun effort pour que notre formation et notre travail se termine dans les bonnes conditions.

Nous tenons aussi à remercier en particulier tous les étudiants du Master II ISI de l'année 2012/2013 qui nous ont aidé de très près pendant notre parcours.

Nous tenons aussi à remercier vivement les membres du jury pour l'honneur qu'ils nous ont fait en acceptant de juger ce travail.

Enfin, nous remercions nos parents, toute notre famille et nos amis pour leur soutien et leurs encouragements.

Sommaire

Table des Figures

Liste des Tableaux

Liste des Graphes

Acronymes

ARP: Address Resolution Protocol

ASCII: American Standard Code for Information Interchange

ASN.1: Abstract Syntax Notation 1

BER: Basic Encoding Rule

CSMA/CD: Carrier Sense Multiple Access

CSMA/CD: Carrier Sense Multiple Access with Collision Detection

DES: Data Encryption Standard

EGP: External Gateway Protocol

FDDI: Fiber Distributed Data Interface

FTP: File Transfer Protocol

HTTP: HyperText Transfer Protocol

IANA: Internet Assigned Numbers Authority

ICMP: Internet Control Message Protocol

IETF: Internet Engineering Task Force

IP: Internet Protocol

ISO: International Organisation for Standardization

ITU-T: International Télécommunications Union-Telecommunications

LAN: Local Area Network

MAN: Metropolitan Area Network

MAU: Multistation Access Unit

MD5: Message Digest 5

MIB: Management Information base

MN: Managed Node

NMS: Network Management Station

OID: Object Identifier

OS: Operating System

OSI: Open System Interconnexion

P2P: Peer to Peer

SMI: Structure of Management Information

SNMP: Simple Network Management Protocol

TCP: Transmission Control Protocol

UDP: User Datagram Protocol

WAN: Wide Area Network

INTRODUCTION GÉNÉRALE

Les réseaux informatiques ont aujourd'hui autant d'importance que les ordinateurs eux mêmes, au point que la plupart de nos activités ne pourraient plus être envisagées sans la mise en place de ces réseaux. On assiste à leur déploiement à tous les niveaux de la société, dans les entreprises, au niveau national et international, y compris dans les domiciles. Quant aux entreprises, ces réseaux leur apportent un moyen efficace pour mettre en œuvre un travail coopératif, pour faire communiquer des ordinateurs distants, pour partager des données, mais aussi pour imprimer à distance, envoyer des messages, et accéder à des bases de données délocalisées.

Devant la véritable explosion des réseaux et leur importance primordiale dans une infrastructure, les besoins de superviser et surtout de diagnostiquer rapidement les problèmes sont devenus des préoccupations majeures. Le protocole de supervision réseau SNMP se propose de répondre à ce double problématique.

Ce travail se concentre sur la configuration et l'utilisation de protocole SNMP (Simple Network Management Protocol), qui est un protocole de monitoring et de configuration réseau à distance. De nombreux matériels et systèmes d'exploitation incluent la prise en charge de SNMP, la version dépendant des choix des concepteurs.

Le but de notre projet est l'étude du protocole de gestion SNMP qui permettra de mettre en place une interface de gestion et surveillance SNMP d'un réseau.

- Dans le « chapitre 1 » on présente les concepts de base sur les réseaux ainsi que sur les éléments matériels nécessaires. Pour notre part, nous nous intéresserons qu'au protocole SNMP qui est actuellement l'un des protocoles les plus utilisés dans les réseaux.
- Au « chapitre 2 », nous étudierons la MIB (Management Information Base) qui est une base de données ASCII contenant des informations qui le protocole SNMP exploite pour la surveillance et l'administrations des agents SNMP.
- Dans le « chapitre 3 », nous présenterons dans le détail le protocole SNMP. On va présenter les différentes versions SNMP tout en précisant ce qu'elles apportent de plus par rapport à la version de base. Notre principal intérêt portera sur les différentes requêtes SNMP utilisées entre les agent et le Manager.
- Dans le « chapitre 4 » on ferra la conception, c'est-à-dire, la méthode d'analyse de notre application brièvement expliqué.
- Le « chapitre 5 » sera consacré à la réalisation d'une application de surveillance réseau en java basée sur le protocole SNMP et à l'installation et activation des agents SNMP sur Windows (XP, Vista et Seven), sur Linux Ubuntu et sur un routeur Cisco. L'application en java permettra de mettre en évidence les fonctionnalités de SNMP en questionnant, configurant, et surveillant les agent SNMP d'une infrastructure réseau.

De plus nous montrerons les données obtenus pendant la phase de test de notre application en ce qui concerne: le nombre de configurations faites au niveau des agents, le nombre de consultations de la MIB des agents, le nombre de Traps reçus par le Manager SNMP, le nombre des Traps envoyés par les agents, le nombre de déconnections des agents et les statistiques du trafic SNMP sur le réseau.

Chapitre I

Généralités sur les Réseaux

INTRODUCTION

L’évolution technologique de ces dernières années a conduit à la possibilité de construire des systèmes informatiques de plus en plus sophistiqués et de moins en moins encombrants pour permettre d’équiper le maximum de points d’utilisation et constituer, pour tout employé d’une entreprise, d’une administration, et donc d’un établissement d’enseignement, l’outil indispensable améliorant son efficacité et par la suite sa productivité.

De plus en plus, ce sont les réseaux qui nous relient. Les personnes communiquent en ligne depuis n’importe où. Une technologie efficace et fiable permet au réseau d’être disponible n’importe quand et n’importe où. Alors que notre réseau humain continue de s’étendre, la plateforme qui relie ce réseau et le prend en charge doit également se développer.

Au lieu de développer des systèmes uniques et distincts pour fournir chaque nouveau service, l’industrie du réseau dans son ensemble a développé des moyens pour à la fois analyser la plateforme existante et l’améliorer petit à petit. Ainsi, les communications existantes sont maintenues tandis que de nouveaux services sont intégrés, économiques et bénéficiant d’une technologie fiable.

Dans le cadre de ce chapitre, nous allons présenter quelques notions sur les réseaux informatiques.

1. Les réseaux informatiques

1.1. Définition

Un ***réseau*** est un ensemble d'équipements informatiques reliés physiquement entre eux par un support de transmission (en général des câbles) et qui peuvent communiquer. Ces équipements informatiques sont appelés des nœuds ou encore des stations (des ordinateurs, mais aussi des imprimantes et des boîtiers d'interconnexion).

Une connexion réseau ne nécessite pas forcément un câble en cuivre (une ligne physique), mais elle peut être réalisée par laser (infrarouge), par ondes courtes ou par satellite.

1.2. Avantages d'un réseau

La mise en réseau d'ordinateurs présente des avantages dans les domaines suivants : partage des informations, partage des applications et des équipements matériels et support administratif. Ces avantages contribuent à accroître la productivité.

a) Partage des informations

La possibilité de partager rapidement et à moindre coût des informations et des données constitue l'avantage principal des réseaux. De nombreuses entreprises utilisent aujourd'hui un réseau pour prendre en charge leurs services de messagerie électronique ou leurs tâches de planification.

b) Partage du matériel et des logiciels

Avant l'avènement des réseaux, chaque utilisateur devait disposer de sa propre imprimante et autres périphériques, ce qui pouvait se révéler extrêmement coûteux pour une grande entreprise. Grâce aux réseaux, ces coûts ont été réduits de façon spectaculaire, car plusieurs utilisateurs peuvent désormais partager un équipement ou une application.

c) Administration et support centralisés

Les réseaux simplifient également les tâches d'administration et de support. En effet, l'administrateur système peut effectuer des tâches administratives depuis tout ordinateur du réseau. Il est également plus simple pour le personnel technique d'assurer le support d'une seule installation d'un système d'exploitation ou d'une application plutôt que d'avoir à superviser une multitude de versions et de configurations.

1.3. Composantes d'un réseau

Le chemin emprunté par un message depuis une source jusqu'à une destination peut être aussi simple que la connexion entre deux ordinateurs via un seul câble ou aussi complexe qu'un réseau parcourant le globe terrestre. Cette infrastructure réseau constitue la plateforme qui prend en charge notre réseau humain. Elle fournit le canal stable et fiable à travers lequel nos communications peuvent s'établir.

Les périphériques et les supports représentent les éléments physiques ou le matériel du réseau. Le matériel correspond souvent aux composants visibles de la plateforme réseau,

tel qu'un ordinateur portable, un ordinateur de bureau, un commutateur, ou le câblage qui sert à relier les périphériques. Parfois, certains composants ne sont pas visibles. Dans le cas d'un support sans fil, les messages sont transmis à travers l'air, à l'aide d'une fréquence radio ou d'ondes infrarouges invisibles.

Les services et les processus constituent les programmes de communication, appelés logiciels, qui sont exécutés sur les périphériques réseau. Un service réseau fournit des informations en réponse à une demande. Les services incluent de nombreuses applications réseau courantes que les personnes utilisent quotidiennement, telles que les services d'hébergement de messagerie et les services d'hébergement Web. Les processus fournissent les fonctionnalités qui dirigent et déplacent les messages à travers le réseau. Les processus nous semblent moins évidents, mais ils sont essentiels au fonctionnement des réseaux.

1.4.Commutation

Il existe 2 grandes techniques pour permettre à 2 entités de communiquer :

I) **La commutation de circuits** (utilisée par exemple pour le téléphone) qui affecte un circuit de transmission entre les 2 machines communicantes. Le principe d'une communication est le suivant :

- ✓ On envoie un signal pour établir le circuit
- ✓ Établissement de la communication
- ✓ Libération du circuit en fin de communication

L'avantage de cette méthode est qu'elle garantit la performance de la communication, puisque le circuit est entièrement disponible pour une communication, mais elle coûte chère et est peu évolutive, si aucun circuit n'est disponible pas de communication, si on ne parle pas on paye quand même !

II) **La commutation de paquets :** On découpe l'information à transmettre en petits paquets qui vont chacun contenir l'adresse de destination et l'adresse d'origine et une partie des informations à transmettre. Les paquets transitent indépendamment les uns des autres sur un support qui peut transmettre d'autres paquets destinés à d'autres machines. La machine qui reçoit remet les paquets en ordre. Les performances se dégradent avec la charge sur le réseau mais on a un meilleur rendement du support qui n'est pas monopolisé par une seule communication, on peut faire tourner les programmes de gestion de réseau en tâche de fond.

Les composants électroniques des cartes réseaux valident la réception de ces paquets. Quel que soit le type de réseau, ces paquets "élémentaires" dont les données à transmettre sont "encadrées" par des informations de contrôle du réseau, sont appelés des *trames (frame).* La méthode adoptée par la plupart des systèmes de réseau est la *commutation de paquets.*

1.5.Système d’exploitation réseau

Le noyau d'un réseau réside dans son système d'exploitation. De la même manière qu'un ordinateur ne peut pas fonctionner sans un système d'exploitation, un réseau ne peut pas fonctionner sans un système d'exploitation réseau. Celui-ci assure les services de base aux ordinateurs sur le réseau. Ces services sont les suivants :

- ✓ coordination des activités des différents périphériques sur le réseau afin d'assurer l'exécution de la communication au moment et selon le mode requis ;
- ✓ accès des clients aux ressources réseau, notamment aux fichiers de données et aux périphériques, comme les imprimantes et les télécopieurs ;
- ✓ sécurité des données et des périphériques sur le réseau.

1.5.1. Fonctionnalités des systèmes d'exploitation réseau

Un système d'exploitation réseau doit prendre en charge les mécanismes qui permettent aux applications de communiquer les unes avec les autres : par exemple, les applications qui permettent à plusieurs ordinateurs de participer conjointement à une tâche unique, comme un calcul mathématique. Un système d'exploitation réseau doit également prendre en charge plusieurs processeurs, les clusters de lecteurs de disque et les fonctions de sécurité des données. Enfin, un système d'exploitation réseau doit être fiable et pouvoir récupérer les données rapidement en cas d'erreur.

1.6. Types de réseaux

Lorsque l'on parle de réseau informatique, il faut distinguer 3 types de réseaux dont les limites ne sont pas fixées de manière absolue et qui peuvent former, ensemble, un réseau d'entreprise.

1.6.1. Les réseaux locaux (LAN: Local Area Network)

Ce sont des réseaux individuels s’étendant généralement sur une zone géographique unique et fournissant des services et des applications aux personnes au sein d’une structure organisationnelle commune, telle qu’une entreprise, un campus ou une région. En règle générale, un réseau local est administré par une organisation unique. Le contrôle administratif qui gère les stratégies de sécurité et de contrôle d’accès s’applique au niveau du réseau. De tels réseaux offrent en général une bande-passante comprise entre 4Mbit/s et 100 Mbits/s.

1.6.2. Les réseaux métropolitains (MAN: Metropolitan Area Network)

Ce type de réseau est apparu relativement récemment et peut regrouper un petit nombre de réseau locaux au niveau d'une ville ou d'une région. L'infrastructure peut être privée ou publique. Par exemple, une ville peut décider de créer un 'MAN' pour relier ses différents services disséminés sur un rayon de quelques kilomètres et en profiter pour louer cette infrastructure à d'autres utilisateurs. La bande-passante peut être de quelques centaines de Kbits/s à quelques Mbits/s.

1.6.3. Les réseaux distants (WAN: Wide Area Network)

Les organisations individuelles utilisent généralement des connexions via un réseau de fournisseurs de services de télécommunications. Ces réseaux qui connectent des réseaux locaux et métropolitains à des emplacements géographiquement séparés sont appelés réseaux étendus (WAN, Wide Area Networks). Bien que l'organisation gère l'ensemble des stratégies et de l'administration des réseaux locaux aux deux extrémités de la connexion, les stratégies au sein du réseau du fournisseur de services de communications sont gérées par le fournisseur de services de télécommunications.

Les réseaux étendus utilisent des périphériques réseau spécialement conçus pour effectuer les interconnexions entre les réseaux locaux. En raison de l'importance de ces périphériques sur le réseau, la configuration, l'installation et la gestion de ces périphériques sont des domaines qui font partie du fonctionnement du réseau d'une organisation.

Les modems sont un des éléments de base des WANs.

La bande-passante va de quelques kbits/s à quelques Mbit/s. Une valeur typique pour une ligne louée est de 64kbits/s (en fonction des services offerts).

1.7.Topologies

Les différentes manières de faire partager à plusieurs machines le même support de transmission sont couramment appelées les topologies du réseau. Historiquement les topologies de réseaux étaient étroitement liées avec l'architecture physique du câblage. Ce n'est plus le cas aujourd'hui. Les topologies de réseaux sont des représentations simplifiées de la façon dont circulent les paquets sur un réseau et sont liées au type d'interface réseau qui est installés dans les ordinateurs.

1.7.1. Topologies physiques

La topologie physique décrit la mise en pratique du réseau logique (câblage etc.)

a) Topologie en bus :

Dans une topologie en bus, chaque ordinateur d'un réseau est connecté à un câble continu, ou segment, qui connecte la totalité du réseau en ligne droite. Dans ce type de topologie, un paquet est transmis à toutes les cartes réseau du segment. En raison du mode de transmission des signaux électriques sur ce câble, les extrémités de ce dernier doivent être terminées par des périphériques appelés terminaisons, qui représentent les limites du signal et définissent le segment. En cas de rupture en un point du câble ou d'absence de terminaison sur l'une des extrémités, le signal effectuera un aller-retour continuel sur le réseau, et toutes les communications seront interrompues. Le mot « bus » désigne la ligne physique qui relie les machines du réseau.

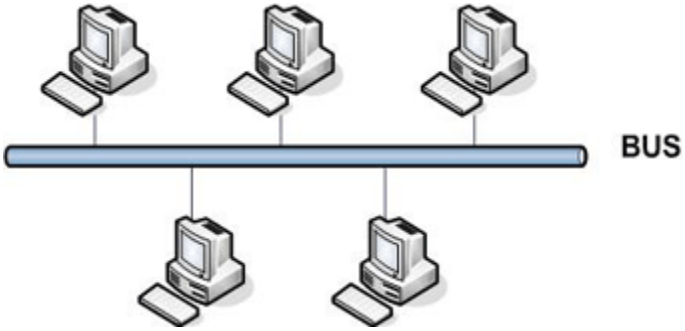


Fig1.1. Topologie en bus.

Cette topologie a pour avantage d'être facile à mettre en œuvre et de posséder un fonctionnement simple. En revanche, elle est extrêmement vulnérable étant donné que si l'une des connexions est défectueuse, l'ensemble du réseau en est affecté.

b) Topologie en étoile :

Dans une topologie en étoile, les segments de câble de chaque ordinateur sur le réseau sont connectés à un composant central, ou concentrateur. Un concentrateur est un périphérique qui raccorde plusieurs ordinateurs. Dans une topologie en étoile, les signaux sont transmis de l'ordinateur au concentrateur, et de ce dernier à tous les ordinateurs du réseau. À grande échelle, plusieurs réseaux locaux peuvent être interconnectés dans une topologie en étoile. Le principal avantage de la topologie en étoile est que si un ordinateur tombe en panne, il est le seul à ne plus pouvoir transmettre ou recevoir des données. Le reste du réseau fonctionne normalement.

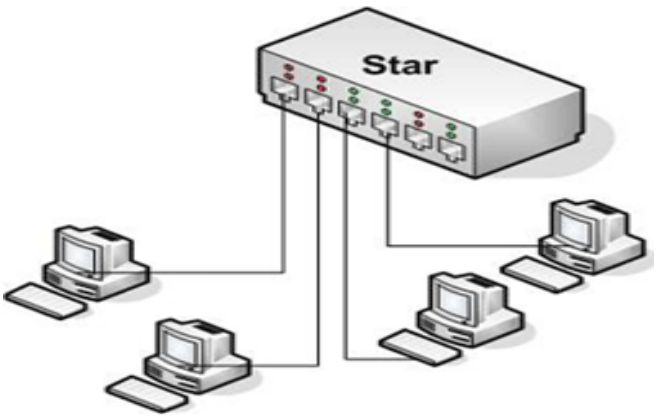


Fig.1.2. Topologie en étoile.

L'inconvénient de cette topologie est qu'en cas de défaillance du concentrateur, l'ensemble du réseau est en panne, puisque tous les ordinateurs lui sont connectés.

De plus, la topologie en étoile génère du bruit sur le réseau.

c) Topologie en anneau :

Dans une topologie en anneau, les ordinateurs sont reliés par un seul câble en anneau. Contrairement à la topologie en bus, elle ne contient pas d'extrémités terminées. Les signaux transitent dans une seule direction selon une boucle, en

passant par chaque ordinateur, qui joue le rôle de répéteur pour régénérer le signal avant de le transmettre à l'ordinateur suivant. À grande échelle, plusieurs réseaux locaux peuvent être interconnectés dans une topologie en anneau, en utilisant un câble coaxial ThickNet ou à fibres optiques.

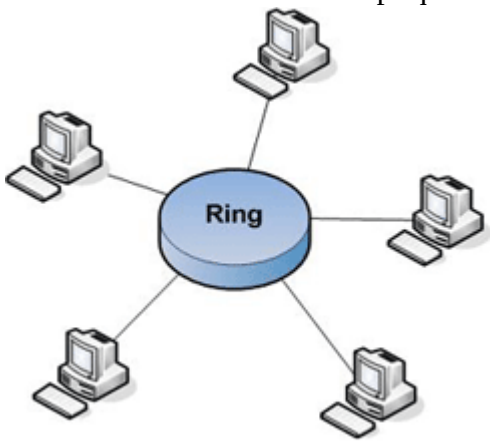


Fig.1.3. Topologie en anneau.

En réalité, dans une topologie anneau, les ordinateurs ne sont pas reliés en boucle, mais sont reliés à un répartiteur (appelé MAU, Multistation Access Unit) qui va gérer la communication entre les ordinateurs qui lui sont reliés en accordant à chacun d'entre-eux un temps de parole.

d) Topologie en arbre :

Aussi connu sous le nom de topologie hiérarchique, le réseau est divisé en niveaux. Le sommet, le haut niveau, est connectée à plusieurs nœuds de niveau inférieur, dans la hiérarchie. Ces nœuds peuvent être eux-mêmes connectés à plusieurs nœuds de niveau inférieur. Le tout dessine alors un arbre, ou une arborescence.

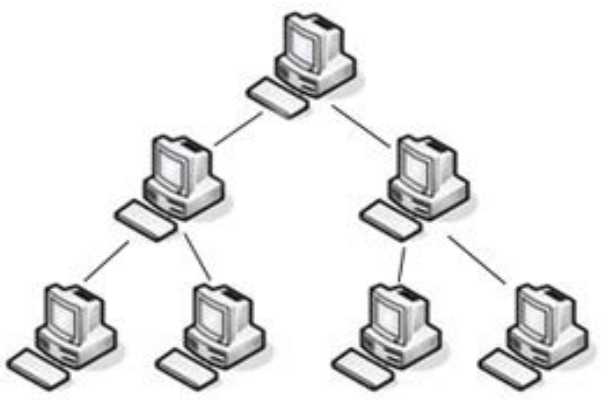


Fig.1.4. Topologie en arbre.

e) Topologie maillée :

Dans une topologie maillée, chaque ordinateur est connecté à chacun des autres ordinateurs par un câble séparé. Cette configuration fournit des itinéraires de routage redondants sur le réseau pour qu'en cas de défaillance d'un câble, un autre prenne le trafic en charge et que le réseau continue à fonctionner. À grande échelle, plusieurs réseaux locaux peuvent être interconnectés dans une topologie maillée, en utilisant des lignes téléphoniques dédiées, un câble coaxial ThickNet ou à fibres optiques.

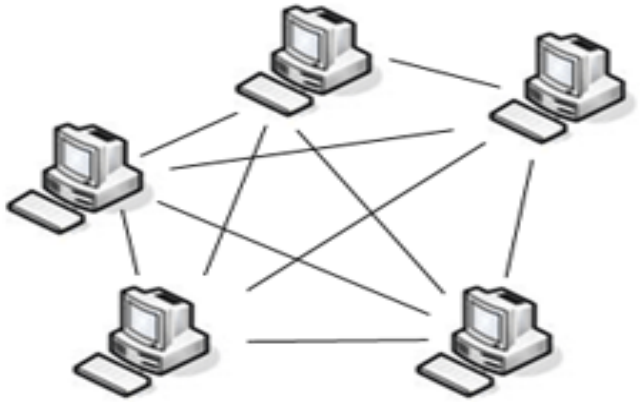


Fig.1.5. Topologie maillée.

Le principal avantage de la topologie maillée est sa capacité de tolérance de panne grâce à la redondance des itinéraires de routage sur le réseau. Comme cette redondance nécessite plus de câbles que les autres topologies, la topologie maillée peut s'avérer coûteuse.

f) Topologie Hybride :

La structure hybride de réseau emploie un mélange de différents genres de structures de réseau.

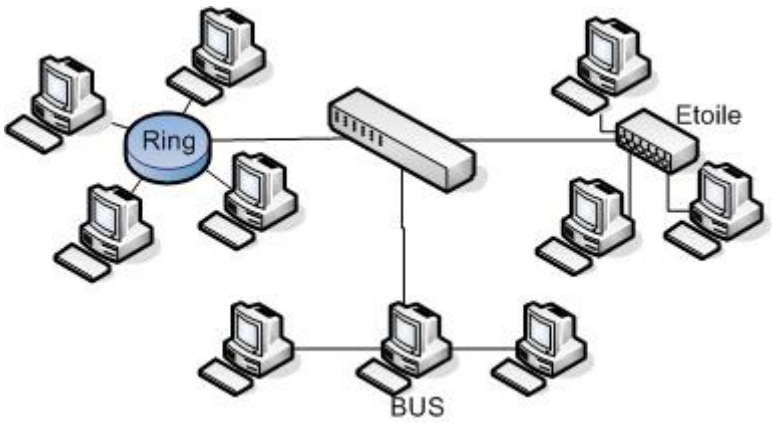


Fig.1.6. Topologie hybride.

1.7.2. Topologies logiques

La topologie logique décrit le mode de fonctionnement du réseau, la répartition des nœuds et le type de relation qu'ont les équipements entre eux. Les topologies logiques les plus courantes sont Ethernet, Token Ring et FDDI.

a) Ethernet :

Sur un réseau Ethernet, chaque station est identifiée par une adresse sur 6 octets. Ces adresses sont notées en hexadécimal (« 08:80:D3:A0:18:43 »). Cette adresse est statique et inscrite "en dur" sur la carte interface réseau. Les trames (de longueur variable comprises entre 72 et 1526 octets) sont diffusées sur l'ensemble du câble et sont "absorbées" par les extrémités munis de résistances.

La méthode d'accès utilisé par Ethernet est appelée CSMA/CD (Carrier Sense Multiple Acces with Collision Detection), Accès Multiple avec Écoute de Porteuse avec Détection de Collision.

La composition d'une trame Ethernet a le format suivant :

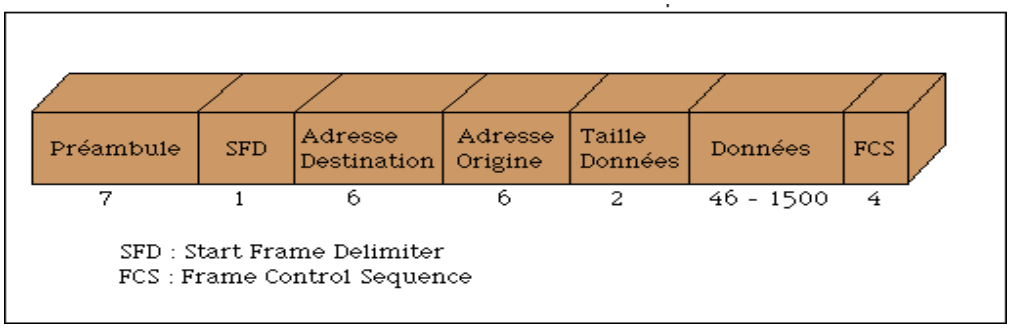


Fig. 1.7.1. Trame Ethernet

La vitesse de circulation des trames sur le câble est de 10 Mbits/s donc 10 bits/µs.

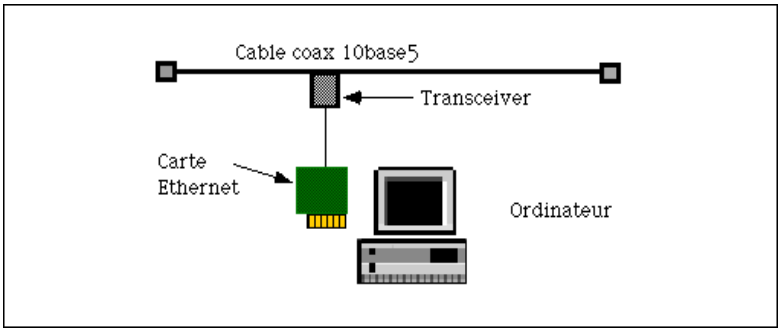
❖ **Méthode d'accès**

La méthode d'accès au réseau utilisée avec Ethernet est appelée CSMA/CD.
CSMA/CD est un ensemble de règles qui déterminent la façon dont les périphériques du réseau répondent lorsque deux de ces périphériques tentent de transmettre simultanément des données sur le réseau. La transmission simultanée de données par plusieurs ordinateurs provoque une collision. Tous les ordinateurs du réseau, clients et serveurs, vérifient le câble sur lequel s'effectue le trafic réseau. Un ordinateur ne transmet des données que lorsqu'il détecte que le câble est libre et exempt de trafic. Une fois que l'ordinateur a transmis des données sur le câble, aucun autre ordinateur ne peut transmettre des données tant que les données d'origine n'ont pas atteint leur destination, libérant ainsi le câble.

Lorsqu'il détecte une collision, un périphérique attend pendant un délai aléatoire, puis tente de retransmettre le message. S'il détecte de nouveau une collision, il attendra deux fois plus longtemps avant de retransmettre le message.

On distingue 3 architectures de câblage en Ethernet :

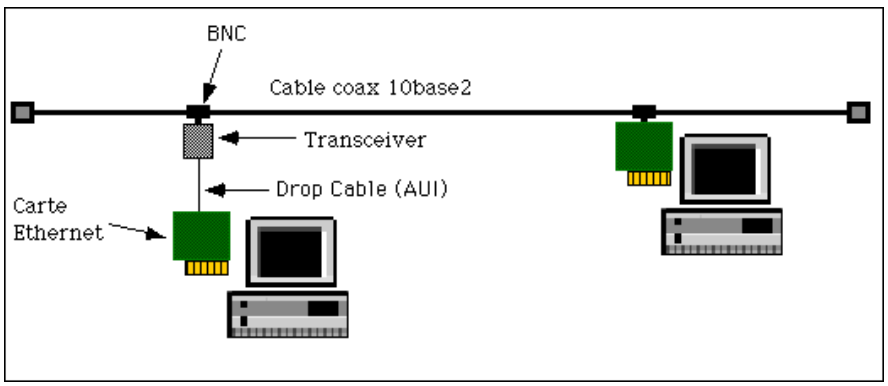
i) Le bus en gros câble coaxial (10base5, câble jaune, Thick Ethernet).



- ✓ La longueur maximum d'un segment doit être inférieure à 500 m
- ✓ Le nombre maximum de transceiver est 100, donc pas plus de 100 stations pour un segment !!
- ✓ 2 transceiver doivent être séparés d'au moins 2,5 m sur le câble.
- ✓ Avec des boîtiers spéciaux appelés *répéteurs* on peut mettre chaîner 3 segments de 500 m avec des câble de liaison qui peuvent faire eux-mêmes 500 m
- ✓ La distance maximale entre 2 stations ne peut dépasser 2500 m

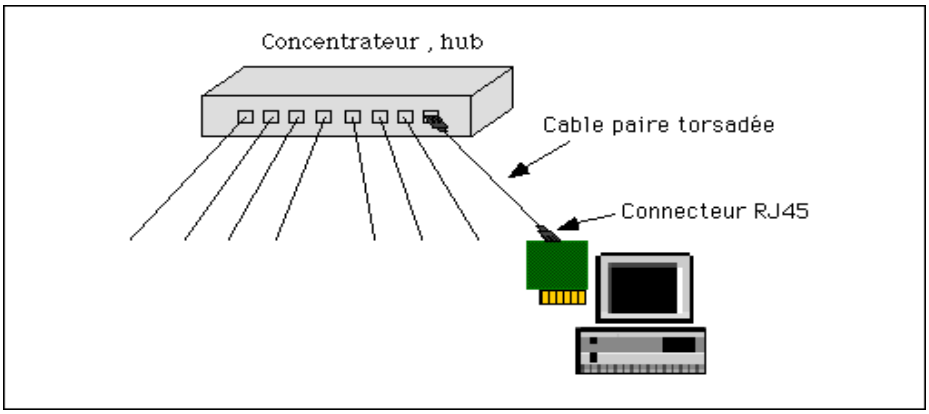
ii) Le bus en câble fin coaxial (10base2, Thin Ethernet, CheaperNet).

On peut utiliser la même technique de câblage que pour 10Base5 ou raccorder directement la carte interface sur le câble coaxial avec un connecteur BNC.



- ✓ Longueur max. d'un segment 185 m.
- ✓ Nombre de stations max. = 30
- ✓ On peut également raccorder plusieurs segments par l'intermédiaire de *répéteurs*

iii) L'étoile en paire torsadée. (10baseT)



- ✓ La longueur de chaque segment ne peut excéder 100 m, avec une seule station en terminaison de segment.
- ✓ Une terminaison peut aboutir à une autre étoile, on peut cascader ainsi plusieurs étoiles. Le nombre dépend du type de matériel.
- ✓ Une étoile est un boîtier qu'on trouve sous divers appellation comme, ***Hub, concentrateur, multi-répéteur...***

On peut, bien sûr, mixer les différentes techniques de câblages, ce qui peut donner un nombre varié de combinaisons d'architectures de câblage.

b) TOKEN-RING (Anneau à jeton)

Le jeton est une structure binaire ("trame") représentant le droit ou l'autorisation à transmettre sur le réseau pour le nœud qui l'acquiert. Le jeton circule sur une trame vide en passant d'un nœud à l'autre. Avec cette méthode, il ne peut y avoir, sur le réseau, qu'un seul nœud en transmission, les problèmes de collision sont ainsi éliminés. Deux états distincts caractérisent le jeton, l'état libre et l'état occupé. Ces deux états sont représentés par un bit dans la structure du jeton.

Lorsqu'un nœud prêt à émettre est traversé par un jeton libre, il commence par changer l'état du jeton en le rendant occupé, puis insère son message dans la trame contenant le jeton. La trame parcourt l'anneau jusqu'au destinataire qui le lit et le réémet. Lorsque l'émetteur revoit son message il aura la confirmation de la bonne réception et il change l'état du jeton en libre.

- ✓ Le débit de Token-Ring est de 16 Mbits/s

L'architecture de câblage est, en général l'étoile, avec un concentrateur qui est chargé de surveiller en permanence la présence et l'absence d'une des stations, et en cas d'absence de fermer automatique le circuit pour permettre aux transmissions de continuer. Ce concentrateur est souvent désigné dans la terminologie Token-Ring par **MAU** pour Médium Access Unit. (Le MAU est en fait un transceiver)

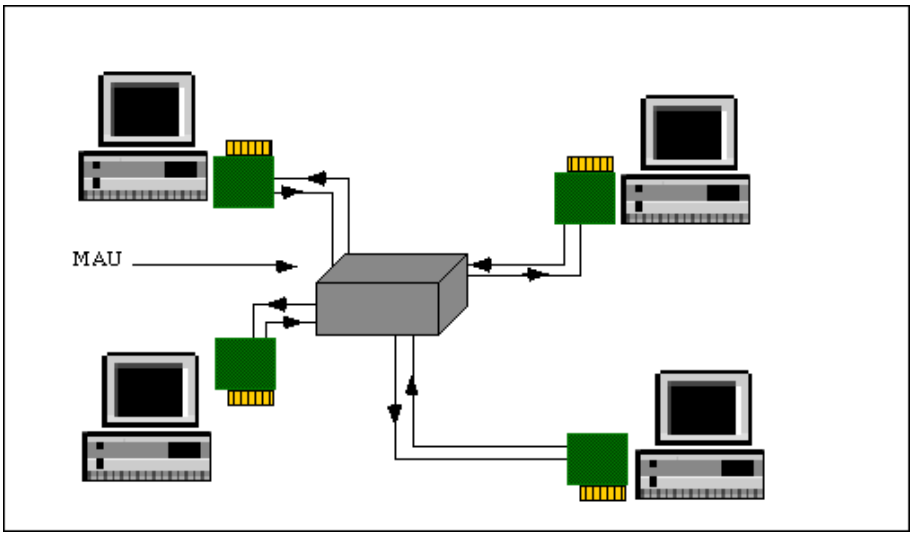


Fig. 1.7.2. Token Ring

Cette méthode d'accès à été introduit par IBM, elle est utilisée pour des réseaux de PC dans beaucoup d'entreprises commerciales.

c) FDDI

La technologie LAN FDDI (Fiber Distributed Data Interface) est une technologie d'accès au réseau sur des lignes de type fibre optique. Il s'agit en fait d'une paire d'anneaux (l'un est dit primaire, l'autre, permettant de rattraper les erreurs du premier, est dit secondaire).

Le principe est le même que Token-Ring, sauf qu'il utilise un support fibre optique et qu'il peut atteindre un débit 100 Mbits/s. Il est structuré en double anneau avec des jetons circulant en sens inverse, ce qui permet une meilleure sécurité en cas de coupure de l'anneau.

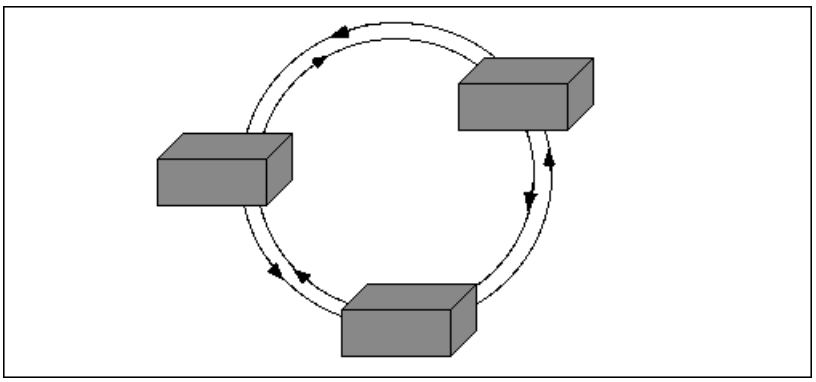


Fig. 1.7.3. FDDI

2. Les architectures

Pour comprendre comment les protocoles TCP/IP permettent d’assurer les communications réseau, il faut comprendre les différents concepts de ces

communications réseau. Le modèle OSI est un modèle conceptuel qui fait souvent office de référence pour comprendre les communications réseau.

2.1. L'architecture « OSI Open System Interconnexion »:

Le modèle de référence OSI est une représentation abstraite en couches servant de guide à la conception des protocoles réseau. Il divise le processus de réseau en sept couches logiques, chacune comportant des fonctionnalités uniques et se voyant attribuer des services et des protocoles spécifiques.

Dans ce modèle, les informations sont transmises d'une couche à l'autre, en commençant au niveau de la couche application sur l'hôte émetteur, puis en descendant dans la hiérarchie jusqu'à la couche physique, pour ensuite transiter sur le canal de communication vers l'hôte de destination, où les informations remontent la hiérarchie jusqu'à la couche application.

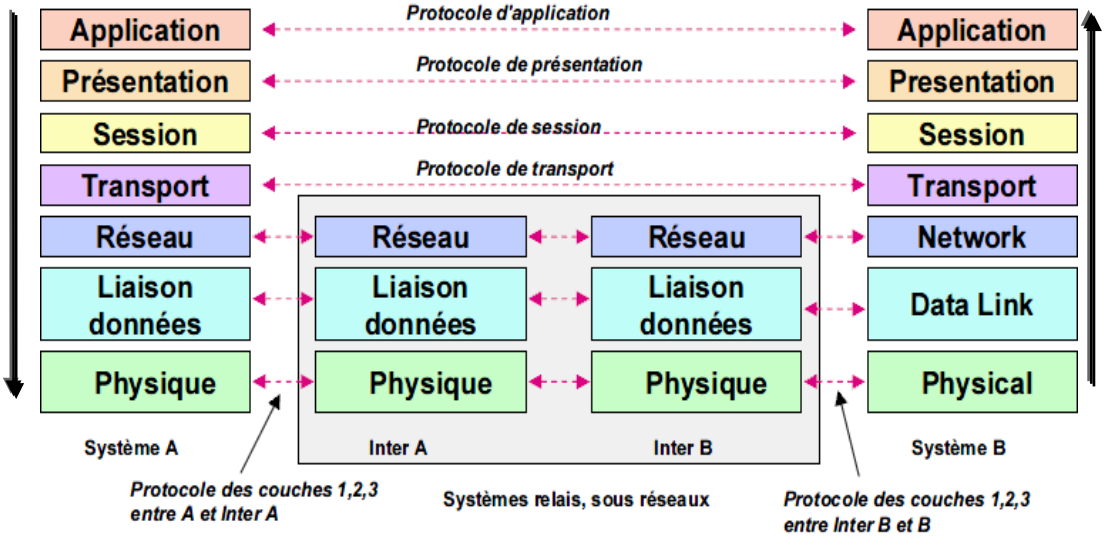


Fig. 1.8. Architecture OSI

Les couches sont généralement regroupées en couches basses, généralement les couches 1, 2 et 3 qui sont les couches proches du matériel et en couches hautes de la couche 4 à la couche 7 qui sont le plus proches des logiciels.

2.1.1. Couche Application

Couche sept. Fournit à des programmes, tels que les navigateurs Web et les systèmes de messagerie, un moyen d'accès aux services réseau.

2.1.2. Couche présentation

Couche six. La couche présentation s'occupe de la représentation des données circulant entre les différents systèmes d'un réseau. Elle transforme la syntaxe interne des données générées par la couche application en une syntaxe de transfert adaptée à la transmission des données via un réseau. Lorsque les données arrivent sur l'ordinateur

destinataire, la couche de présentation de cet ordinateur va décoder la syntaxe de transfert et la transformer en syntaxe locale.

2.1.3. Couche session

Couche cinq. La couche session permet à deux applications de créer une connexion permanente.

2.1.4. Couche transport

Couche quatre. La couche transport veille à ce que les paquets soient livrés dans l'ordre dans lequel ils ont été envoyés, sans perte ni duplication de données.

Dans le contexte du modèle de référence OSI, un *paquet* consiste en une enveloppe électronique contenant des informations constituées entre la couche session et la couche physique du modèle OSI.

2.1.5. Couche réseau

Couche trois. La couche réseau détermine le chemin d'accès physique des données à transmettre, en fonction des conditions de fonctionnement du réseau, de la priorité du service ou autre.

2.1.6. Couche liaison de données

Couche deux. La couche liaison de données est chargée d'assurer un transfert de trames de données exempt d'erreurs entre deux ordinateurs, via la couche physique.

Dans le contexte du modèle OSI, une *trame* est une enveloppe électronique contenant des informations incluant les paquets et les autres informations ajoutées par les sept couches du modèle.

Les couches situées au-dessus de la couche liaison de données peuvent assurer une transmission quasiment exempte d'erreurs sur le réseau.

2.1.7. Couche physique

Couche un. La couche physique établit les mécanismes et l'interface physique permettant de transmettre de façon brute un flux de bits de données sur le câble.

Les protocoles des couches du modèle OSI utilisent des noms différents pour désigner les unités de données qu'ils créent. Au niveau de la couche liaison de données, c'est le terme *trame* qui est utilisé. Au niveau de la couche réseau, on parle de *datagramme*. Le terme plus générique de *paquet* est utilisé pour décrire l'unité de données créée au niveau de n'importe quelle couche du modèle OSI.

2.2.L'architecture TCP/IP :

La suite TCP/IP est constituée d'une suite de protocoles normalisés assurant la communication au sein d'un environnement hétérogène. Les tâches impliquées dans l'utilisation du modèle TCP/IP, dans le cadre d'un processus de communication, sont réparties entre des protocoles organisés en quatre couches distinctes constituant la pile de protocoles TCP/IP. Ces couches sont les suivantes :

- ✓ Couche application
- ✓ Couche transport
- ✓ Couche internet
- ✓ Couche accès réseau

2.2.1. Avantages du protocole TCP/IP

Le fait de répartir les différentes fonctions du réseau entre plusieurs protocoles indépendants, plutôt que de créer un protocole unique, fournit un certain nombre d'avantages :

- ✓ Les protocoles indépendants facilitent la prise en charge de diverses plates-formes. La création ou la modification de protocoles pour prendre en charge de nouvelles normes ne requiert pas la modification de l'ensemble de la pile de protocoles.
- ✓ Le fait d'avoir plusieurs protocoles intervenant sur la même couche permet aux applications de sélectionner uniquement les protocoles qui fournissent le niveau de service requis.
- ✓ Étant donné que la pile est divisée en couches, le développement des protocoles peut être effectué simultanément, par différentes personnes spécialisées dans le traitement d'opérations sur une couche précise.

2.3. Correspondance entre le modèle TCP/IP et le modèle OSI

Le modèle OSI définit des couches distinctes liées à la mise en paquet, l'envoi et la réception de données sur un réseau. La suite de protocoles constituant la pile TCP/IP prend en charge ces différentes fonctions.

a) Protocole : Description des mécanismes permettant la gestion des paquets d'information et leur transition du réseau à l'application. Par extension, logiciel (software) fonctionnant sur une machine et permettant cette gestion interne.

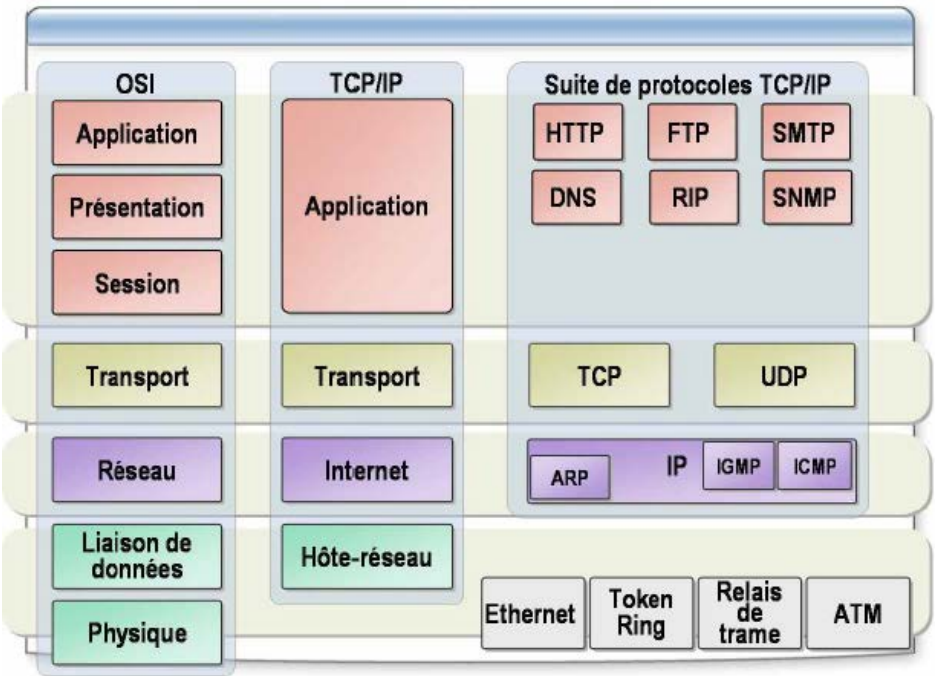


Fig. 1.9. Architecture TCP/IP

2.3.1. Couche Application

La couche application du modèle TCP/IP correspond aux couches application, présentation et session du modèle OSI. Elle fournit des services et des outils permettant aux applications d'accéder aux ressources du réseau. Les deux services de cette couche sont les suivants : Windows Sockets et Network Basic Input/Output Systems (NetBIOS). Ils fournissent tous les deux des interfaces d'application standard permettant aux programmes d'accéder aux services réseau.

b) Protocoles de la couche application

Certains des protocoles qui interviennent au niveau de cette couche se connectent à d'autres hôtes réseau ou communiquent avec eux. Ils sont décrits dans le tableau ci-dessous :

Protocole	**Description**
HTTP	Hypertext Transfer Protocol. Ce protocole spécifie les processus d'interaction client/serveur entre les navigateurs Web et les serveurs Web.
FTP	File Transfer Protocol. Ce protocole effectue des transferts de fichier et exécute les tâches de gestion de base sur les ordinateurs distants.
SMTP	Simple Mail Transport Protocol. Ce protocole assure le transport de courriers électroniques d'un serveur à un autre ou d'un client à un serveur.
DNS	Domain Naming System. Ce protocole assure la résolution de noms d'hôte Internet en adresses IP pour les communications réseau.
RIP	RIP Protocole de routage d'information (Routing Information Protocol). Ce protocole permet aux routeurs de recevoir des informations sur les autres routeurs du réseau.
SNMP	Protocole simplifié de gestion de réseau (Simple Network Management Protocol). Ce protocole nous permet de recueillir des informations sur des périphériques réseau, tels que les concentrateurs, les routeurs et les ponts. Les informations collectées sur un périphérique sont définies dans une base d'informations de gestion (MIB, Management Information Base).

Tableau 1.1 Description des protocoles de la couche Application

2.3.2. Couche Transport

La couche transport du modèle TCP/IP correspond à la couche transport du modèle OSI. Elle garantit que les données sont bien acheminées vers la destination voulue et assure la communication entre les deux extrémités par le biais de l'un des deux protocoles suivants :

Protocoles	**Description**
TCP	Protocole de contrôle de transmission (Transmission Control Protocol). Ce protocole, orienté connexion, garantit la fiabilité des communications. Il est utilisé par des applications qui assurent le transfert d'un grand nombre de données en une fois ou qui nécessitent la garantie de la réception des données.
UDP	User Datagram Protocol. Ce protocole assure des communications sans connexion et ne garantit pas que les paquets atteignent leur destination. C'est la couche application qui se charge de la fiabilité de la transmission. Les applications utilisent le protocole UDP pour accroître la vitesse de communication grâce à un temps système réduit par rapport à l'utilisation de

	TCP. Le protocole SNMP utilise le protocole UDP pour l'envoi et la réception de messages sur le réseau. En général, les applications transfèrent une petite quantité de données à la fois par le biais d'UDP.

Tableau 1.2 Description des protocoles de la couche Transport

2.3.3. Couche Internet

La couche Internet correspond à la couche réseau du modèle OSI. Les protocoles de cette couche assurent l'encapsulation des données de la couche transport en unités appelées paquets ainsi que l'adressage et le routage de ces paquets vers leur destination.

Il existe quatre protocoles au niveau de cette couche :

Protocoles	Description
IP	Internet protocol. Ce protocole est chargé de l'adressage et du routage des paquets entre les hôtes et les réseaux.
ARP	Protocole de résolution d'adresse (Address Resolution Protocol). Ce protocole permet d'obtenir les adresses matérielles des hôtes situés sur le même réseau physique.
IGMP	Protocole de gestion de groupes Internet (Internet Group Management protocol). Ce protocole gère les appartenances des hôtes aux groupes de multidiffusion IP.
ICMP	Internet Control Message Protocol. Ce protocole envoie des messages et signale les erreurs concernant l'acheminement des paquets.

Tableau 1.3 Description des protocoles de la couche Internet

2.3.4. Couche Accès réseau

La couche accès-réseau correspond aux couches liaison et physique du modèle OSI. Cette couche spécifie les exigences relatives à l'envoi et à la réception de paquets. Elle est chargée d'envoyer ou de recevoir les données du réseau physique.

3. Internet, extranet et intranet

3.1. Internet

L'***internet*** est un inter réseau, c'est-à-dire un ensemble de réseaux, mais il s'agit d'un système très particulier :

- ✓ il se compose de systèmes informatiques totalement indépendants les uns des autres ;
- ✓ n'importe qui peut accéder à ces systèmes ;
- ✓ mais il n'y a que deux canaux d'accès: les protocoles HTTP (web) et SMTP (e-mail).

3.2.Intranet

Un ***intranet*** est un internet privé, c'est-à-dire un internet dont l'accès est réservé aux employés de l'entreprise concernée.

Les notions de réseau local et d'intranet n'ont rien à voir l'une avec l'autre :

- ✓ si les ordinateurs d'un site sont reliés par des câbles ou des ondes et fonctionnent ensemble, c'est un réseau local ;
- ✓ si les utilisateurs des ordinateurs d'une organisation ont accès à un site web réservé aux employés de l'entreprise, c'est un intranet.

L'intranet d'une multinationale couvre le monde entier.

3.3.Extranet

Un réseau extranet est un réseau du type Internet (donc essentiellement basé sur le protocole IP), c'est une extension du système d'information de l'entreprise à des partenaires situés au-delà du réseau.

L'accès à l'extranet se fait via Internet, par une connexion sécurisée avec mot de passe dans la mesure où cela offre un accès au système d'information à des personnes situées en dehors de l'entreprise. L'extranet est donc en général un site à accès sécurisé qui permet à l'entreprise de n'autoriser la consultation d'informations confidentielles qu'à certains intervenants externes comme à ses fournisseurs, ses clients, aux cadres situés à l'extérieur de l'entreprise, aux commerciaux, etc.

L'extranet est un système supplémentaire offrant par exemple aux clients d'une entreprise un accès privilégié à certaines ressources informatiques de l'entreprise.

4. Modèle client/serveur

Dans le modèle client/serveur, le périphérique demandant les informations est nommé client et celui répondant à la demande est nommé serveur. Les processus client et serveur sont considérés comme faisant partie de la couche application. Le client commence l'échange en demandant des données au serveur, qui répond en envoyant un ou plusieurs flux de données au client. Les protocoles de couche application décrivent le format des requêtes et des réponses entre clients et serveurs. Outre le transfert de données effectif, cet échange peut également nécessiter des informations de contrôle, telles que l'authentification de l'utilisateur et l'identification d'un fichier de données à transférer.

Le client

- ➢ effectue une demande de service auprès du serveur (***requête***)
- ➢ initie le contact (parle en premier), ouvre la session

Le serveur

- ➢ est la partie de l'application qui offre un service
- ➢ est à l'écoute des requêtes clientes
- ➢ répond au service demandé par le client (***réponse***)

Le client et le serveur ne sont pas identiques, ils forment un *système coopératif*

- ➢ les parties client et serveur de l'application peuvent s'exécuter sur des systèmes différents
- ➢ une même machine peut implanter les côtés client ET serveur de l'application
- ➢ un serveur peut répondre à plusieurs clients simultanément

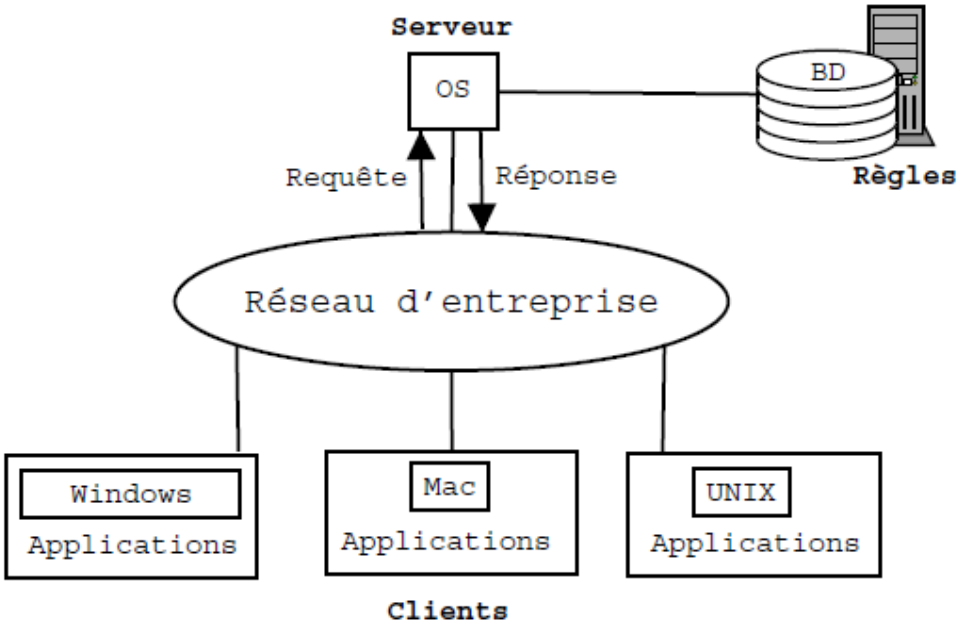


Fig. 1.10. Modèle Client/Serveur

4.1.Le middleware

- Assure les connexions entre les serveurs de données et les outils de développement sur les postes client
- Ensemble de services logiciels construits au dessus d'un protocole de transport afin de permettre l'échange de requêtes et des réponses associées entre client et serveur *de manière transparente*.
- Les services du middleware sont un ensemble de logiciels répartis qui existe entre l'application, l'OS et les services réseaux sur un nœud du réseau.

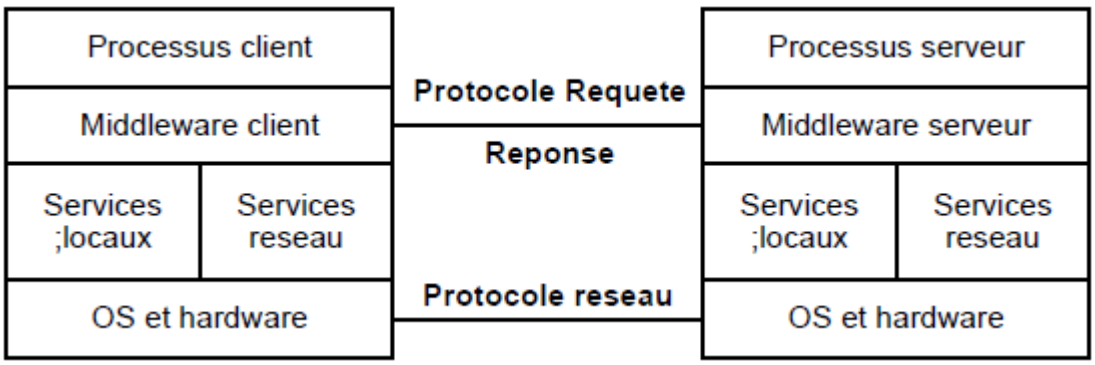


Fig. 1.11 Architecture Middleware

4.1.1. Fonctions d'un middleware

Le middleware assure plusieurs fonctions :

- ✓ procédure d'établissement/ fermeture de connexion
- ✓ exécution des requêtes/ récupération des résultats
- ✓ initiation des processus sur différents sites
- ✓ services de répertoire (nommage)
- ✓ accès aux données à distance
- ✓ gestion des accès concurrents
- ✓ sécurité et intégrité

- ✓ monitoring
- ✓ terminaison des processus
- ✓ mise en cache des résultats, des requêtes

5. Le modèle Peer-to-Peer

5.1.Définition et caractéristiques

Dans son essence, l'informatique pair-à-pair se définit comme le partage des ressources et des services par échange direct entre systèmes. Ces échanges peuvent porter sur les informations, les cycles de traitement, la mémoire cache ou encore le stockage sur disque des fichiers.

Contrairement au modèle client / serveur, chaque système est une entité réseau complète qui remplit à la fois le rôle de serveur et celui de client. Avec le peer-to-peer, les ordinateurs personnels ont le droit de faire partie du réseau.

Le peer-to-peer désigne donc une classe d'applications qui tirent partie des ressources matérielles ou humaines qui sont disponibles sur le réseau Internet.

Une caractéristique des réseaux peer-to-peer est que la qualité et la quantité des données disponibles augmentent à mesure que le nombre d'utilisateurs augmente. La valeur du réseau augmente donc avec sa popularité.

Enfin, la pertinence d'un système peer-to-peer réside dans sa capacité à localiser les ressources efficacement quelle que soit la taille du réseau. Ainsi, ces systèmes doivent reposer sur des méthodes efficaces de découverte des ressources désirées.

5.2.Réseaux Peer to Peer

Dans un réseau Peer to Peer, au minimum deux ordinateurs sont connectés via un réseau et peuvent partager des ressources (par exemple, des imprimantes et des fichiers) sans disposer de serveur dédié. Chaque périphérique final connecté (nommé homologue) peut opérer en tant que serveur ou en tant que client. Un ordinateur peut remplir le rôle de serveur pour une transaction tout en servant simultanément de client pour un autre ordinateur. Les rôles de client et de serveur sont définis en fonction de chaque requête. Contrairement au modèle client/serveur, qui utilise des serveurs dédiés, les réseaux Peer to Peer décentralisent les ressources sur un réseau.

5.3.Applications Peer to Peer

Une application Peer to Peer (P2P), contrairement à un réseau Peer to Peer, permet à un périphérique d'opérer à la fois comme client et comme serveur au sein d'une même communication. Dans ce modèle, chaque client est un serveur et chaque serveur un client. Les deux peuvent lancer une communication et sont considérés comme égaux dans le processus de communication. Cependant, les applications Peer to Peer nécessitent que chaque périphérique final fournisse une interface utilisateur et exécute un service en tâche de fond. Lorsqu'on lance une application Peer to Peer spécifique, elle invoque l'interface utilisateur et les services en tâche de fond requis. Les périphériques peuvent ensuite communiquer directement.

Certaines applications Peer to Peer utilisent un système hybride dans lequel le partage des ressources est décentralisé mais les index pointant vers l'emplacement des ressources sont stockés dans un répertoire centralisé. Dans un système hybride, chaque homologue accède à

un serveur d'index pour obtenir l'emplacement d'une ressource stockée chez un autre homologue. Le serveur d'index permet également de connecter deux homologues, mais une fois ceux-ci connectés, la communication s'effectue entre les deux homologues sans autre communication vers le serveur d'index.

Les applications Peer to Peer peuvent être utilisées sur des réseaux Peer to Peer, des réseaux client/serveur et via Internet.

6. Le concept de supervision réseau

La supervision réseau a pour but de surveiller le bon fonctionnement des réseaux.

Ce concept est né au début des années 1980, lors de l'explosion de la mise en place de réseaux informatiques dans les entreprises. La taille grandissante de ceux-ci ainsi que leur hétérogénéité posaient un réel problème de gestion et d'administration, multipliant les besoins en main d'œuvre d'experts administrateurs. C'est donc à cette époque qu'ont été menées les premières réflexions sur un nouveau concept, celui de la supervision.

La supervision devait être capable de s'adapter à des milieux hétérogènes, d'automatiser le contrôle des réseaux et de générer un ensemble de statistiques donnant une meilleure vision du réseau, permettant par là-même d'anticiper les besoins de celui-ci.

La supervision peut ainsi se définir comme étant l'utilisation de ressources réseaux adaptées (matérielles ou logicielles) afin d'obtenir des informations sur l'utilisation et sur l'état des réseaux et de leurs composants (logiciels, matériels).

Ces informations peuvent alors servir d'outils pour gérer de manière optimale (automatique si possible) le traitement des pannes ainsi que la qualité des réseaux (problèmes de surcharge). Elles permettent également de prévoir toute future évolution nécessaire.

Conclusion

Le réseau est devenu une ressource indispensable (voir vitale) au bon fonctionnement d'une organisation, une entreprise, etc.

Au cours de ce chapitre, on a fait un survol général des réseaux, c'est-a-dire, ses modes de connexion, les topologies, types d'architectures, les différentes protocoles nécessaires à la communication entre les différentes couches, etc. Des éléments essentielles qui nous aiderons à mieux comprendre les notions des réseaux.

Dans le chapitre suivant on introduit la MIB, une base de données contenant des informations relatives aux éléments à gérer.

Chapitre II

Management Information Base (MIB)

1. Présentation de la MIB

Comme n'importe quel système de gestion, les systèmes de gestion des réseaux, basés sur la suite de protocoles TCP/IP, possèdent une base de données contenant des informations relatives aux éléments à gérer. Avec les architectures TCP/IP et OSI, cette base de données est connue sous le nom de *Management Information Base* (MIB). Chaque ressource à gérer (par exemple une imprimante, un *Hub*, un routeur, un Switch ou un serveur e-mail) est représentée sous forme d'un objet. La notion d'objet dans ce contexte est différente de celle utilisée dans la programmation orientée objet. En effet, un objet dans la programmation orientée objet possède des attributs et des méthodes et peut supporter les notions d'héritage et de polymorphisme. Cependant, un objet au sens des MIB ne reconnaît aucune de ces notions. Vis-à-vis des spécifications ASN.1, la MIB est défini comme étant un fichier ASCII constitué d'un ensemble de définitions d'objets. Les règles sémantiques et les méthodes d'exploitation sont spécifiées par ces objets. Du côté des agents et des applications de gestion, une MIB est constituée par des instances d'objets, qui ne sont que des variables. Ces dernières correspondent aux abstractions des différentes ressources à gérer.

Chaque nœud du système maintient une MIB qui reflète l'état des ressources gérées qu'il possède. Une entité de gestion (application chez une station de gestion) peut contrôler les ressources d'un nœud en lisant les valeurs des objets de la MIB et les gère en modifiant et en mettant à jour ces mêmes valeurs.

Pour que la MIB réponde à tous les besoins d'un tel système de gestion des réseaux, il faut tenir compte de deux points:

a) avoir une structure d'objet commune pour présenter un certain type de ressource au niveau de tous les nœuds;

b) avoir un schéma commun de représentation des objets pour supporter l'interopérabilité.

Le premier point est solutionné en utilisant une définition commune à tous les objets. Le deuxième point est solutionné en définissant un unique et commun modèle d'information de gestion *(Structure of Management Information*, SMI). Tout objet défini au sein de la MIB doit l'être au moyen de SMI pour que les MIB soient interopérables.

Tout élément du réseau doté d'agent SNMP dispose d'une MIB qui est géré par l'agent SNMP contenant l'ensemble des informations d'administration qui lui concerne. La consultation de la MIB est indépendante du matériel et du logiciel utilisé, cette consultation est réalisée grâce au protocole SNMP qui permet de retrouver les informations à consulter, les paramètres à modifier, les alarmes à émettre, etc.

L'organisation et la mise en œuvre de la MIB ne font pas l'objet d'une normalisation. Les informations (données) qu'elle contient se présentent sous la forme de scalaires (valeurs uniques) ou des tableaux de scalaires et qui sont constitués de compteurs, de seuils, de répertoires, de noms et d'adresses, etc.

L’accès à la MIB peut se faire soit localement par l’intermédiaire des mécanismes non normalisés, soit à distance en utilisant des protocoles de gestion.

2. Structure de la MIB

On sait maintenant que la MIB est l'équivalent d'une base de données, constituée d'un ensemble d'objets. À chaque objet est associé un identificateur unique nommé OBJECT IDENTIFIER (OID) dont la valeur est constituée d'une séquence d'entiers non négatifs. L'ensemble des objets définis au sein d'un MIB est structuré à la façon d'un arbre. Les objets sont aux feuilles de l'arbre. À chaque nœud de l'arbre, sauf à la racine, est associé un entier. La concaténation des entiers sur un chemin de l'arbre, de la racine à une feuille, constitue l'identificateur de l'objet qui est à la feuille. Trois nœuds sont situés en dessous de la racine: *iso, ccitt* et *joint-iso-ccitt*. Par exemple, l'identificateur du nœud *internet* est "*internet OBJECT IDENTIFIER ::= {iso(1) org(3) dod(6) 1}",* cet identificateur a la valeur 1.3.6.1. Cette dernière sert comme préfixe pour tous les nœuds figurant en dessous du nœud internet. Généralement, un identificateur sert à identifier le nœud ainsi que tous les objets qui sont situés en dessous de lui.

La structure de la MIB est normalisée ainsi que les appellations des diverses rubriques qui rendent les informations plus lisibles, elle est organisée hiérarchiquement, de la même façon que l'arborescence des domaines Internet. Elle contient une partie commune à tous les agents SNMP et une partie spécifique aux agents SNMP d'un même type de matériel ou au constructeur. En réalité, chaque niveau de la hiérarchie est repéré par un index numérique et SNMP n'utilise que cette façon de faire. Dans la structure arborescente de la MIB, les branches peuvent se développer, celles qui n'aboutissent pas sont là pour indiquer qu'il y a le plus souvent des objets dessus, mais les branches qui semblent finies ne le sont pas forcément. Avec ce genre d'organisation, un nœud sera accessible lorsque l'on connaîtra son chemin complet, depuis la racine de l'arbre.

Notons que les appellations en texte ("member-body" par exemple), peuvent varier légèrement d'une implémentation à l'autre, elles ne sont là que pour rendre les choses plus compréhensibles. Ce qui ne varie jamais, en revanche, c'est l'index qui y correspond, placé entre parenthèses sur le dessin. Pour un réseau IP, c'est clairement le nœud "dod (6)" qui sera le plus utilisé et particulièrement le sous nœud "Internet (1)".

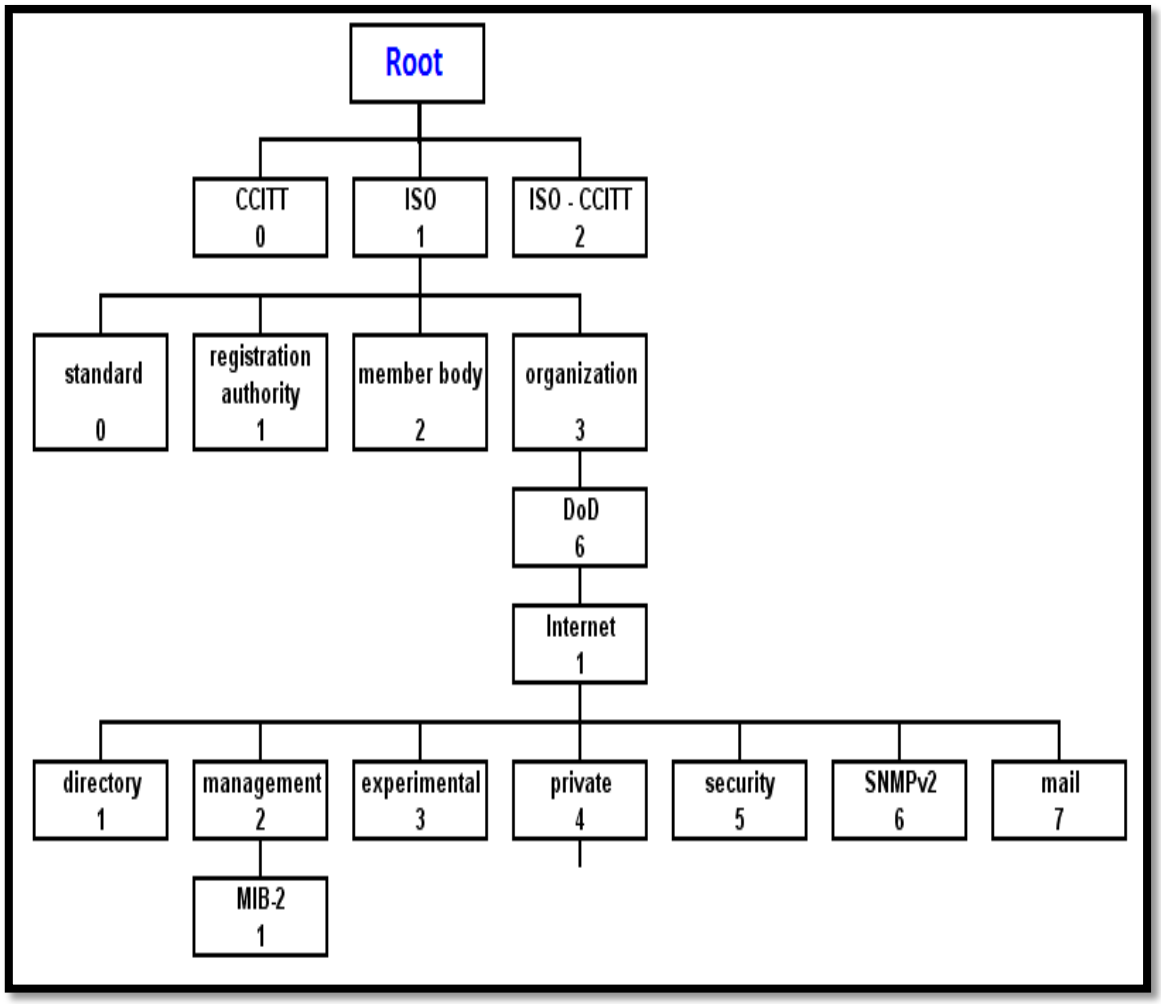


Figure 2. 1 : Structure arborescente de la MIB

Comme cet arbre va dépendre de l'agent, il sera pratique de disposer d'un outil permettant de le parcourir pour en découvrir la structure. Cet outil se trouvera dans le manager SNMP.

3. 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 1).

La classification des objets est arborescente. L'identificateur d'un objet est défini, en ASN.1 par le chemin qui conduit à l'objet.

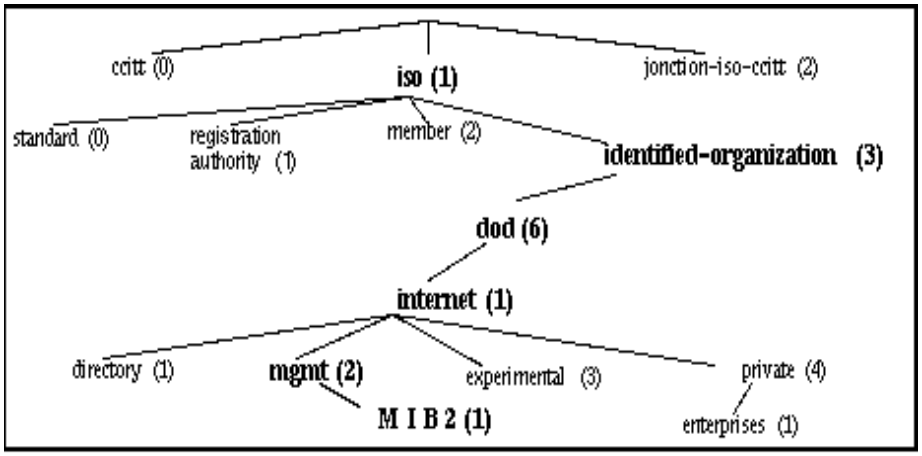


Figure 2.2 : Exemple de représentation de données dans la MIB

Par exemple, pour accéder à un objet management (mgmt), son identificateur autrement appelé **OID** commencera par **1.3.6.1.2** (**iso.org.dod.internet.mgmt**).

4. Les différentes MIBs

4.1.La MIB-1 (MIB standard)

La MIB standard qui est apparue en Mai 1990 a connu une évolution rapide pendant les dernières années. Afin de simplifier l'implantation et l'attribution des OIDs, la MIB est constituée de huit groupes (systèmes, interfaces, adress translation, IP, ICMP, TCP, UDP, et EGP) comme illustré dans la figure suivante.

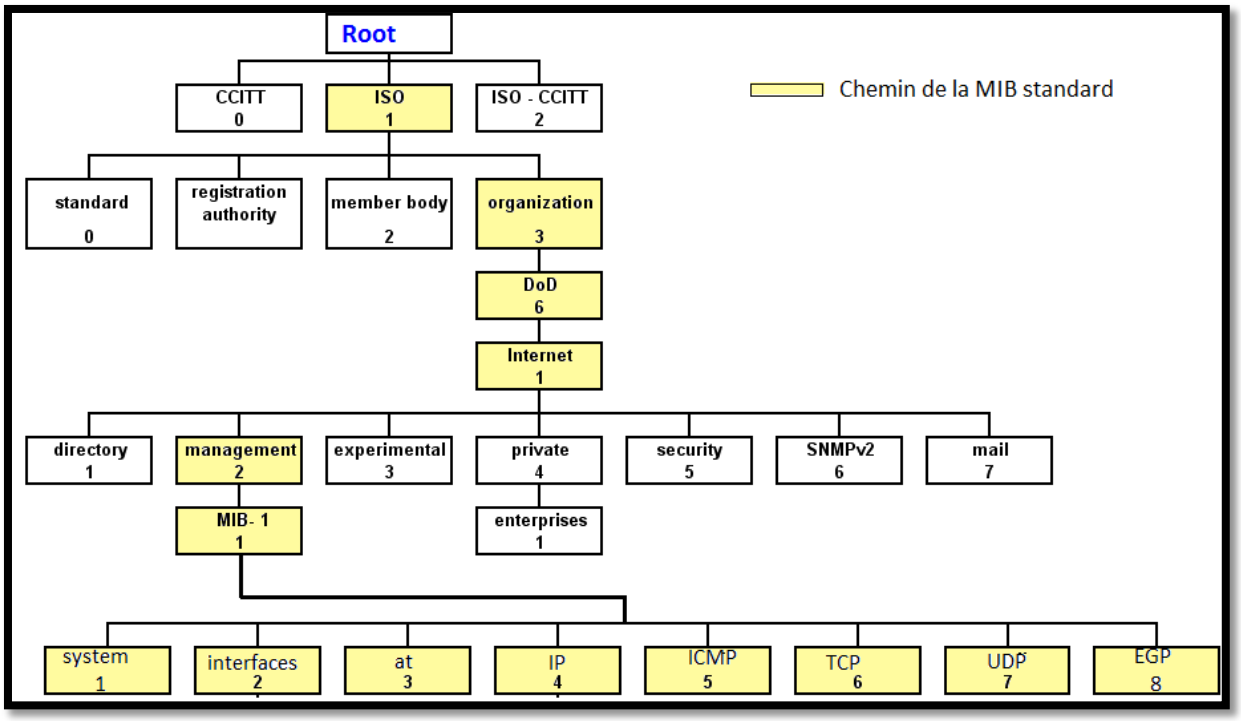


Figure 2.3: Structure arborescente standard (MIB-1)

Pour qu'un client accède à ces objets, il faut qu'il soit au courant de leur existence. Une MIB contient un certain nombre d'informations standards : c'est la MIB standard. Or pour la plupart des éléments réseau, on rajoute un certain nombre d'objets propres à un agent pour en exploiter les possibilités : c'est la MIB privée**.** Par exemple, dans la MIB standard il y a des compteurs qui gèrent les paquets émis ou reçus sur chaque interface de l'appareil. Parce que n'importe quel client est capable de lire ces compteurs, des constructeurs différents sont capables de retrouver ces informations.

On peut donc dire que la MIB standard désigne le plus petit dénominateur commun entre tous les types de matériel que l'on peut rencontrer sur un réseau.

Le premier standard utilisé pour la définition des objets d'administration de la MIB standard fut la MIB-I, son OID 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	Egp	6

Tableau 2.1: Objets de la MIB I

4.2.MIB II

La version 2 de la MIB est apparue en 1991, c'est une amélioration de la MIB standard (RFC 1212 et 1213). Ce standard est appelé MIB-II et contient 172 éléments, elle a remplacé actuellement la MIB I. Son OID est aussi : 1.3.6.1.2.1

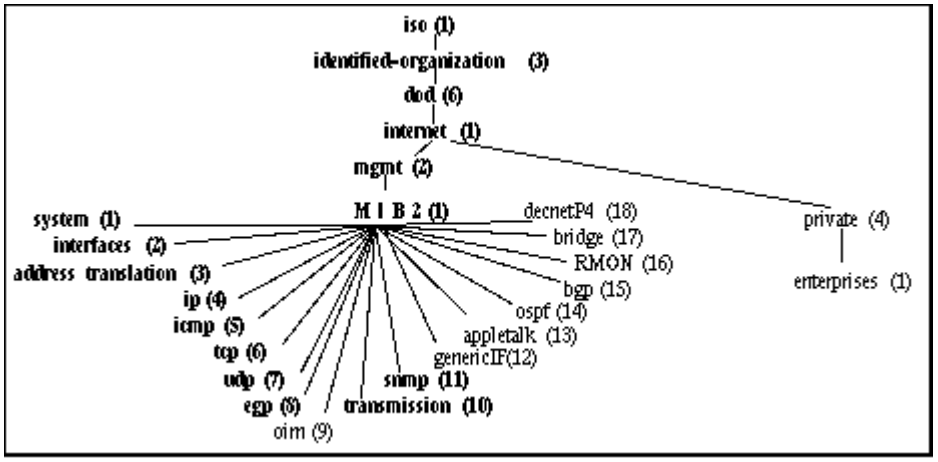


Figure 2. 4 : MIB II

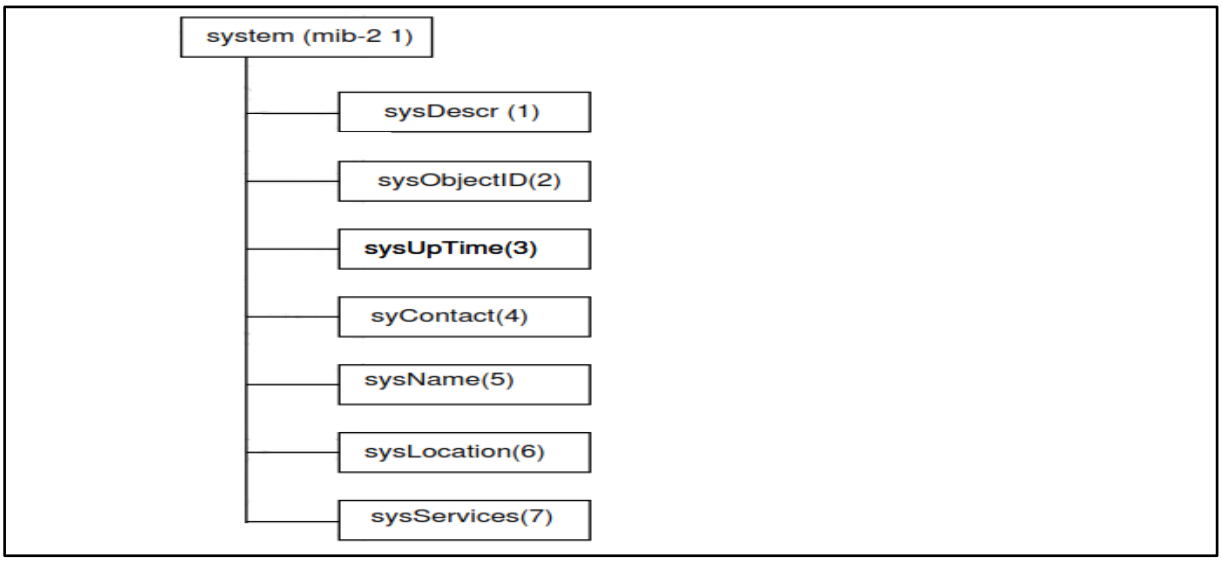
Sa définition est la suivante:

Numéro	Objet	Nombre de sous-objets	Description
1	*System*	7	Informations générales concernant l'agent 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 ***at***
11	*SNMP*	30	Statistiques du trafic SNMP

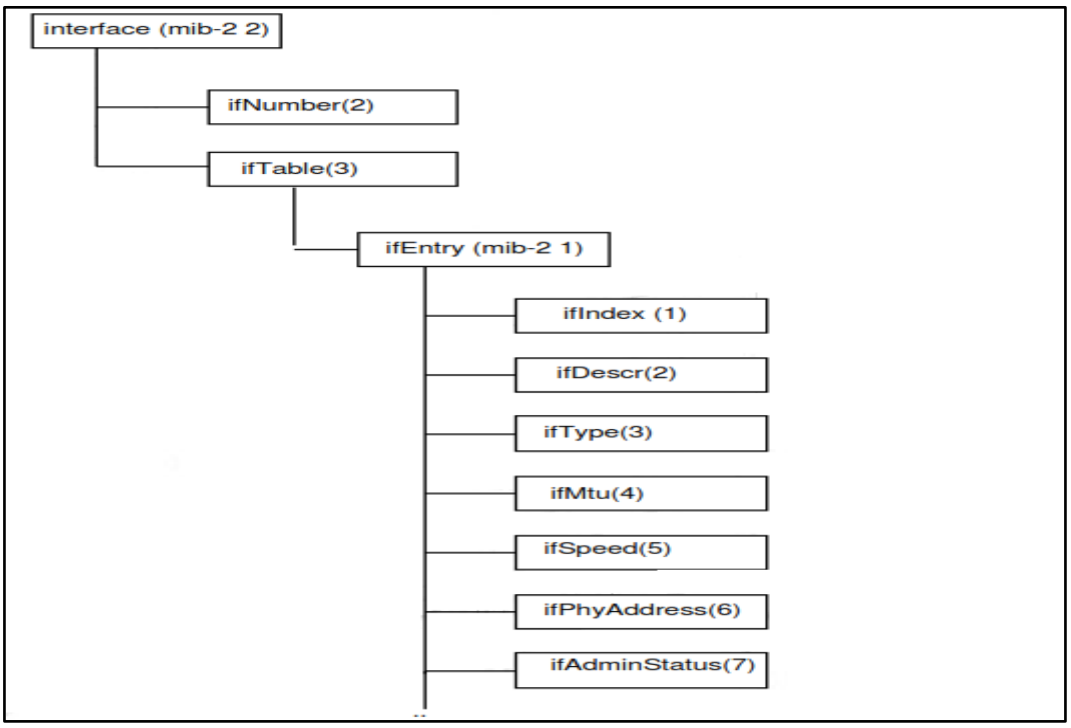
Tableau 2. 2 : description de quelques objets de la MIB II

4.2.1. Description de quelques objets de la MIB II

System : correspond au nom de l'agent, no de version, type de la machine, nom du système d'exploitation, type de logiciel réseau en ASCII imprimable.

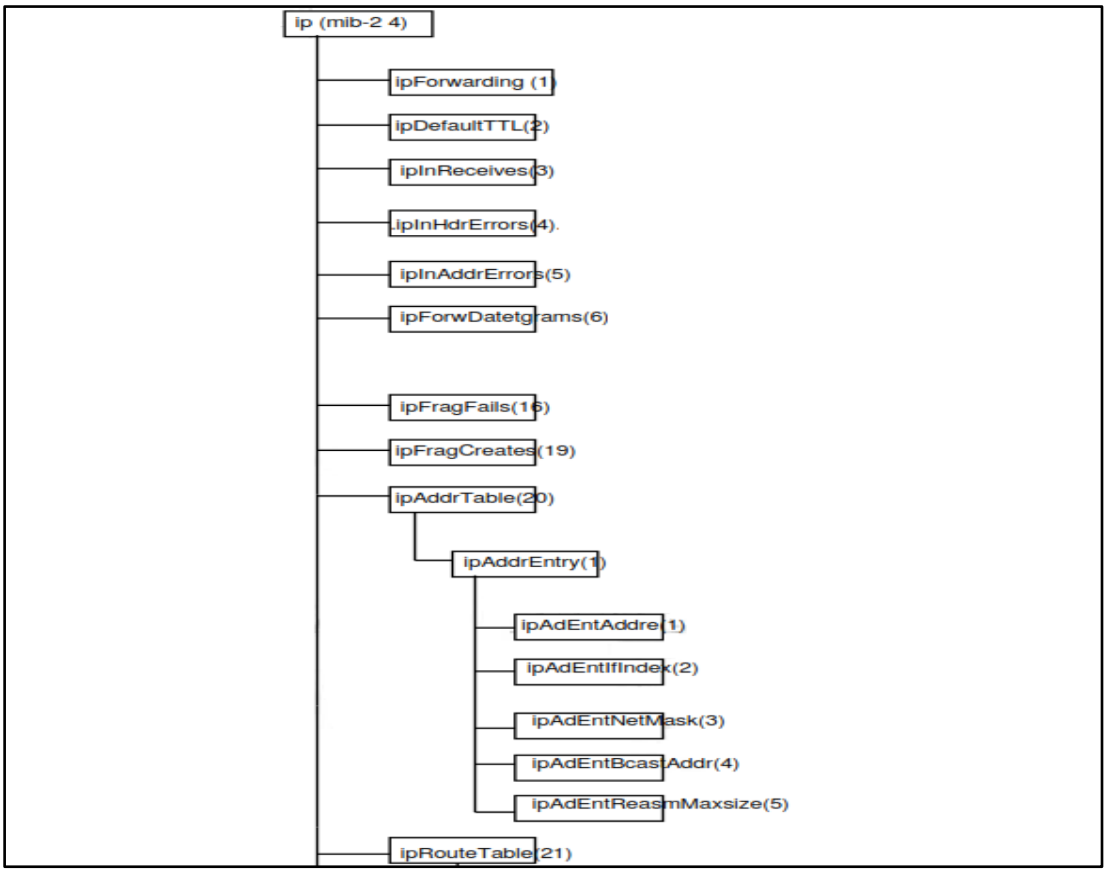


Interface : les différentes interfaces réseau d'une machine (nombre d'interface, type des interfaces et nom du fabricant, vitesse des interfaces, nombre de paquets entrants, sortants, en erreur, etc.



At : conservé pour des raisons de compatibilité avec MIB-I. gère une table de translation entre des adresses réseau de niveau logique (IP) et adresses spécifiques (Ethernet). équivalent à la table ARP.

IP : très souvent considéré comme la partie la plus important de la MIB, il gère plusieurs paramètres comme la durée de vie par défaut des paquets IP, le nb de paquets reçus ou envoyés, le nb de paquets réassemblés avec succès ainsi que le nb de fragments crées, la table de routage si elle existe, le masque sous-réseau, l'adresse physique, etc.



ICMP :Il comprend 26 compteurs:

- pour chaque message ICMP, 2 compteurs pour compter les messages reçus et émis

- 4 compteurs pour compter le nombre total de messages ICMP reçus, reçus par erreur ou non envoyés.

TCP : rend compte des connexions TCP en cours et des paramètres de type nombre max de connexions simultanées permises, nombre d'ouverture active et l'état de chaque connexion (écoute, time-wait).

UDP :il comprend 4 compteurs renseignant sur le nombre de datagramme UDP envoyés, reçus, ou en erreur

EGP : gère le protocole EGP (External gateway protocol)(routage des paquets entre routeurs). Cette partie de la MIB donne des informations sur les routeurs tels que; le nombre de paquets entrants, sortants, en erreur et la table des routeurs adjacents.

Transmission : ne contient que type Object Identifier (transmission number) qui permet d'identifier le type de media utilisé pour la transmission.

SNMP : requis pour chaque entité mettant en œuvre le protocole SNMP. Contient le nombre de message SNMP entrants et sortants, le nombre de mauvaises versions reçues ou de nom de communauté invalide, la répartition du type de requêtes reçues et envoyées (Get, GetNext, Set et trap).

4.3.RMON MIBs

4.3.1. RMON 1- MIB :

Compte tenu du nombre important de réseaux et de leur éparpillement, leur gestion est devenue plus nécessaire. Afin de mieux maîtriser cette gestion un équipement supplémentaire appelé enquêteur (probe) est mis en place, il contient RMON1-MIB pour fournir à la station de gestion des informations et des statistiques. RMON1-MIB est défini dans le RFC 1757, son OID est {1.3.6.1.2.1.16} et contient neuf groupes qui ont un statut optionnel, mais l'implémentation de quelques groupes nécessite d'autres, par exemple le groupe Filter nécessite le groupe capture.

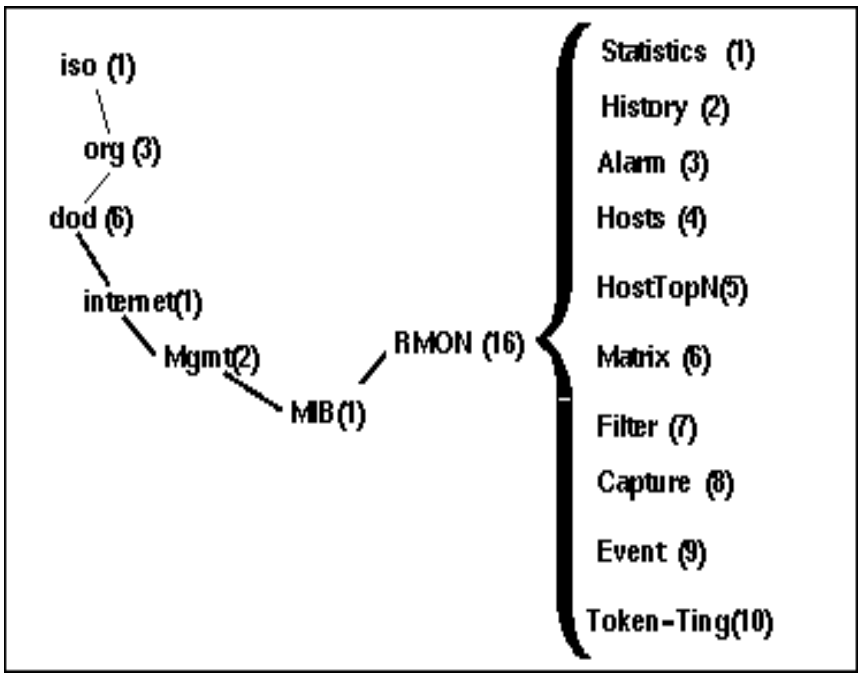


Figure 2. 5 : la MIB RMON

Groupe	Description
Statistics	Fournit des statistiques sur le Proxy, tel que le nombre et la taille des paquets, les broadcaste, les collisions…
History	Archive des échantillons statistiques périodes d'analyse
Alarm	Compare les échantillons statistiques avec les seuils prédéfinis activant des alarmes en cas de problème
Host	Maintient les statistiques sur les éléments du réseau, incluant les adresses MAC disponibles.
hostTopN	Fournit les rapports classés par la table des statistiques des éléments, indiquant quels sont les premiers éléments d'une catégorie particulière

Matrix	Stocke les statistiques dans une matrice de trafic qui suit toute conversation entre les éléments du réseau
Filter	Autorise les paquets à être appropriés grâce à un filtre
Capture	Autorise les paquets à être capturés après le passage sur un canal
Event	Contrôle la création et la notification des événements qui concernant Trap SNMP

Tableau 2.3 : Groupes Ethernet RMON MIB

4.3.2. RMON-2 MIB

RMON1-MIB traite des opérations de gestion seulement dans deux couches OSI : physique et liaison de données. De ce fait, elle ne peut avoir les informations sur les autres couches. Les RFC 2021 a étendu la RMON1-MIB en RMON2-MIB pour être capable de manipuler les autres couches en ajoutant dix nouveaux groupes dont les OID sont de {RMON.11} jusqu'à {RMON.20}. Ces dix groupes sont décrits comme suit :

Groupe	Description
protocolDir (11) (protocol Directory)	Enumère dans une table les inventaires des protocoles surveillés
protocolDist (12) (protocol Distribution)	Collecte l'ensemble des octets et des paquets des différents protocoles détectés dans un segment réseau
addressMap (13) nlHost (14)	• Fait correspondre les adresses de la couche réseau avec celle de la couche MAC. • Stocke ces informations dans des tables. • Mesure le trafic transmis ou reçu de chaque nouveau adresse de la couche réseau capturé par le probe. • Stocke l'information dans des tables.
nlMatrix (15)	• Mesure le trafic envoyé entre chaque paire d'adresses réseau capturé par le probe. • Stocke ces informations dans des tables.
alHost (16) (Application Layer)	• Mesure le trafic transmis ou reçu suivant un protocole à chaque adresse d'un nœud. • Stocke ces informations dans des tables.
alMatrix (17) (Application Layer Matrix)	• Mesure le trafic envoyé entre chaque paire d'adresses réseau suivant protocole. • Stocke ces informations dans des tables. • Ce groupe est similaire au groupe nlMatrix mais la différence réside dans l'intégration du protocole.
usrHistory (18) (user history)	• Combine les mécanismes des groupes alarm (3) et history (2) de RMON1 pour fournir l'historique de la collection des spécifications de l'utilisateur. • Stocke ces informations dans des tables.
probeConfig (19) (robeConfiguration) rmonConformance(20) (rmon Conformance)	• Contrôle la configuration des différents paramètres opérationnels, tels que les groupes d'Ethernet et Token Ring RMON qui sont supportés par le probe. • Spécifie les demandes d'adaptation des MIBs RMON2

Tableau 2. 4: Groupes RMON2 MIB

4.4.PRIVATE MIB

Les Privates MIBs sont des MIBs conçues par les constructeurs pour gérés leurs équipements. Les MIBs privées des sociétés trouvent leur place sous le nœud ***Enterprises*** dans l'arbre de nommage:

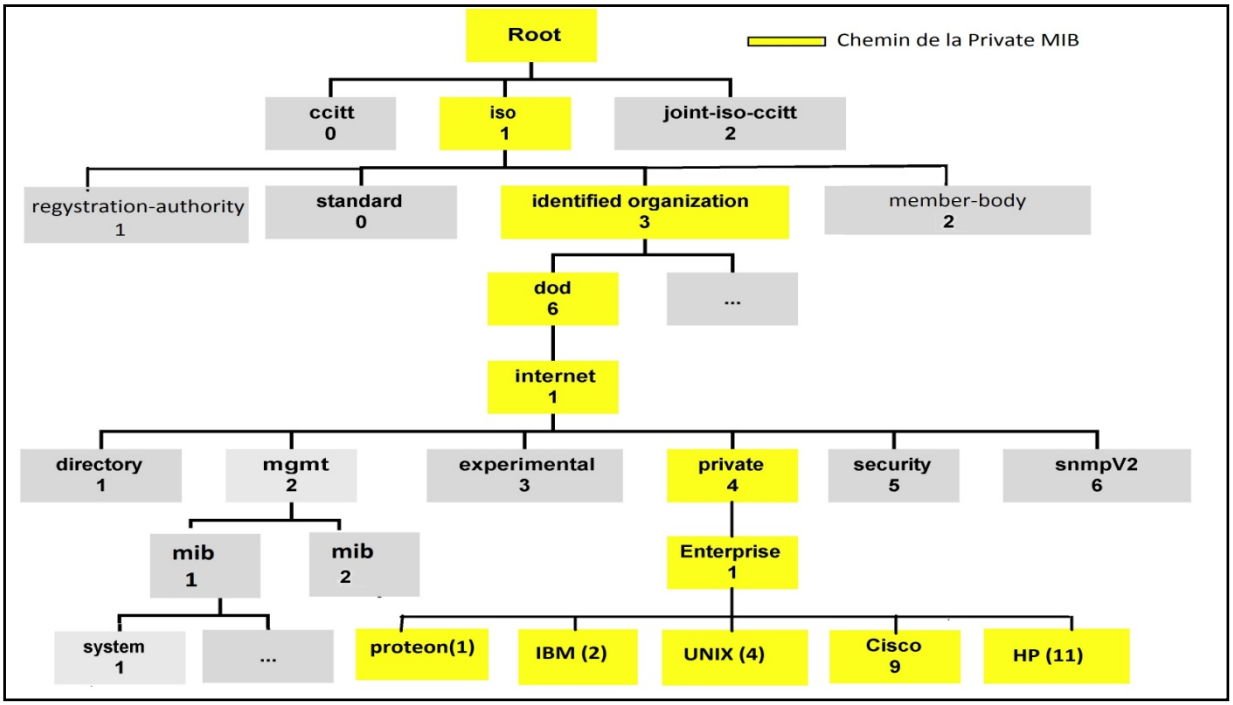


Figure 2.6 : private MIB

La MIB privée de la société Cisco a comme OID **1.3.6.1.4.1.9** qui correspond à: **iso.org.dod.internet.private.enterprises.cisco**

Les équipementiers implémentent des objets supplémentaires grâce à la branche Private Entreprise du MIT (Management Information Tree). Les routeurs travaillant au niveau 3 de l'OSI, cela permet de prendre en compte un grand nombre d'objets car on dispose de plus d'informations. Les routeurs Cisco introduisent ainsi environ 600 objets supplémentaires. Un utilitaire permet de « compiler » la MIB standard et la MIB propriétaire de façon à ce qu'elles soient toutes deux accessibles par l'administrateur de réseau de façon transparente. Par exemple, lorsque l'administrateur souhaite intégrer la gestion d'un nouvel équipement dans une plate-forme, sa première tache consiste à compiler la MIB que lui aura fourni le constructeur de cet équipement.

Une autre MIB appelée manager-to-manager spécifie comment les différentes plates-formes d'administration peuvent coopérer pour administrer un seul réseau. Une MNS(manager network station) peut être ainsi interroger par une autre NMS qui se comporte alors vis-à-vis d'elle comme un agent SNMP. Les fonctions d'interrogation, de

programmation des seuils d'alerte et de remontée d'alarmes peuvent ainsi être distribuées sur différentes machines.

5. Structure des informations de gestion (SMI)

Afin de bien localiser les nombreuses informations de la MIB proposées par chaque agent, une structure arborescente particulière appelée *structure of management information (SMI)* a été mise en place. Chacune de ces informations peut être retrouvée à partir de son nom de variable. Compte tenu de l'importance de ces variables, des logiciels ont été conçus pour permettre l'exploration de la MIB.

La SMI organise, nomme et décrit de façon arborescente les informations de la MIB, chaque objet est caractérisé par:

- un nom **OID (Object IDentifier)** qui identifie l'objet d'une manière unique
- une **syntaxe (ASN.1)** qui définit le type de données (entier, chaîne de caractères…)
- un **codage (BER)** qui décrit comment l'information est associée avec les objets administrés.

5.1.Nom d'objets OID (Object IDentifier)

Un objet est un programme répertorié dans une base de donnée, il est identifié de manière unique ; un sous objet est un objet hérité d'un autre objet. La racine de la structure arborescente (SMI) ne dispose pas de OID, elle a un nom de communauté qui est par défaut « public », elle est caractérisée par des valeurs numériques spécifiques qui désignent les trois branches connectées suivantes :

- ITU-T (International Télécommunications Union-Telecommunications) : OID= 0
- ISO (International Organisation for Standardization) : OID= 1
- ISO-ITU-T (cooperation ISO IUT-T): OID=2

Pour identifier une position particulière dans l'arbre, les valeurs numériques dans une chaîne sont énumérés et séparés par des points.

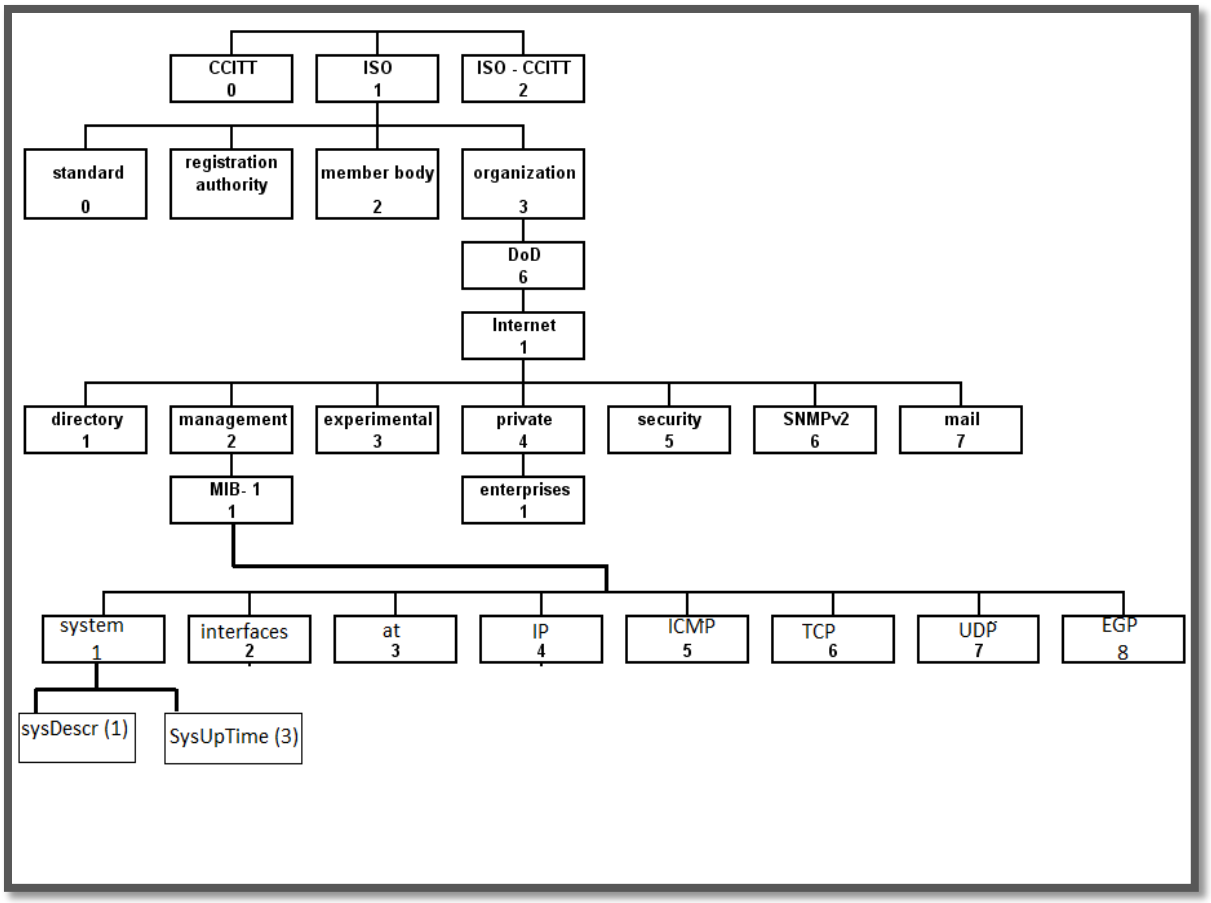


Figure 2.7 : Structure des OID

Exemple N°1 : Pour identifier l'objet *Security* au niveau de l'arbre, il y'a lieu de débuter par la racine et ensuite descendre jusqu'à trouver la chaîne :

{1.3.6.1.5}= ISO.Organization.Departement of Defense.Internet.Secutity}

Exemple N°2 : l'OID {1.3.6.1.2.1.1.3.0} spécifie la description du système *sysUpTime* de l'objet géré dans le groupe *System* du sous objet management, le zéro à la fin de l'OID indique la position finale de l'objet sysUpTime (feuille).

Il faut noter que l'objet *Internet* dispose de sept branches :

- **Directory (1)** sous objet {1.3.6.1.1} est réservé pour un usage futur.
- **Mgmt (2)** sous objet {1.3.6.1.2} inclut les MIBs standards il est géré par l'IANA (Internet Assigned Numbers Authority).
- **experimental(3)** sous objet{1.3.6.1.3} est utilisé pour les expériences Internet, une fois validées, elles se déplacent vers l'une des six autres branches Internet.
- **private(4)** sous objet (1.3.6.1.4} destinée aux équipementiers pour enregistrer leurs produits, cet objet est en plein développement.
- **security(5)** sous objet {1.3.6.1.5} réservé pour security-related objects.
- **snmpV2(6)** sous objet {1.3.6.1.6} pour les objets SNMP version 2.
- **Mail(7)** sous objet {1.3.6.1.7} pour les objets mail.

5.2.Le langage ASN.1

La station d'administration utilise le langage Abstract Syntaxe Notation 1 (ASN1) pour dialoguer avec les différents agents. Ce langage définit les spécifications pour la notation des données et fournit des règles pour combiner les éléments ou messages. Il est construit pour représenter la structure d'information (messages) au niveau de la machine utilisé en définissant :

- Les types de *variables* (simples, structurés, définis, étiquettes) et leurs valeurs.
- Les *macros* qui sont des instructions permettant de définir les objets à administrer.
- Les *modules* qui regroupent un ensemble d'informations.

5.2.1. Les Variables:

Le type de variables représente une classe de données qui définit la structure des variables que la machine doit avoir pour comprendre et traiter les informations. Il existe quatre sous-ensembles (Types simples, Types structurés, Types définis et Types Etiquettes).

a) Types simples :

Pour garder la simplicité du SNMP, le SMI utilise un certain nombre de sous ensembles de type de données ASN.1 qui sont les types *INTEGER, OCTET STRING, OBJECT IDENTIFIER* et *NULL*.

- *INTEGER* est un type simple avec une valeur unique positive, négative ou nulle.
- *OCTET STRING* est un type qui désigne les valeurs de séquences ordonnées de zéro ou plusieurs octets. SNMP utilise trois types de chaîne de caractères:
  - **DisplayString :** permet d'afficher les octets en caractères ASCII
  - **OctetBitString :** permet de spécifier que les chaînes ont une longueur supérieure à 32 bits.
  - **PhysAddress :** permet de représenter les adresses physiques (MAC) dans la version 2 de la MIB.
- *OBJECT IDENTIFIER* représente le type qui permet d'identifier les objets.
- *NULL* est un type avec une valeur unique appelée null, il est utilisé comme palliatif, mais pas couramment pour les objets SNMP. NULL est assigné aux valeurs d'une variable inconnue comme ce qui est du cas de GetRequest PDU

b) Les types structurés (Constructeurs):

Les constructeurs SEQUENCE et SEQUENCE OF définissent les tables et les vecteurs. Par convention, le nom d'un objet *table* possède un suffixe *table* et un nom pour un objet vecteur a un suffixe *Entry*.

- *SEQUENCE* est un type qui se réfère à un type ordonné, une liste de types ou un type fixe qui peuvent être définis de manière unique et optionnelle. Chaque valeur du nouveau type est une liste ordonnée ou une liste de valeurs.
- *SEQUENCE OF* est un type structuré qui définit une référence à un ou plusieurs types existants, chaque valeur dans le nouveau type est un ordonnancement d'une ou plusieurs valeurs. il définit les vecteurs dans les tableaux contrairement à *SEQUENCE* qui utilise des éléments d'un seul type ASN.1.

c) Les types définis :

Les types définis sont soit simples soit types complexes généralement il sont descriptifs, les principaux types sont définis dans le RFC 1155 et incluent les types suivants: *Networkadress, Ipadress, Counter, Gauge, Timeticks et Opaque.*

- *Network Address* est un type construit pour représenter les adresses à partir de plusieurs familles de protocoles. Le type de primitive (type simple ou défini), *CHOISE* fournit des alternatives entre les autres types.
- *Ipaddress* est un type d'application qui représente une adresse IP sur 32 bits, il est représenté comme un *Octet String* de longueur 4 octets.
- *Counter* est un type d'application qui représente un entier positif qui s'incrémente tant que sa valeur maximale (2^{32}–1), n'est pas attente. Par convention le nom d'objet utilisant *Counter* se termine par un (s) minuscule.
- *Gauge* est un type d'application qui représente un entier positif qui augmente ou diminue et qui se bloque à la valeur maximale qui est semblable à la valeur maximale du *Counter.*
- *Timeticks* est un entier positif qui mesure le temps (centièmes de seconde) depuis un temps initial.
- *Opaque* est un type qui permet de convertir les données ASN.1 en données BER (Basic Encoding Rule). Ce type est très peu utilisé avec le protocole SNMP.

d) Les types étiquettes (tagged type) :

Les étiquettes permettent à la machine de distinguer entre les différents objets. Les étiquettes utilisent les types définis au préalable, puis ajoutent des informations uniques. ASN.1 définit quatre classes d'étiquettes; universal, application, context specific, et private.

5.2.2. Les Macros

La définition d'un objet au sein d'un MIB est effectuée à base de macros. La macro utilisée pour définir un objet MIB est nommée *OBJECT-TYPE*. Cette dernière est reconnue dans les deux versions de SMI. Par contre, dans SMIv2 quelques champs ont été renommés et d'autres ont été ajoutés. Les différents champs utilisés par OBJECT-TYPE sont les suivants:

- *SYNTAX:* il permet de spécifier la syntaxe d'un objet. La syntaxe est construite en utilisant les types universels et les types application;
- *MAX-ACCESS:* il définit le mode d'accès permis à une instance d'un objet. Les valeurs possibles sont: *read-only, read-write, read-create, not-accessible et accessiblefor-notify;*
- *STATUS:* désigne la validité de la définition de l'objet. Les valeurs possibles de ce champ sont *current* (valide), *deprecated* (remplacée par une autre) et *obsolete* (non valide et ne peut plus être appliquée);
- *DESCRIPTION:* il correspond à une description textuelle (chaîne de caractères) de la sémantique du type d'objets en question;
- *UNITS:* contient une définition textuelle des unités associées à un objet (par exemple, seconde pour le cas du temps);
- *REFERENCE:* contient une référence textuelle à un objet défini dans d'autres modules MIB;
- *INDEX:* définit comment les rangées d'une table sont indexées. Ce champ illustre de façon ordonnée les différents champs qui entrent dans la composition de l'index d'une table. II n'y a aucune restriction sur le nombre d'index. Typiquement, ils correspondent à des objets colonnes de la table;
- *AUGMENTS:* permet d'étendre le nombre de colonnes d'une table sans toucher à sa structure *(sans* redéfinir la table). AUGMENTS reçoit comme paramètre l'identificateur d'une rangée *d'une* autre table. Cette dernière est appelée *table de base;* par contre, la table utilisant le champ AUGMENTS est appelée table d'augmentation. Une table de base peut être étendue par plusieurs tables d'augmentation. Cet aspect est semblable à la notion d'héritage;
- *DEFVAL:* spécifie la valeur à affecter à une instance d'un objet colonne lors de sa création. Aucune valeur initiale n'est attribuée aux objets correspondant aux index d'une table, ni à un objet scalaire. Seuls les objets colonnes accessibles en mode read-create peuvent avoir un champ DEFVAL.

Il faut noter que les champs SYNTAX, ACCESS (ou MAX-ACCESS) *et* STATUS (INDEX ou AUGMENTS pour les tables) sont obligatoires lors de la définition d'un objet. Par contre, les champs UNITS, REFERENCE, DESCRIPTION et DEFVAL sont optionnels.

a) Définition des tables

La macro OBJECT-TYPE est utilisée pour définir une table d'objets. Une table SNMP est composée d'un ensemble de rangées et de colonnes. L'argument du champ SYNTAX pour une table doit être *SEQUENCE OF <sequence>*. Enfin, la valeur du champ ACCESS ou du champ MAX-ACCESS d'une table doit être *not-accessible* (vu qu'une table n'est pas directement accessible avec les opérations de SNMP). Voici un exemple de définition d'une table:

if Table OBJECT-TYPE

SYNTAX SEQUENCE OF IfEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "...n ::= { interfaces 2 }

Suite à la définition d'une table, vient la définition de ses rangées. Le constructeur OBJECT-TYPE est aussi utilisé dans ce but. L'argument du champ SYNTAX d'une rangée doit être un identificateur pour une séquence (l'identificateur de la rangée). La valeur des champs ACCESS ou MAX-ACCESS doit être aussi *not-accessible.* La valeur du champ STATUS d'une rangée donnée doit être la même que celle de la table. Le champ INDEX ou AUGMENTS spécifie comment les instances des objets colonnes de la table sont identifiées.

SMI mentionne que la valeur OID attribuée à une rangée doit être la même que celle de la table dont elle fait partie, tout en ajoutant la valeur 1 à la fin de cette même valeur OID. Par exemple, si la valeur OID d'une table nommée *printerTable* est *z,* alors la valeur OID de sa rangée (printerEntry) sera *x.1.* Voici un exemple de définition de rangées:

ifEntry OBJECT-TYPE SYNTAX IfEntry

MAX-ACCESS not-accessible
STATUS carrent
DESCRIPTION n..." INDEX { if Index } ::= { ifTable 1 }
IfEntry ::= SEQUENCE {
if Index INTEGER,
ifDescrOCTET STRING,
if TypeINTEGER,
ifNtu INTEGER,
ifSpeed Gange,
ifPhysAddressOCTET STRING,
ifAdminStatusINTEGER,
ifOperStatus INTEGER,
ifLastChange TimeTicks,
ifInOctets Counter,

ifInUcastPkts Counter,
ifInNUcastPkts Counter,
ifInDiscards Counter,
ifInErrors Counter,
ifInUnknovnProtos Counter,
ifOutOctets Counter,
ifOutUcaatPkta Counter,
ifOutNUcastPkts Counter,
ifOutDiscards Counter,
ifOutErrora Counter,
ifOutgLen Gange

5.2.3. Modules

Dans les paragraphes précédents, nous avons présenté et défini les objets MIB utilisés pour représenter les ressources d'une entité gérée par un système de gestion des réseaux. En plus, nous avons défini les tables MIB qui servent à regrouper un ensemble d'objets. Avec ces deux notions (objet et table), on peut écrire une MIB. Tout est structuré en groupes d'objets. La façon dont cette structure est représentée est l'objet de cette sous-section.

Une spécification d'une MIB SNMP est structurée en trois sections:

1. **Description des thèmes:** cette section fait la correspondance entre les ressources gérées et les définitions des informations de gestion contenues dans le(s) module(s) MIB. Elle permet aussi d'expliquer l'interaction entre les objets d'une MIB complexe.
2. **Module(s) de la MIB:** cela constitue la partie principale d'une MIB. Un module est écrit dans la syntaxe ASN.1.
3. **Références:** cette section fournit la liste de toutes les sources pouvant aider à mieux comprendre les objets gérés.

Un module MIB est constitué des sections suivantes :

1. Délimiteurs *(<moduleName>* et END): ils permettent de nommer le module et de le délimiter des autres modules.
2. Section de *liaison (<importedItems>): dans* cette section sont spécifiés les éléments définis dans d'autres modules et utilisés dans le module courant.
3. Identité du module *(<moduleIdentityDefinition>):* cette section identifie les modules MIB.
4. Définitions (<definitions>): elle contient les définitions de tous les objets et groupes d'objets à gérer, faites au moyen des macros. L'ensemble de ces macros est le suivant:

- *IMPORTS:* elle est utilisée pour spécifier des éléments définis dans d'autres modules MIB et qui sont utilisés par le module courant.
- *MODULE-IDENTITY* elle est utilisée pour spécifier des informations relatives aux modules MIB SNMP. Ceci inclut l'historique de révision des modules, le nom de l'organisation qui a défini le module, l'adresse du groupe de support technique relative à ce module et une description de haut niveau des modules. Cette macro doit être utilisée une fois et doit être placée au début du module.

- *OBJECT-IDENTITY:* permet de spécifier un identificateur dans un module MIB. Une valeur OID est toujours affectée à cette macro.

- *TEXTUAL-CONVENTION:* elle est utilisée pour créer un nouveau type de données. Ceci est fait en ajoutant d'autres règles d'utilisation à un type de données de base ou à un type de données déjà défini.

- *OBJECT-TYPE:* elle est utilisée pour définir un objet.

- *SEQUENCE et SEQUENCE OF:* la macro SEQUENCE OF est utilisée pour définir les rangées d'une table. Par contre, la macro SEQUENCE est utilisée pour spécifier les objets colonne d'une rangée d'une table.

- *NOTIFICATION-TYPE:* elle est utilisée pour spécifier les événements qu'un agent peut rapporter à des stations de gestion.

- *OBJECT-GROUP:* elle est utilisée pour définir un ensemble de types d'objets inter-reliés. Ceci a pour but de mentionner qu'une relation existe entre un ensemble d'objets, ce qui permet une meilleure organisation des modules MIB.

- *NOTIFICATION-GROUP:* c'est l'équivalent de la macro OBJECT-GROUP, mais pour les événements émis par un agent
- *MODULE-COMPLIANCE:* cette macro permet de définir les aspects obligatoires lors de l'implantation d'un ou de plusieurs modules MIB.

Le champ MODULE est utilisé une ou plusieurs fois pour mentionner chaque module requis. Chaque section MODULE spécifie les groupes d'objets nécessaires (MANDATORY-GROUPS) et les groupes d'objets optionnels pour l'implantation du module. Pour qu'une implantation soit conforme aux spécifications, elle doit implanter tous les objets appartenant aux groupes mentionnés *dans* le champ MANDATORY-GROUPS. Pour chaque groupe qui est conditionnellement obligatoire ou optionnel, il y a spécification d'un champ GROUP. Le champ DESCRIPTION décrit les conditions sous lesquelles un groupe d'objets est obligatoire (par exemple si un protocole particulier est implanté ou si un autre groupe est défini).

- *AGENT-CAPABILITIES:* cette macro est utilisée pour documenter les fonctionnalités d'un agent SNMP. La macro spécifie des raffinements ou des modifications relatifs aux macros OBJECT-TYPE définies dans les modules.

Le champ PRODUCT-RELEASE mentionne la version de l'agent décrit par le champ DESCRIPTION. Le reste de la macro contient une section pour chaque module MIB pour lequel l'agent réclame une implémentation complète ou partielle. La description de chaque module commence par un champ SUPPORTS qui identifie le module. Ensuite, le champ INCLUDES spécifie la liste des groupes d'objets appartenant à ce module et qui sont implantés par l'agent. Enfin, pour chaque groupe supporté, est spécifiée la liste des objets implantés par l'agent, d'une façon différente de ce qui a été défini par la macro OBJECT-TYPE.

5.3.Encodage Basic Encoding Rule (BER)

Ces règles servent à encoder les spécifications décrites par ASN.1. (BER Basic Encoding Rule) et permettent de construire des messages qui seront envoyés sur le réseau de sorte qu'à leur réception ils puissent être décodé indépendamment du type de machine utilisée. Trois champs sont nécessaires dans BER:

- **Le *type*** de données est le premier champ. Ce champ est divisé en trois parties. Le deux bits de poids fort indiquent la nature de l'étiquette :
  - 00 : UNIVERSAL
  - 01 : APPLICATION
  - 10 : CONTEXT
  - 11 : PRIVATE

Le bit suivant indique si le reste de données correspond à un type primitif (type simple ou type défini) (0) ou à une construction (type structuré) (1). Par exemple, un entier est primitif alors qu'une séquence est une construction. Les cinq bits restant définissent le type. Si le nombre de types ne peut pas être codés sur les cinq bits restants, ce champ est rempli par les bits à 1 et un entier est ajouté à la place afin de coder le type.

La *longueur* des données est codée dans le deuxième champ. Il code uniquement la longueur de données si le bit de poids fort est mis à 0, la longueur tient sur un octet. Si ce bit est à 1(c'est-à-dire pour les longueurs supérieures à 128 octets) un second octet est utilisé. Les *données* sont placées dans le troisième champ.

Conclusion

En résumé, on peut dire que la MIB (Management Information Base) définit la gestion des informations concernant les objets communs, le nom, et la syntaxe des dispositifs géré. La compréhension de la structure des MIB est nécessaire pour pouvoir développer des applications de gestion basées sur SNMP. Dans ce chapitre, nous avons expliqué les notions de base de la base de données MIB, des ses différentes versions, puis nous avons parlé du SMI (structure of management information) où les trois points principal ont été, les Nom d'objets OID (Object IDentifier), le langage ASN.1 et l'Encodage Basic Encoding Rule (BER). Le prochain chapitre sera entièrement consacré au protocole SNMP.

Chapitre III

Simple Network Management Protocol (SNMP)

Introduction

La surveillance et la détection de pannes sont des aspects important dans plusieurs domaines. Dans un réseau informatique, il est en effet difficile de déterminer si un périphérique est non fonctionnel ou simplement lent. Or, cette distinction est importante: dans le premier cas une reconfiguration est nécessaire, dans le second, une simple attente suffit souvent. Implémenter une détection de pannes efficace est un problème difficile, ce qui fait qu'en général, les programmes utilisent des approches simplistes comme des temps limites fixés.

Pour contrer ce problème, la notion de service de détection de panne a été proposée depuis plusieurs années. Un tel service offre de nombreux avantages: il permet de n'avoir qu'une seule implémentation partagée par plusieurs applications, cette implémentation peut-être changée sans affecter les applications, et le service peut utiliser des techniques complexes sans compliquer les applications.

Plusieurs services de détection de pannes ont été proposés, mais aucun n'est utilisé en dehors de sa niche d'origine. Une des raison à cela est le fait qu'il n'existe aucune interface standard pour un tel service.

Dans ce chapitre nous présentons le protocole SNMP (Simple Network Management Protocol). Ce standard est très largement utilisé pour la surveillance de l'infrastructure réseau.

1. C'est quoi SNMP?

SNMP (Simple Network Management Protocol) est un protocole simple de gestion de réseau développé par un groupe de travail de l'IETF dans le cadre de la définition d'un système de gestion pour les réseaux. Le protocole SNMP fournit un moyen de surveiller et de contrôler les périphériques réseaux, ainsi que de gérer les configurations, la collecte de statistiques, les performances et la sécurité. Plusieurs versions se sont succédées dans le temps dont les principales sont: SNMPv1 (1990), SNMPv2 (1993) et SNMPv3 (1999). Ce protocole se situe au niveau de la couche application et de la couche transport du modèle TCP/IP et permet le dialogue entre la station d'administration et les équipements dotés d'agent SNMP.

La station d'administration permet à travers ce dialogue de:

- Contrôler un réseau à distance en interrogeant ses éléments sur leur état.
- Modifier leur configuration.
- Gérer les logiciels et les bases de données à distance.

2. Historique

La première version de SNMP, SNMPv1, a été conçue à la fin des années 80 et standardisée dans le courant de l'année 1990. Sa conception permettait de gérer la surveillance, la détection des pannes, la reconfiguration, etc. Cependant, un certain nombre de lacunes persistaient : manque de hiérarchie, peu de codes d'erreur et de notifications, faibles performances, sécurité laxiste, etc.

L'ensemble de ces problèmes a entraîné le développement d'une nouvelle version de SNMP, nommée SNMPv2, et dont la conception a commencé en 1993. Toutefois, plusieurs éditeurs ont rejeté les standards proposés, conduisant à la création d'autres normes :

- **SNMPv2p** : Beaucoup de travaux ont été exécutés pour faire une mise à jour de SNMPv1. Ces travaux ne portaient pas seulement sur la sécurité. Le résultat est une mise à jour des opérations du protocole, des nouvelles opérations, des nouveaux types de données. Cette version est décrite dans les RFC 1441, RFC 1445, RFC 1446, RFC 1448 et RFC 1449.
- **SNMPv2c:** Cette version du protocole est appelée « community string based SNMPv2 ». Ceci 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 dans les RFC 1901, RFC 1905 et RFC 1906.
- **SNMPv2u**: Cette version du protocole utilise les opérations, les types de données de SNMPv2c et la sécurité basée sur les usagers. Cette version est décrite dans les RFC 1905, RFC 1906, RFC 1909 et RFC 1910.
- **SNMPv2***: Cette version combine les meilleures parties de SNMPv2p et SNMPv2u, mais les documents qui décrivent cette version n'ont jamais été publiés.

La version la plus utilisée de SNMP est actuellement SNMPv2c, mais la tendance s'inverse avec l'introduction en 1999 de la troisième version du protocole : SNMPv3. Cette version ajoute à la précédente une sécurité plus importante, ainsi qu'une gestion hiérarchisée, mais sa complexité accrue entraîne des difficultés d'implémentation et une charge de mise en œuvre plus délicate que sur les versions précédentes.

3. SNMP Version 1

3.1.Présentation

La version originale de SNMP (SNMPv1) dérive du protocole SGMP (Simple Gateway Monitoring Protocol) qui a été publié en 1988. SNMP utilise une architecture basée sur un modèle particulier ***client/serveur*** *(Manager/Agent)*, c'est-à-dire qu'il y'a un seul *client (Manager)* et plusieurs *serveurs (agents).* Le principe de cette architecture est basé sur la collecte d'informations relatives aux équipements à gérer. Le manager exécute des requêtes qui lui permettent de gérer et de contrôler les agents.

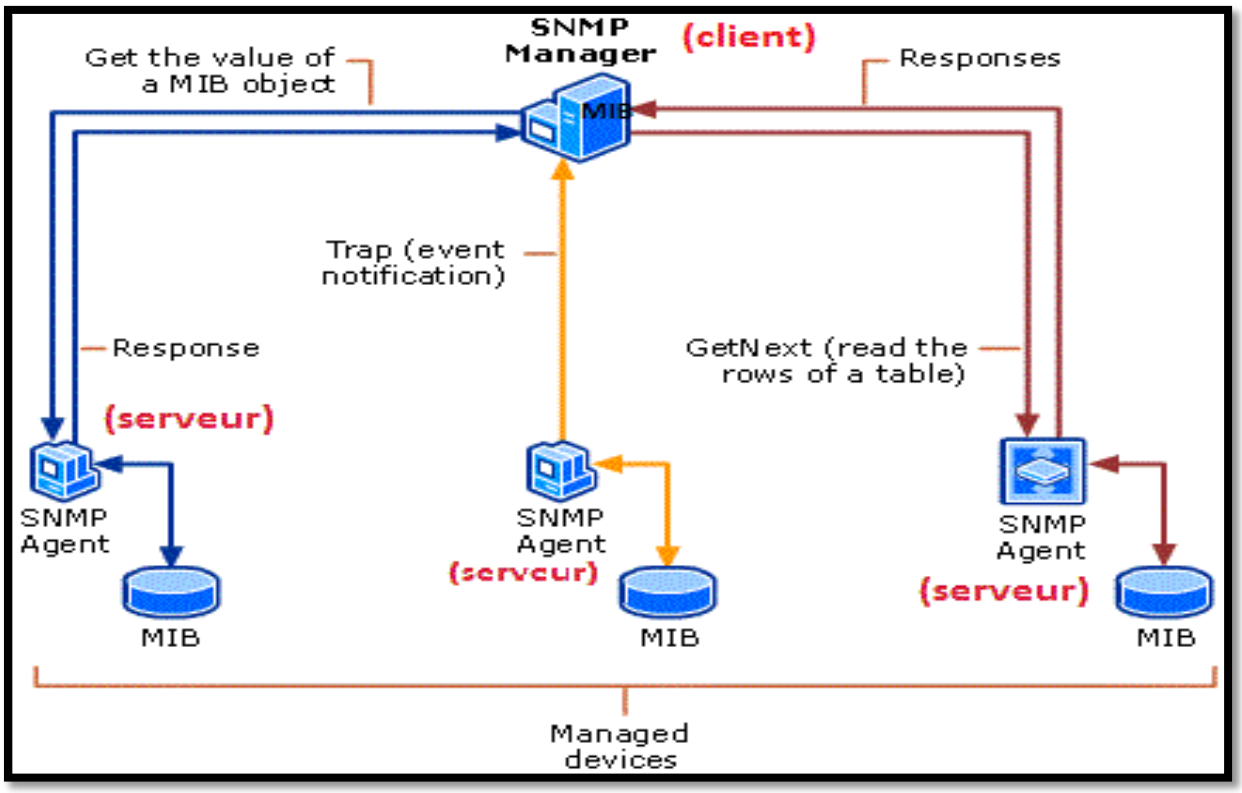


Figure 3.1 : Architecture manager/agent de SNMPv1

Le système d'administration réseau se compose des éléments suivants:

- Des nœuds administrés (Managed Node) chacun contenant un *agent SNMP*.
- Une station d'administration (Network Management Station).
- Un protocole réseau (*SNMP*) qui permet à la station d'administration (NMS) d'échanger des informations d'administration avec les agents (MN).
- Un certain nombre de bases de données(MIB agent et MIB manager)

3.2.Les requêtes SNMPv1

Les requêtes sont les messages que s'échangent la station de gestion (plate-forme d'administration ou Manager) et les agents administrés via le protocole SNMP, ces messages étaient au nombre de cinq (version 1): *Get*, *GetNext*, *Set*, *GetResponse* et les *Trap*.

Dans les versions ultérieures de nouveaux messages ont été définis pour permettre de nouvelles fonctionnalités au protocole SNMP.

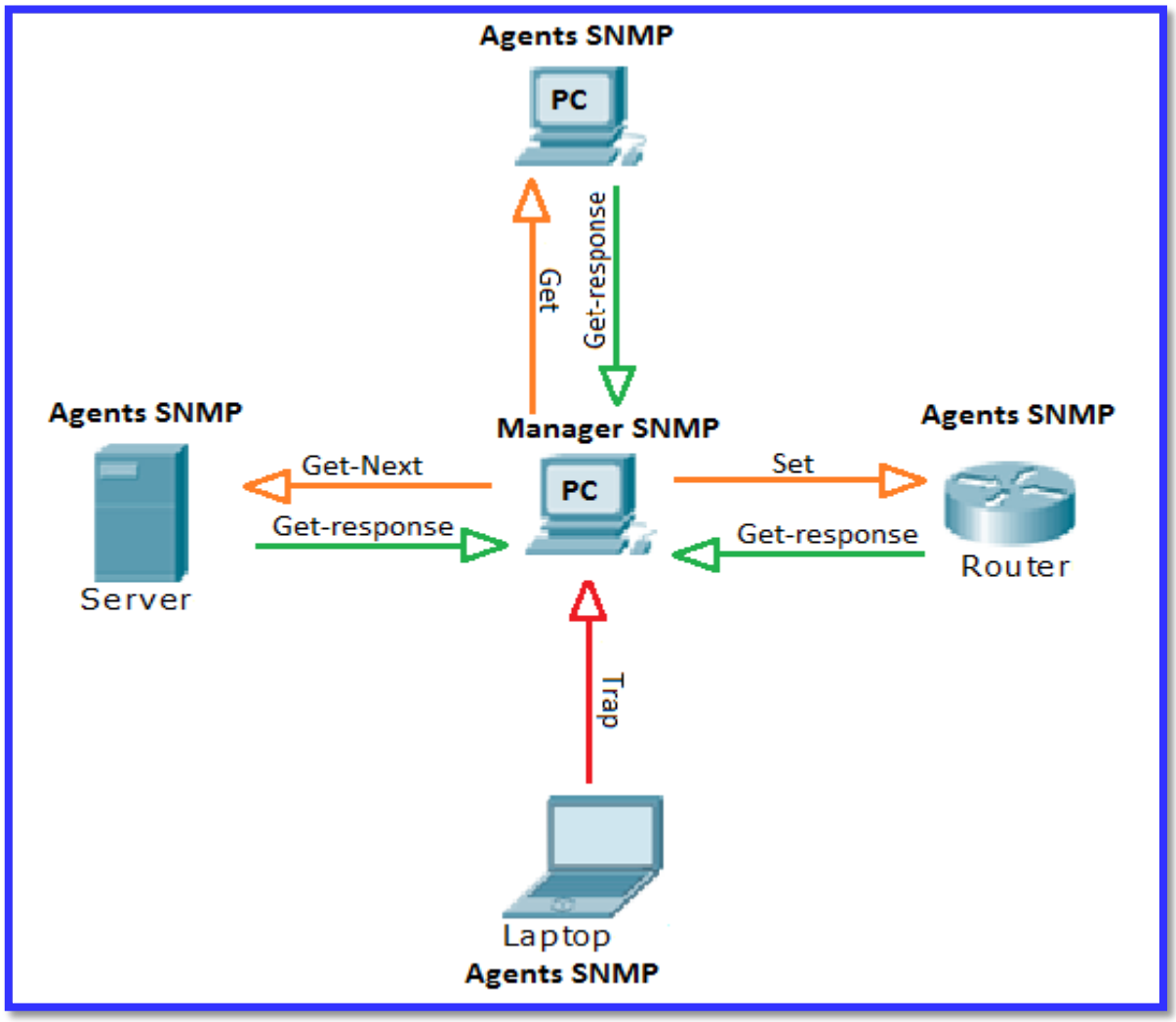


Figure 3.2 : Exemple de requêtes SNMPv1

Sur cette figure nous pouvons observer la station de gestion qui envoie des requêtes (*Get*, *GetNext*, *Set)* aux équipements, qui à leurs tours lui répondent (*GetResponse)*, et un équipement qui envoie une requête (*Trap)* pour signaler un événement. La figure suivante nous donne l'architecture interne du protocole SNMP dans le modèle TCP/IP.

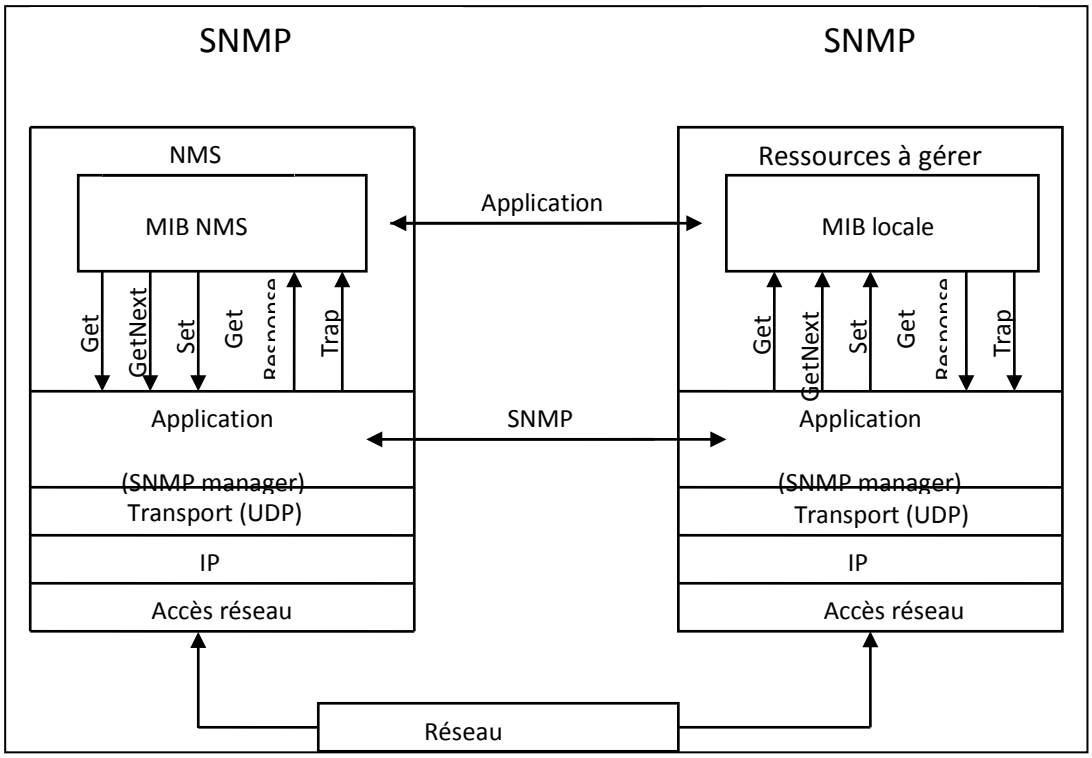


Figure 3.3: architecture interne du protocole SNMP

3.2.1. Requêtes Manager et requêtes agent

Parmi les cinq types de messages ou requêtes qui sont échangés entre les agents et le manager. Trois de ces requêtes (*Get* , *Set* ,*GetNext*) sont initialisées par le manager et auxquelles l'agent doit répondre grâce à la requête (*GetReponse)*. La seule opération initialisée par un agent est une interruption qui permet d'alerter le système de gestion de la présence d'un événement extraordinaire, tel que la violation de mot de passe.

Le protocole SNMP utilise le port UDP 161 pour les requêtes *Get, GetNext, Set et GetResponse* et le port UDP 162 pour les *Traps* .

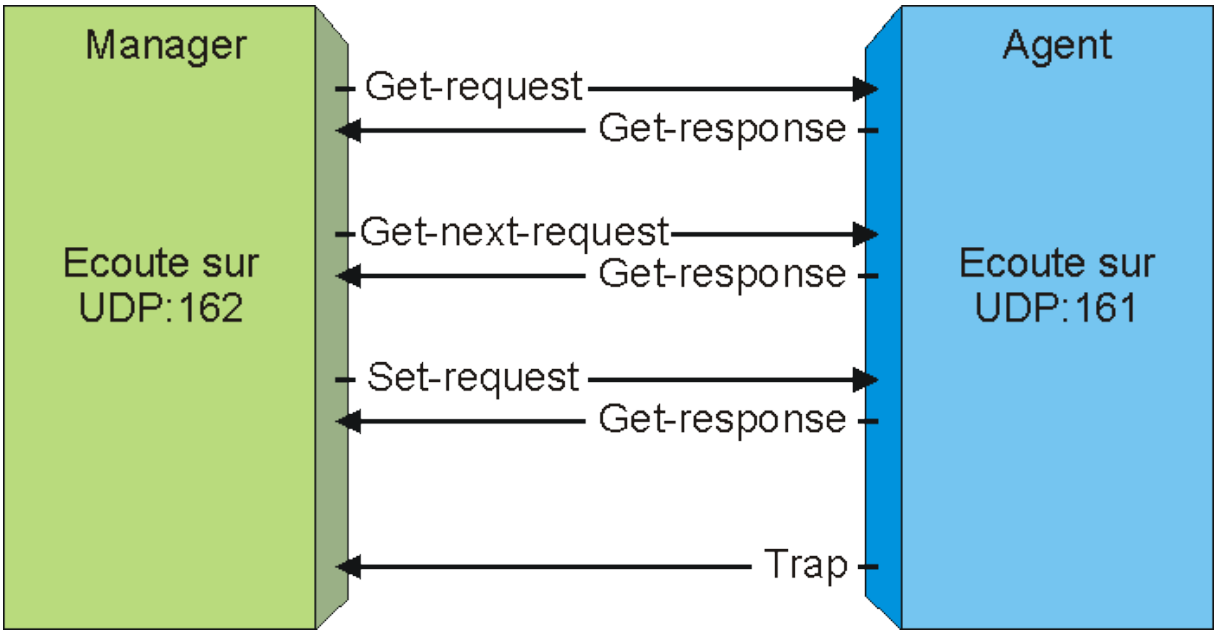


Figure 3.4 : requête SNMP et numéros de port

GetRequest (Get): c'est une requête initialisé par le manager dont l'objectif est l'obtention de la valeur courante d'un objet de la MIB géré par l'agent.

GetNextRequest (GETNEXT): c'est une requête initialisé par le manager dont l'objectif est obtention de la valeur courante du prochain objet de la MIB géré par un agent à partir d'un objet courant.

SetRequest (SET): c'est une requête initialisé par le manager dont l'objectif est mise à jour de la valeur courante d'un objet de la MIB géré par un agent.

GetResponse: c'est une réponse initialisé par l'agent dont l'objectif est le renvoi de la valeur d'un objet de la MIB géré par un agent.

Trap: le message Trap est une requête initialisé par l'agent pour signaler un événement, elle est subdivisé en 6 types suivants:

- **ColdStart :** démarrage à froid de l'agent SNMP. C'est à dire que certaines données reçues précédemment par le manager peuvent être invalides.
- **Warmstart :** démarrage à chaud de l'agent SNMP, les données envoyées au manager restent valides, même celles envoyées avant ce démarrage
- **LinkDown :** un lien physique ne fonctionne plus (panne d'un adaptateur).

- **LinkUp :** un lien physique fonctionne (adaptateur mis en fonctionnement).
- **AuthentificatlonFailure :** une requête SNMP est refusée due à une community non valide.
- **EgpNelghborloss :** perte d'un lien de communication (gated ne peut plus communiquer avec son voisin Egp).

3.3.Encapsulation du message SNMP

La couche application envoie l'ensemble de la trame aux couches inférieurs pour être encapsulée et transformée en datagrammes, puis en une trame IP jusqu'à la couche réseau.

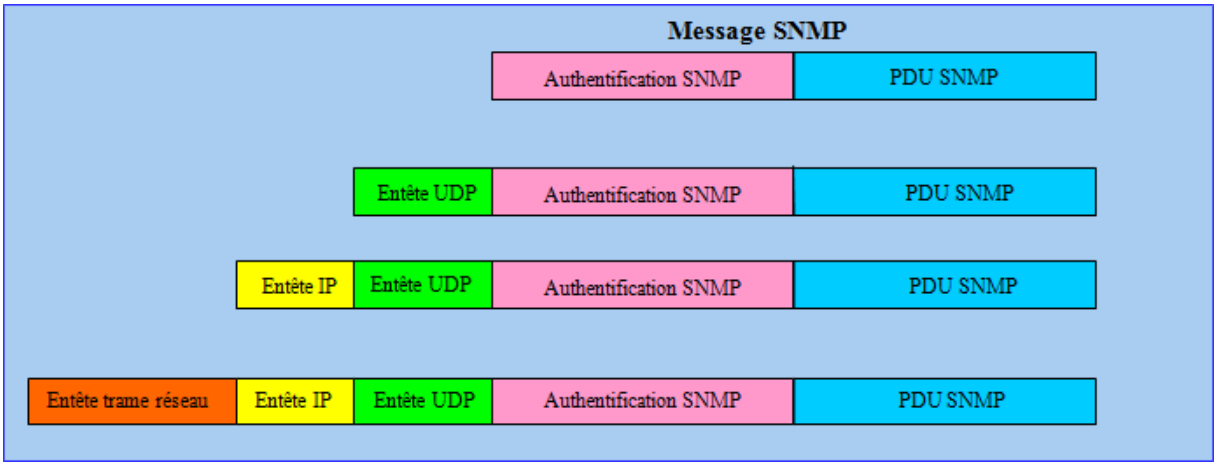


Figure 3.5: encapsulation du message SNMP

L'utilisation du protocole UDP est justifiée pour sa simplicité, sa rapidité, ce qui permet de signaler très rapidement des alarmes au manager.

3.4.Format de trame SNMPv1

3.4.1. Format des requêtes Manager et réponses de l'agent

La structure générale d'une requête SNMPv1 est donnée par la figure ci-dessous:

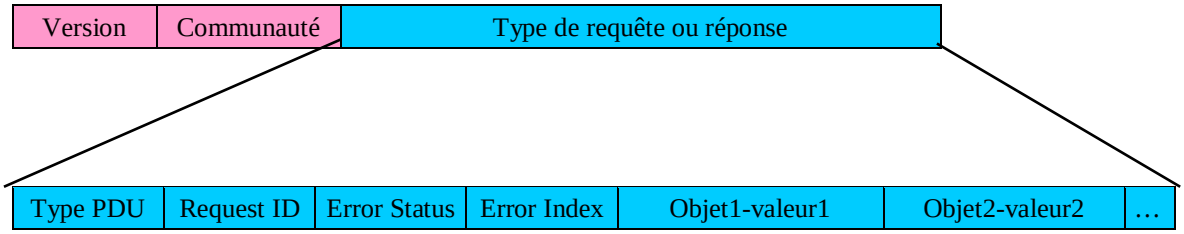


Figure 3.6: Structure des PDUs GetRequest, GetNextRequest, SetRequest et GetResponse

- Le champ *version* indique la version du protocole SNMP (0=SNMPv1, 1=SNMPv2, 2=SNMPng et 3=SNMPv3).
- Le champ *communauté* est une chaîne de caractères qui est comme un mot de passe de validation d'une requête SNMP par l'agent (GET, GETNEXT, SET) ou par le manager (TRAP). Si la communauté est incorrecte, la requête est rejetée. La communauté passe en clair sur le réseau. C'est une faille importante de sécurité de SNMPv1. Il faut attendre les versions ultérieures de SNMP pour combler cette lacune.
- Le champ *type de requêtes ou réponse* est composé comme suit :

  - Le champ PDU type : Spécifie le type de PDU que le message SNMP contient :

PDU	La valeur du champ PDU Type
GetRequest	0
GetNextRequest	1
GetResponse	2
SetRequest	3
Trap	4

Tableau 3.1 : PDUs SNMPv1

  - Le champ RequestID est de type INTEGER qui établit une liaison entre la demande du manager et de la réponse de l'agent.
  - Le champ ErrorStatus est de type INTEGER entier énuméré qui indique une opération normale (noError) ou un de ces 5 types d'erreurs :

Erreur	**Valeur**	**Signification**
noError	0	Opération manager/agent correcte
TooBig	1	La taille du PDU GetResponse demandé excède une limitation
noSuchName	2	Le nom d'objet demandé n'est pas compatible avec les noms disponibles dans la MIB
badValue	3	Un SetRequest contient une incompatibilité entre le type, la longueur et la valeur de la variable
readOnly	4	Essai de modification d'une variable en lecture
genErr	5	Autres erreurs, non définies qui sont apparues.

Tableau 3.2 : Valeurs du champ Error Status

- Si une de ces 5 erreurs se présente, le champ *ErrorIndex* définit une entrée dans la liste des variables qui sont à l'origine du problème. Par exemple si on a une erreur readOnly, le champ *Error Index* contient la valeur 4.
- Les variables d'enchaînement (VarBind) doivent lier le nom de la variable avec sa valeur.
- VarBind List est une liste de variables d'enchaînement (VarBind).
- Dans le champ variable Binding PDUs SNMP, le mot Object identifie le nom de la variable (OID et la valeur) avec la valeur à communiquer, autrement dans le messages GetRequest ou GetnextRequest une valeur de type de données spéciale NULL est utilisé.

3.4.2. Format de la requête TRAP

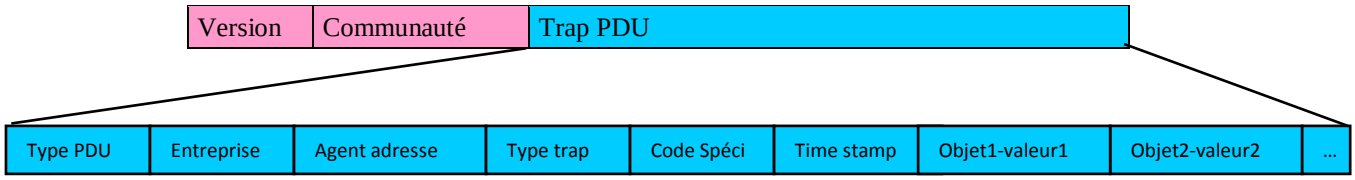


Figure 3.7 : structure trame PDU

3.5.Inconvénients du SNMPv1

Le premier défaut de SNMP est qu'il contient des failles de sécurité à travers lesquelles des intrus peuvent accéder aux informations transitant sur le réseau et provoquer par exemple un shut-down de terminaux. La solution à ce problème a été apportée par le SNMPv3 qui implémente des mécanismes de sécurité en ce qui concerne le caractère privé des données, l'authentification et le contrôle d'accès.

SNMP utilise le protocole de transport UDP (peu fiable) qui fonctionne en mode non connecté, il n'y a donc pas de reprise en cas d'erreur, ni de contrôle de flux. Le Manager surveille son environnement en procédant à des interrogations régulières de ses agents, c'est ce que l'on appelle le Polling. Cette surcharge de trafic n'est pas trop gênante sur un réseau local mais devient embarrassante via le réseau public.

4. SNMPv2 (Version 2)

La version 2 du protocole SNMP a été conçue pour réduire la surcharge du trafic (polling) généré par la version 1 ce qui se traduit par une limitation des flots d'information de contrôle par une nouvelle commande GetBulk et une commande GET améliorée. Pour la prise en charge de la coopération entre managers, SNMPv2 introduit la nouvelle commande INFORM qui permet l'échange d'informations entre les MIBs des différents managers. Un Manager utilise cette commande pour envoyer une information non sollicité (signaler un débit excessif sur une ligne de communication) à un autre Manager. Cette information est consignée dans la nouvelle MIB Manager à Manager.

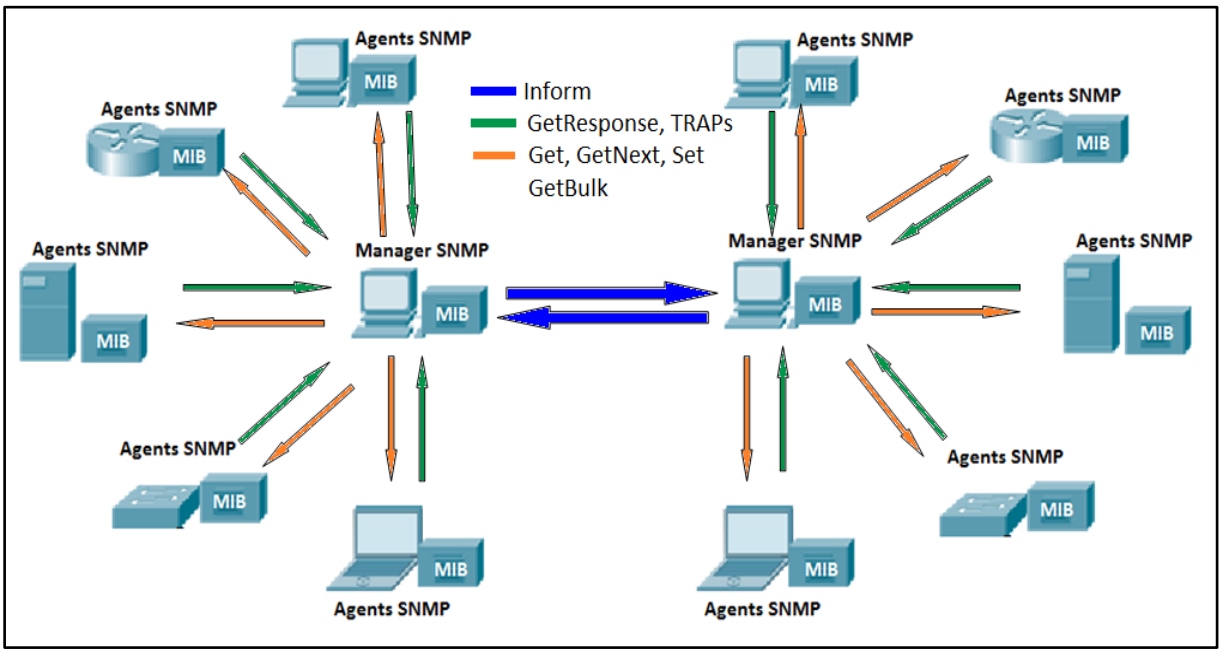


Figure 3.8: Architecture manager/agent dans la version 2

Cette architecture est légèrement différente de celle de la version 1 du protocole SNMP, l'apport ici est la possibilité de communication entre managers.

4.1.Intérêts de SNMPv2

- Un nouveau modèle administratif, possibilité de décentraliser la gestion.
- L'introduction de la notion de dialogue de manager à manager (primitive inform).
- La définition d'une nouvelle primitive (GetBulk).
- L'amélioration des messages de type Trap.

- Tentative d'introduction de mécanismes de sécurité qui garantissent l'authentification par message digest (MD5), la confidentialité des messages SNMP par cryptographie (DES) et un mécanisme d'anti rejet par synchronisation des horloges.

4.2.Le PDU GetBulkRequest

L'une des nouveautés apportées avec SNMPv2 est associée à ce type de messages. GetBulkRequest a pour but de minimiser le nombre de messages échangés pour lire un grand nombre d'informations de gestion. Ce genre de PDU utilise le principe de sélection par ordre lexicographique.

Parmi les avantages de GetBulkRequest, on distingue principalement:

- une réduction de la taille et de la complexité des applications supportées par le protocole de gestion.
- de plus, si une demande est très grande du point de vue du nombre de blocs de données, l'agent retourne le plus grand nombre de données possible, au lieu d'envoyer un message d'erreur *tooBig*. Dans ce cas, la station n'a qu'à envoyer une autre demande GetBulkRequest pour les données dont la demande de lecture vient d'échouer.

4.3.Le PDU InformRequest

Ce PDU est échangé entre stations dans le but de partager des informations de gestion. Ce type de PDU est envoyé aux destinations spécifiées dans l'objet *SNAIPv2EventNotifyTable*, défini dans le MIB concernant les gestionnaires (manager-to-manager), ou bien aux destinations spécifiées par l'application.

Lorsqu'un message InformRequest est reçu, l'entité SNMPv2 réceptrice (station réceptrice) détermine en premier lieu la taille du message à retourner. Si elle dépasse une limite locale ou la taille maximale d'un message, alors un PDU est émis avec une valeur *tooBig dans* le champ error *status*, une valeur nulle dans le champ *error-index* et un champ nul dans *variable-bindings.* Par contre, si le PDU reçu n'est pas trop grand, alors la station réceptrice fait passer son contenu à l'application destination, puis génère un PDU contenant les mêmes valeurs *request-id et variable-bindings* que celles du message InformRequest, avec un champ error *status* égal à noError et une valeur nulle dans le champ *error-index*.

4.4.Le PDU Trap

La syntaxe et la sémantique ne sont pas les mêmes entre les deux versions de SNMP. Dans la version SNMPv2, le message Trap a le même format que les autres PDU, sauf GetBulkRequest. Ceci a pour avantage de faciliter le traitement d'un message par le récepteur du message (la station de gestion). De même qu'avec SNMPvl, aucun accusé de réception n'est retourné à l'agent suite à l'émission d'un Trap.

4.5.Nouveaux types de données

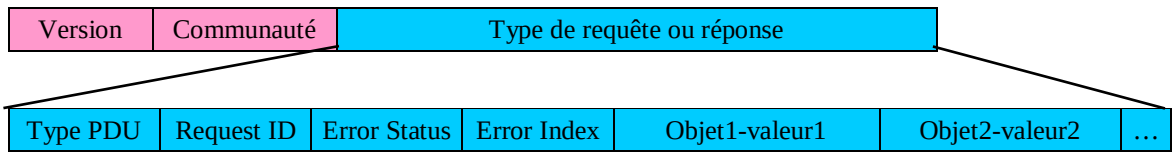
SNMP utilise des types de données pour définir des modules MIB. Avec la version SNMPv2, quelques types ont été éliminés, d'autres ajoutés et d'autres figurent dans les deux versions. Les types ajoutés concernent principalement le support des données représentées sur 32 bits *(Integer32, Unsigned32, Gauge32, Counter32),* sur 64 bits *(Counter64) et* le pseudo-type BITS.

4.6.Format des trames SNMPv2

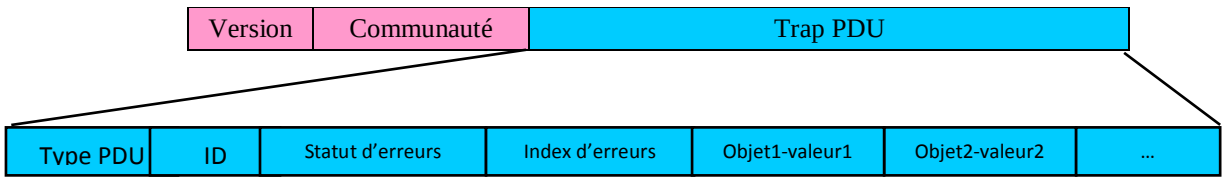
Les protocoles SNMPv1 et SNMPv2 utilisent le même format de message, la principale différence se situe dans le champ de la version. Le champ *version number* vaut 0 pour SNMPv1 et 1 pour SNMPv2.

Les primitives GetRequest, GetReponse, GetNextRequest et SetRequest utilisent la même structure de donnée.

PDU's GetRequest, GetNextRequest, SetRequest et GetResponse



PDU's Trap et Inform



PDU GetBulk

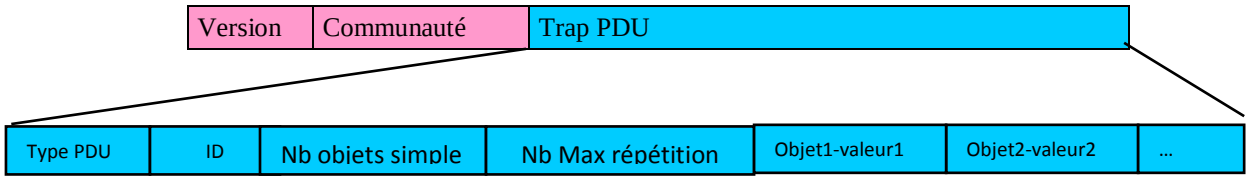


Figure 3.9 : formats des trames SNMPv2

4.7.**Inconvénients du SNMPv2**

Vu la persistance de l'insécurité l'IETF a chargé deux groupes d'expert pour améliorer et sécuriser la version 2 du SNMP, afin qu'elle soit normalisée. Après les travaux les deux groupes n'ont pas pu se mettre d'accord sur un consensus concernant le fonctionnement du mécanisme de sécurité. Pour cette raison, deux versions ont été proposées (SNMPv2u et SNMPv2*) qui malheureusement, d'après certains experts ne peuvent coexister, donc n'est pas une solution à long terme. Toutefois, les membres de la commission se sont entendus uniquement sur les améliorations de ce protocole et non sur la sécurité. L'IETF a terminé ses travaux en publiant une version de SNMPv2 (on l'appelle SNMPv2c, RFC 1901, RFC 1905 et RFC 1906) sans sécurité. Par la suite une version intermédiaire apparaîtra connu sous le nom de SNMPng (next generation SNMP).

5. SNMPv3 (version 3)

L'amélioration la plus importante apportée par SNMPv3 concerne la sécurité : authentification, et control d'accès. La sécurité a été la plus grande faiblesse du SNMP depuis sa création, l'authentification dans les versions antérieures de SNMP est réalisée grâce à un mot de passe (corde de la communauté) qui est transmis en clair entre un manager et un agent ce système ne fournit aucune sécurité. Ce mot de passe peut être intercepté et être utilisé pour accéder aux ressources du réseau.

Le protocole SNMPv3 apporte une amélioration du point de vue sécurité par rapport aux versions précédent. SNMPv3 soutient toutes les opérations définies par la Versions 1 et 2, il n'y a pas de nouvelles applications (requêtes) créées, la seule nouveauté concerne la gestion de la sécurité.

La version 3 de SNMP est constituée de trois modules :

- **Message Processing and Control**, qui définit la création et la fonction d'analyse des messages
- **Local Processing**, qui s'occupe des contrôles d'accès et l'exécution de données.
- **Security**, qui permet l'authentification, le chiffrement ainsi que la prise en compte des contraintes de temps dans certains messages SNMP.

En plus de l'amélioration de la sécurité, la notion des managers et agents est abandonnée au profit d'une nouvelle terminologie et dans laquelle les managers et les agents sont appelés entités *de SNMP* qui peuvent être configurés aussi bien comme agent que comme manager. Il est important de noter que le concept d'entité est important parce qu'il définit une architecture, plutôt qu'un simple ensemble de messages. L'architecture de SNMPv3 est proche du modèle client-serveur classique, c'est-à-dire qu'on peut avoir plusieurs clients et plusieurs serveurs. Chaque entité se compose d'un moteur de SNMP et d'une ou plusieurs applications de SNMP.

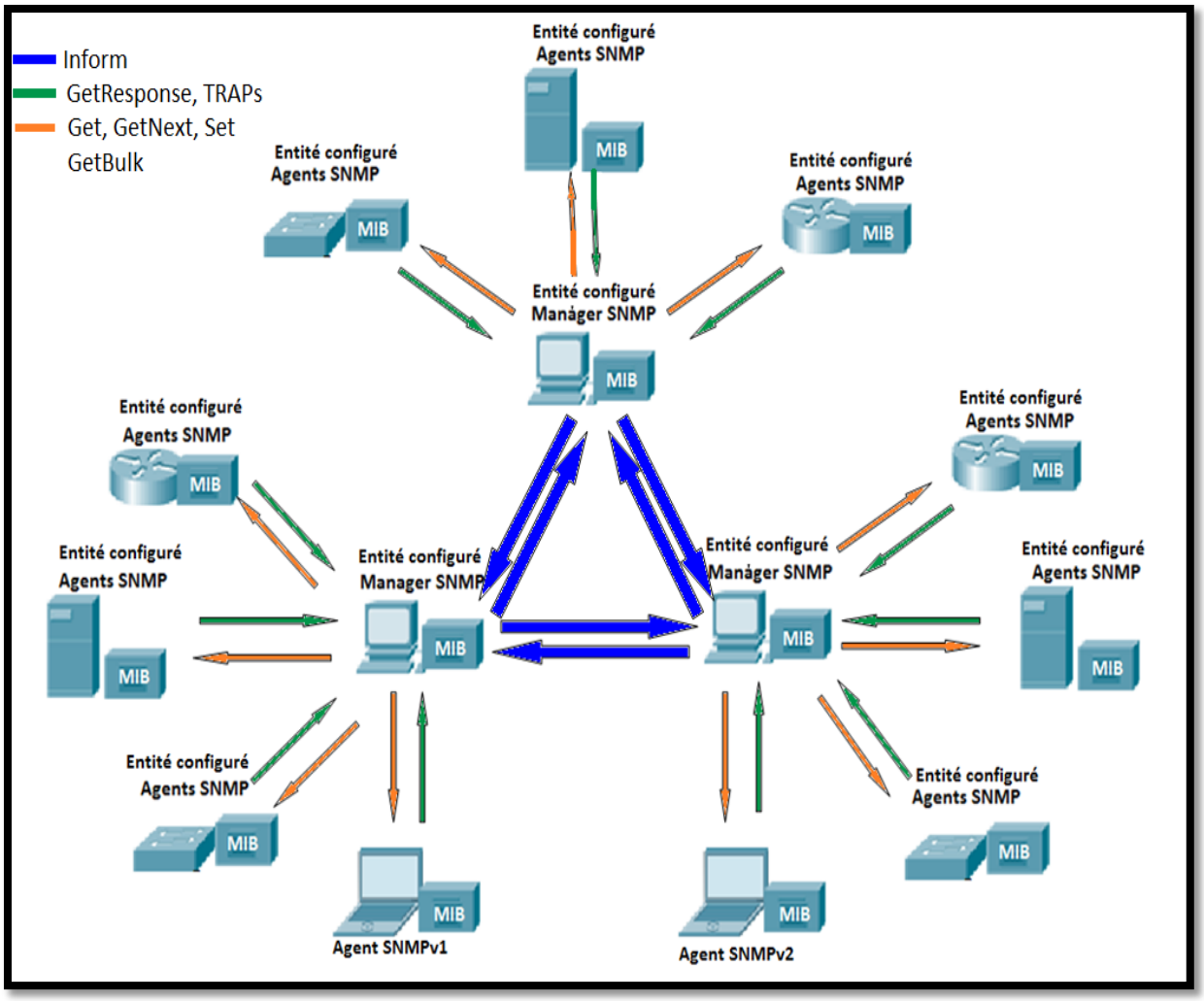


Figure 3.10 : architecture manager/agent SNMPv3

5.1.Le moteur SNMPv3

Le moteur se compose de quatre parties: l'expéditeur, le sous-ensemble de traitement de message, le sous-ensemble de sécurité, et le sous-ensemble de contrôle d'accès.

- **l'expéditeur**: Il doit envoyer et recevoir des messages, il essaye de déterminer la version de chaque message SNMP reçu version 1, 2 ou 3. Si cette version est reconnue, il diffuse le message SNMP vers le sous-ensemble de traitement de message et à d'autres entités SNMP.
- **Le sous-ensemble de traitement de message:** Il prépare les messages pour leur envoie et extrait des données à partir des messages reçus. Un système de traitement de message peut contenir des modules de traitement de message multiple. Par exemple, un sous-ensemble peut avoir des modules pour les demandes SNMPv1, SNMPv2, et

SNMPv3 de traitement. Il peut également contenir un module pour d'autres modèles de traitement qui doivent être définis encore.

- **Le sous-ensemble de sécurité**: Il fournit des services d'authentification et de confidentialité et chaque service dispose de son mot de passe. L'authentification a pour rôle d'assurer que le paquet reste inchangé pendant la transmission, et que le mot de passe est valide pour l'usager qui fait la requête. Pour authentifier l'information qui va être transmise, on doit aussi avoir un mot de passe qui est « partagé ». Le mot de passe ne doit donc être connu que par les deux entités qui s'envoient les messages.
  Les étapes d'authentification sont les suivantes :

  Au niveau de l'entité agent

  - Le transmetteur groupe des informations à transmettre avec le mot de passe.
  - On passe ensuite ce groupe dans la fonction de hachage à une direction (SHA-1 ou MD5 qui sont des algorithmes de cryptage).
  - Les données et le code de hachage sont ensuite transmis sur le réseau.

  Ensuite au niveau du manager

  - Le receveur prend le bloc des données, et y ajoute le mot de passe.
  - On passe ce groupe dans la fonction de hachage à une direction.
  - Si le code de hachage est identique à celui transmis, le transmetteur est authentifié.

  Avec cette technique, le mot de passe est validé sans qu'il ait été transmis sur le réseau. Le service de confidentialité a pour but d'empêcher que quelqu'un n'obtienne les informations de gestion en écoutant sur le réseau les requêtes et les réponses de quelqu'un d'autre, lequel est autorisé à obtenir ces informations. Il emploie l'algorithme de DES pour chiffrer et déchiffrer des messages de SNMP, le principe est le même que pour l'authentification, à part que l'authentification est appliquée à tout le paquet tandis que la confidentialité est seulement appliquée sur le « Scoped-PDU ».

  Actuellement, le DES est le seul algorithme utilisé, bien que d'autres puissent être ajoutés à l'avenir.

- **Le sous-ensemble de contrôle d'accès**: Il est responsable de commander l'accès aux objets de la MIB. On peut définir à quels objets un utilisateur peut accéder, et les opérations qui lui sont permises sur ces objets. Par exemple, vous pourriez vouloir

limiter l'accès lecture/écriture d'un utilisateur à certaines parties *de l'arbre* mib-2, tout en permettant l'accès inaltérable à l'arbre entier.

5.2.Les applications SNMPv3

Les applications sont des processus qui interagissent avec le moteur SNMP en utilisant des messages qui peuvent être définis dans le protocole, ou des messages décrits par une mise en œuvre spécifique du moteur.

Les applications sont développées pour effectuer des opérations de gestion spécifiques dont l'objectif peut varier d'une application à une autre. Toutefois, toutes les applications utilisent en commun le même moteur SNMP pour effectuer les opérations de gestion.

Il existe un certain nombre d'applications, parmi lesquelles nous citons :

- le générateur de commande,
- le répondeur de commande,
- le créateur d'avis,
- le récepteur d'avis,
- l'expéditeur de procuration.

5.3.Structure d'une entité SNMPv3

Jusqu'ici nous avons parlé de l'entité SNMPv3 en termes de définitions abstraites. La figure suivante montre la structure interne d'une entité SNMPv3.

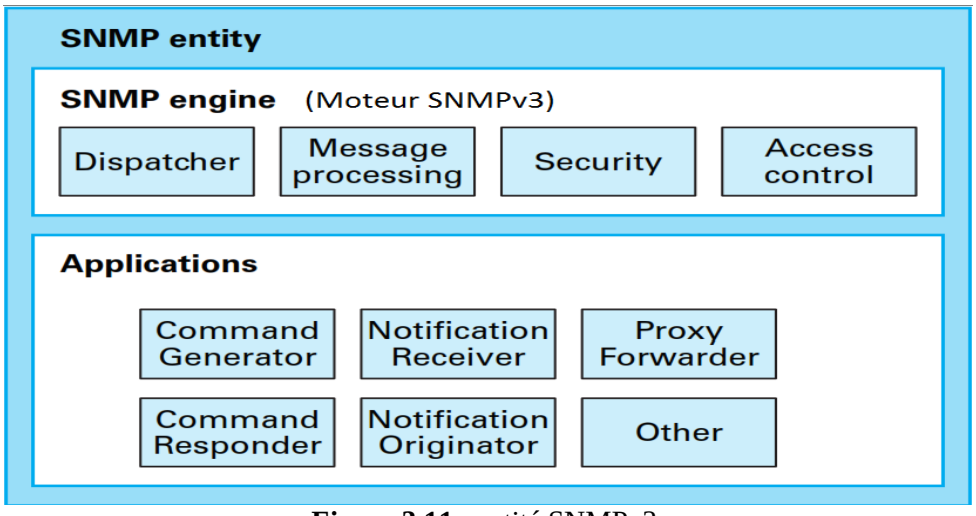


Figure 3.11 : entité SNMPv3

5.4. Format de message SNMPv3

Le cadre SNMPv3 adopte beaucoup de composants qui ont été créés dans SNMPv2, y compris les opérations du protocole SNMPv2, des types de PDU et le format de PDU. Parmi les changements cruciaux faits dans SNMPv3, il inclue une manière plus flexible de définir des méthodes et des paramètres de sécurité, pour permettre la coexistence des techniques multiples de sécurité.

Le format général de message pour SNMPv3 suit toujours la même idée d'un message global qui contient un en-tête et une PDU encapsulée. Cependant, dans la version 3 ce concept est encore raffiné. Les champs de l'en-tête ont été divisés en ceux traitant la sécurité et ceux qui ne traitent pas des sujets de sécurité. Les champs de "non-sécurisé" sont communs à toutes les réalisations SNMPv3, alors que l'utilisation des champs de sécurité peut être travaillée par chaque modèle de la sécurité SNMPv3, et être traitée par le module dans une entité de SNMP qui traite la sécurité. Cette solution fournit une flexibilité considérable tout en évitant les problèmes qui ont infesté SNMPv2.

Le format global du message SNMPv3 est décrit dans la RFC 3412, qui décrit le traitement et l'expédition de message de la version 3.

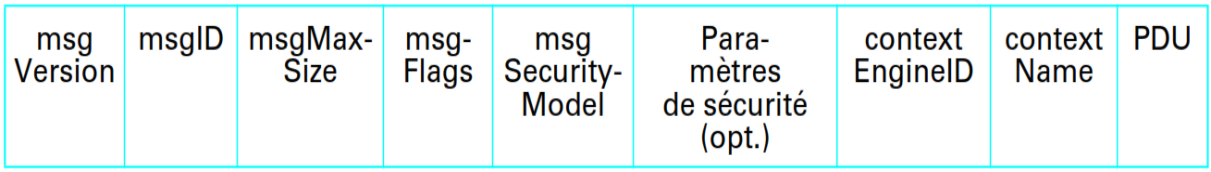


Figure 3.12 format de message SNMPv3

- **msgVersion:** le numéro de version, comme en SNMPv1 et v2, permet au *dispatcher* de savoir à quel sous-système du message *processing* envoyer le message ;
- **msglD:** est l'identifiant du message SNMP qui permet d'associer les requêtes et réponses ;
- **msgMaxSize:** indique la taille maximum de message supportée par l'entité SNMP ;
- **msgFlags:** drapeaux indiquant si une réponse est attendue et si un modèle de sécurité a été utilise.
- **msgSecurityModel:** définit le modèle de sécurité utilisé, ce qui déterminera la signification du bloc suivant (paramètres de sécurité) et déterminera aussi si la suite du message sera cryptée ;

- **Paramètres de sécurité:** C'est le progrès principal de SNMPv3 : SNMPv1 et SNMPv2C offrent un modèle de sécurité réseau à base de noms de communauté, qui n'assure que l'authentification. Il est de plus très facile de le détourner puisque les noms de communauté circulent en clair sur le réseau. Ce modèle reste une option dans SNMPv3.
- **contextEngineID et contextName:** permettent de déterminer le contexte de la requête : en SNMPv3, à l'OID de la MIB qui identifie un objet, se rajoute la notion de contexte qui permet d'avoir plusieurs instances de MIB sur une entité SNMP, qui seront identifiées par un contexte. Par exemple, pour un équipement implémentant deux ponts, le contextName permettra d'identifier la MIB du pont d'où on désire interroger.
- **PDU:** contient les valeurs demandées ou les réponses à des demandes.

6. Les implémentations existantes du protocole SNMP

Il existe des centaines d'implémentations différentes du protocole SNMP, car il s'agit d'un protocole parfaitement bien défini et qu'il est de plus en plus exploité au sein des réseaux. Chaque implémentation a ses propres avantages et inconvénients. Certaines ont pour but de fournir des applications de gestion SNMP, d'autres ont pour but de fournir des bibliothèques de fonctions (API) pour la gestion SNMP. Certaines fournissent les deux, comme la distribution *net-snmp* (www.net-snmp.org) du domaine libre. Celle-ci propose les applications de base pour débuter et utiliser efficacement SNMP.

On retrouve dans la plupart des distributions le même ensemble d'applications de base pour la gestion du matériel via SNMP. Il s'agit des applications suivantes :

- Snmpget : Permet de lire une variable d'un agent SNMP
- Snmpset : Permet de définir la valeur d'une variable d'un agent SNMP
- Snmpwalk : Permet de parcourir une liste de variables d'un agent SNMP
- Snmptrap : Envoie une trap à un manageur
- Snmpbulkget, Snmpbulkwalk : Identique à Snmpget et Snmpwalk mais en utilisant des requêtes de type Snmpbulk.
- Snmpinform : Envoie une requête Inform à un manageur

Ces applications sont généralement basées sur la même architecture de programmation. Certains programmes contiennent directement les applications, ou l'implémentation du protocole, de manière à accélérer la vitesse de traitement des informations.

Par ailleurs, la plupart des distributions Unix ainsi que les distributions Microsoft® Windows Server fournissent un agent SNMP pour contrôler à distance la station et obtenir des informations sur celles-ci. Enfin, la plupart des matériels réseaux administrables d'aujourd'hui embarquent dans leur système d'exploitation un agent SNMP pour gérer le matériel à distance.

7. Avantages et inconvénients de SNMP

Nous l'avons vu, le protocole SNMP a de nombreux avantages en tant qu'outil de gestion réseau :

- Accès centralisé : la gestion réseau s'effectue depuis une machine centrale sans soucis, et c'est même préférable pour la sécurité.
- Sécurité : la sécurité s'est accrue au cours des différentes versions, jusqu'à respecter la plupart des contraintes imposées.
- Fiabilité : le protocole utilisé permet de s'assurer que les requêtes sont bien arrivées à destination et qu'elles ont été correctement interprétées.
- Evolutivité : l'utilisation d'une arborescence pour la gestion des variables permet d'avoir une évolution continuelle des capacités fonctionnelles accessibles via ce protocole.
- Gestion de la diversité : l'utilisation d'une interface standard à tous les matériels permet de contrôler de la même manière tous les équipements réseaux, ce qui a des avantages indéniables lorsque l'on dispose d'un parc informatique très diversifié.

Toutefois, certains reproches peuvent être faits à SNMP : l'interface standard de communication est très pauvre et ne fournit qu'un nombre très limité d'informations : état des interfaces réseaux, nombre d'octets transmis, etc. Mais tous les constructeurs ont décidé d'exploiter leurs spécificités directement dans leur MIB propre plutôt que d'essayer d'uniformiser au maximum et de faire évoluer la MIB standard. Ainsi, même si certaines informations peuvent être obtenues identiquement sur des matériels distincts, il sera parfois nécessaire de rechercher dans la MIB du constructeur pour obtenir des informations plus pointues.

CONCLUSION

Une administration de réseau efficace demande un travail non négligeable en terme de choix et de mise en place d'outils d'organisation.

SNMP est un protocole d'administration de réseau très répandu; car il fournit des mécanismes simples de fonctionnement. Dans sa première version il est composé de trois groupes d'opérations de gestion: Get, Set et Trap. Get permet d'extraire la valeur d'un objet. Deux opérations sont fournies dans ce but: GetRequest et GetNextRequest. Le groupe d'opérations Set est employé pour modifier la valeur des objets. Ces deux groupes d'opérations (Get et Set) sont toujours employés par la station de gestion. Cependant, un agent ne peut invoquer que l'opération Trap. Celle-ci lui permet d'informer une station de gestion qu'un événement s'est produit.

En plus des opérations GetRequest, GetNextRequest, Set Request et Trap, SNMPv2 fournit de nouveaux services pour résoudre les problèmes et les limitations posés par SNMPvl. D'abord, deux nouveaux messages ont été ajoutés: GetBulk, pour la lecture de grandes quantités d'informations, et Inform, qui permet la communication entre stations de gestion.

L'amélioration la plus importante apportée par SNMPv3 concerne la sécurité : authentification, et control d'accès.

Au fil du temps SNMP est devenu un standard incontournable dans le domaine de l'administration de réseaux d'entreprise. Son utilisation s'est étendue au delà, dans le monde du système et des applications. Les limitations de la version 1 et 2 sont comblées avec la version 3. Ce dernier standard demande toutefois plus de travail d'implémentation et de mise en œuvre, et il reste à voir si le marché va l'adopter ou conserver les solutions plus simples de SNMPv1 et SNMPv2C.

CHAPITRE IV:

CONCEPTION

Introduction

Le protocole SNMP a beaucoup d'avantages indéniables que nous avons pu mettre en avant, et les implémentations de celui-ci sont de plus en plus solides et fournissent des bases de plus en plus intéressantes aux développeurs de systèmes.

Notre travail consiste à concevoir une plate forme de surveillance réseau basé sur le protocole SNMP constitué de trois parts : le manager, le client et la MIB. On ferra aussi la mise en place de deux segments réseaux à l'aide d'un router Cisco.

1. Le cahier de charges

Définition :

Le cahier des charges trace les objectifs à atteindre et définit les différentes fonctions et tâches qui seront prêtes à être exécutées lors de la fin de la réalisation de l'application.

Pour notre projet, l'application (développé en Java NetBeans) sera représentée sous forme d'une fenêtre exécutant plusieurs fonctions, et ces derrières feront appel à plusieurs fonctionnalités et librairies de java. En dehors de ça, on ferra aussi la mise en place d'une topologie ayant deux segments réseau et faire communiquer tous les hôtes de la topologie avec chaque machine du réseau.

En résumé, la plate forme à réaliser doit permettre les fonctions suivantes :

- Questionner la MIB des agents se trouvant dans chaque segment réseau.
- Pouvoir récupérer un nombre d'informations sur chaque machine avec le protocole SNMP, et éventuellement les modifier si la donnée est modifiable.
- Un service d'alarme qui indique un disfonctionnement sur une machine du réseau.

La topologie avec le routeur va permettre les fonctions suivantes :

- Configurer l'agent SNMP dans le routeur
- Connecter les hôtes au routeur à travers des Switch à l'aide des câbles RJ45
- Faire communiquer tous les éléments de la structure réseau implémenté
- A chaque changement dans la topologie, signaler le Manager-SNMP

2. Les Architectures :

La figure suivante donne un aperçu global de l'ossature de l'application à développer.

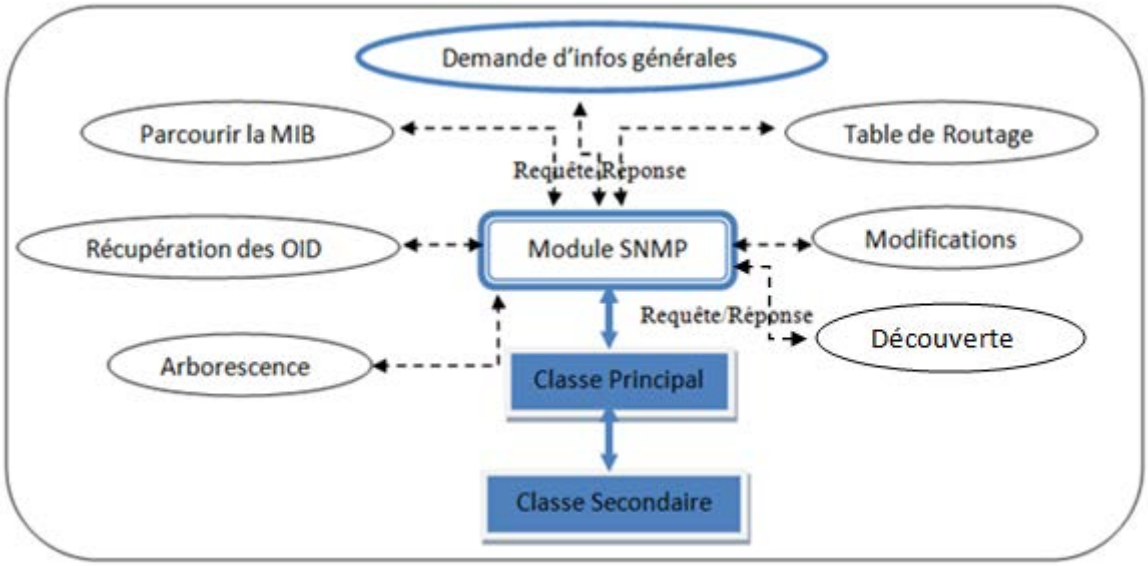


Fig1. Architecture de l'application

2.1.Etude de l'environnement de travail :

L'application est fondée sur deux axes :

- Une interface graphique, qui est la classe principale composé de plusieurs modules d'interrogation SNMP et de configurations.
- Une deuxième interface, la classe secondaire qui s'occupe d'assurer la surveillance du réseau.

2.1.1. La classe principale

La classe principale représente l'interface graphique de l'application, elle permettra de consulter les différents OID de la MIB, et de récupérer leurs valeurs, et cela grâce au module d'interrogation SNMP.

Elle permettra d'avoir une vision globale du réseau et de chacun de ses éléments qui le compose, c'est-à-dire, des machines qui ont l'agent SNMP actif.

2.1.1.1.Module d'interrogation SNMP :

Cette partie de l'application a pour rôle d'interroger une machine, et de récupérer les données de gestion nécessaire à notre application et les transmettre à la classe principale et de récupérer les données de base de chaque éléments.

Même si avec le module d'interrogation de notre application on sera en mesure d'avoir toutes les informations de la MIB de l'agent, ce n'est pas pour autant qu'on sera en mesure de les exploiter, d'une parte, parce qu'on ne connait pas toujours la signification de certaines valeurs retourné et d'autre part, parce que les agents ne retournent pas toujours la même valeur correspondant au même OID.

2.1.1.2.Demande d'informations générales

La demande d'informations générales (***Get Agent SNMP info***) est une fonction qui doit nous permettre d'obtenir les informations de base d'un agent tels que, le type et la version de l'OS, le nom de l'agent et le nom de l'ordinateur sur le réseau ainsi que son emplacement et son adresse MAC.

2.1.1.3.Découverte

La découverte (***DiscoverSNMP***) est une fonction qui va nous retourner tous les hôtes avec le protocole SNMP actif, appartenant au réseau en question.

2.1.1.4.Table de routage

Cette fonctionnalité doit nous fournir la liste des adresses IP contenus dans la machine, c'est-à-dire, l'adresse de bouclage de la machine et son adresse IP respective, en plus de ça, si l'hôte contient aussi des machines virtuelles installées, elle doit aussi retourner les adresses IP des VMware network configurés.

2.1.1.5.Modifications

Dans la configuration d'un agent, il faut que la communauté utilisé soit en lecture/écriture pour qu'on puisse effectuer des modifications telles que :

- ✓ *nom de l'agent*
- ✓ *l'emplacement de l'agent*
- ✓ *nom de l'ordinateur*

sur l'ensemble des éléments qui composent le réseau.

2.1.1.6.Parcourir la MIB

Le parcours de la MIB (**Get all OID values)** va nous permettre de trouver toutes les valeurs correspondant à chaque OID de l'agent questionné.

2.1.1.7.Récupération des OID

Cette partie va implémenter les méthodes de base du protocole SNMP.

i) La méthode GetOID :

Cette méthode va récupérer la valeur d'une variable de la MIB, quelque soit la machine du réseau. Il suffit d'entrer l'adresse IP de la machine, de vérifier la valeur de la « Communauté », ainsi que de donner la valeur de l'identifiant de la variable (OID) dont on veut récupérer la valeur.

ii) La méthode SetOID :

Cette méthode modifie la valeur d'une variable de la MIB. Comme pour GetOID, il suffit de donner l'adresse IP, la « Communauté » et l'OID pour utiliser cette méthode. Mais contrairement à la lecture, l'écriture n'est pas toujours permise, certaines variable de la MIB sont en Lecture-Seul, donc impossible à modifier. Par exemple l'OID « 1.3.6.1.2.1.1.1.0 » qui représentent le type de hardware et software de la machine, est en Lecture-Seul.

L'application SNMP dispose également de la fonction Walk. Cette fonctionnalité a pour but de donner toutes les valeurs de tous les objets de la MIB d'une machine. Elle est développée en utilisant la fonctionnalité GetNext (avoir la prochaine), en commençant par avoir la première valeur du premier objet de la MIB, en suite on avance pas à pas jusqu'à atteindre la fin des objets. Pour ce qui est de la valeur de début si l'utilisateur ne donne pas de valeur, l'application va automatiquement chercher la première valeur de la MIB.

2.1.1.8.Arborescence

L'arborescence (**Get table**) nous donne les informations générales d'une branche complète, c'est-à-dire, si par exemple on veut afficher tous les éléments de la MIB standard (1.3.6.1.2.1.), cette fonction nous retourne tous les objets pouvant être accédés à

partir de cette branche et ainsi de suite. Il faut qu'on sélectionne d'abord sur le champ OID, l'OID qu'on veut exploiter.

2.1.2. La Classe Secondaire

La deuxième interface (récepteur de traps), permet de récupérer les traps envoyé par les machines du réseau. Lorsqu'un élément du réseau entre dans un état anormal, l'agent SNMP de cet élément prévient notre application par le biais d'une trap SNMP. Il faut noter que l'agent SNMP se trouvant sur la machine qui envoi les traps, a besoin d'une petite configuration, entre autre l'adresse IP où sera envoyer les traps générés.

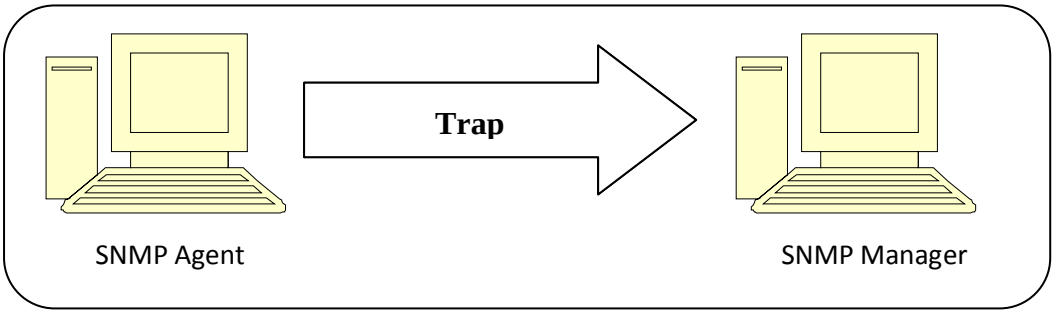


Fig2.Envoi d'une trap

Bien que le manager reçoit les traps, il n'envoi aucun accusé de réception. L'agent ne sait jamais si la trap envoyée a atteint sa destination.

2.2.L'architecture de la topologie

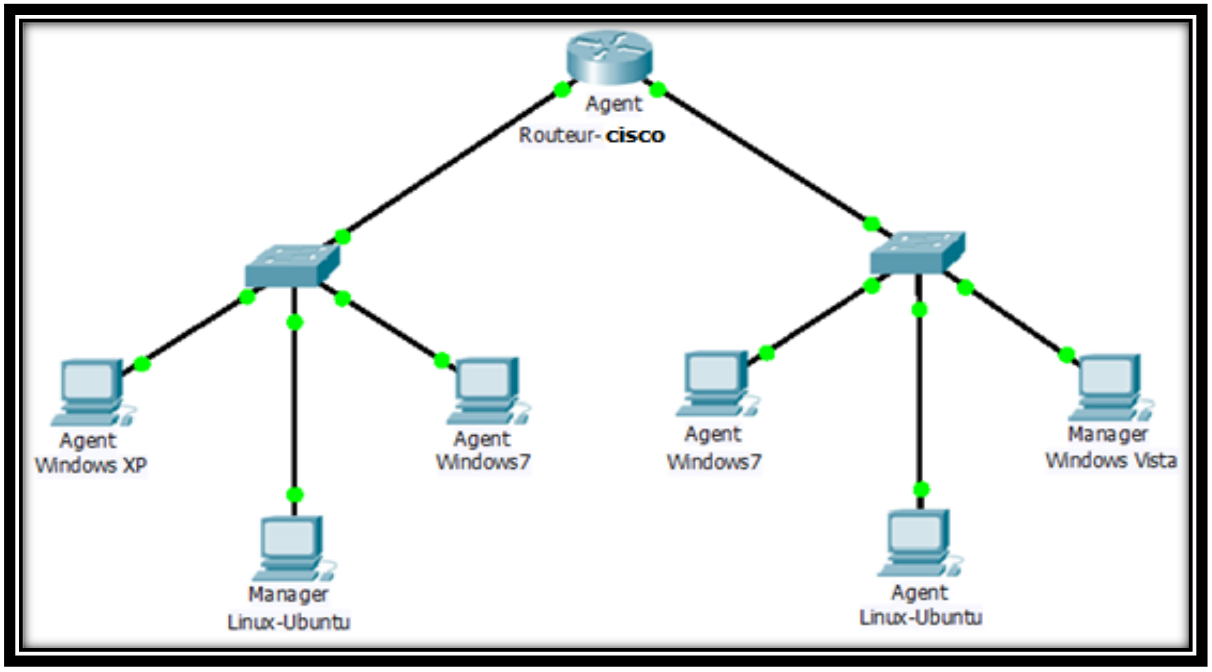


Fig3. Architecture de la topologie

3. Vue d’ensemble

Voila une vue d’ensemble des taches à effectuer :

- ✓ Configure le routeur en agent
- ✓ Configurer les machines
- ✓ Connecter ces machines au routeur
- ✓ Faire communiquer l’ensemble des machines des segments à l’aide du routeur
- ✓ Faire une application Manager SNMP en java
- ✓ Obtenir des informations de la MIB de chaque agent sur le réseau grâce à l’application
- ✓ Reconfigurer les agents SNMP grâce à l’application Manager SNMP
- ✓ Surveille chacun des agents grâce à l’application Manager SNMP

CONCLUSION

Dans ce chapitre on a vu le modèle conceptuel de notre application, modèle nécessaire pour entamer la réalisation.

Dans le prochain chapitre, on passe à la réalisation de l’application dans lequel on présentera les choix effectué afin de réaliser ce qui a été décrit dans ce chapitre.

Chapitre V:

Réalisation

et

Implémentation

Introduction

Ce chapitre est dédié a l'implémentation et la réalisation de notre travail. Comme nous l'avons déjà dit au niveau du chapitre précédent, ce travail consistait à réaliser une application de surveillance réseau exploitant la MIB basée sur le protocole SNMP en utilisant le langage Java, mais aussi la mise en place de deux segments réseaux via un Routeur Cisco.

Dans ce chapitre nous allons présenter et décrire l'environnement de développement ainsi que les programmes et les outils utilisés pour mener à terme notre application.

1. Les programmes

Parmi les programmes qui nous avons utilisés les principaux sont; le GNS3 et le VMware Workstation.

1.1.Le GNS3

Le GNS3 est un logiciel open source qui permet d'émuler des routeurs Cisco, des firewall PIX, des modules Switch, de la même façon que Vmware permet d'émuler des OS comme Windows, Linux ou Solares. Avec lui, on peut construire notre propre architecture réseau comme s'elle était réel, et simuler des architectures complexe (ou simple), puisqu'il permet de connecter dans notre réseau virtuel **de "vrais" hôtes et des hôtes virtuels** créés sous **VirtualBox ou VMware Workstation.**

Contrairement à d'autres simulateurs comme Boson ou PacketTracer, GNS3 émule un IOS Cisco que l'on fournit, et cet IOS se comporte exactement comme s'il tournait sous une plateforme matérielle Cisco. Avec boson ou PacketTracer, ce sont des développements de logiciels qui prennent quelque commande Cisco et qui répondent en conséquence de ce que le programmeur a choisit, on a un fonctionnement approximatif, et on ne pourra pas tester par exemple les nouvelles fonctionnalités d'un IOS.

a) Installation et configuration

GNS3 est téléchargeable depuis le site de http://www.gns3.net/download. Une fois en possession de l'exécutable, on fait deux clique gauche, puis un clique sur suivant jusqu'à options d'installation et on sélectionne toutes les options comme illustré dans figure suivant.

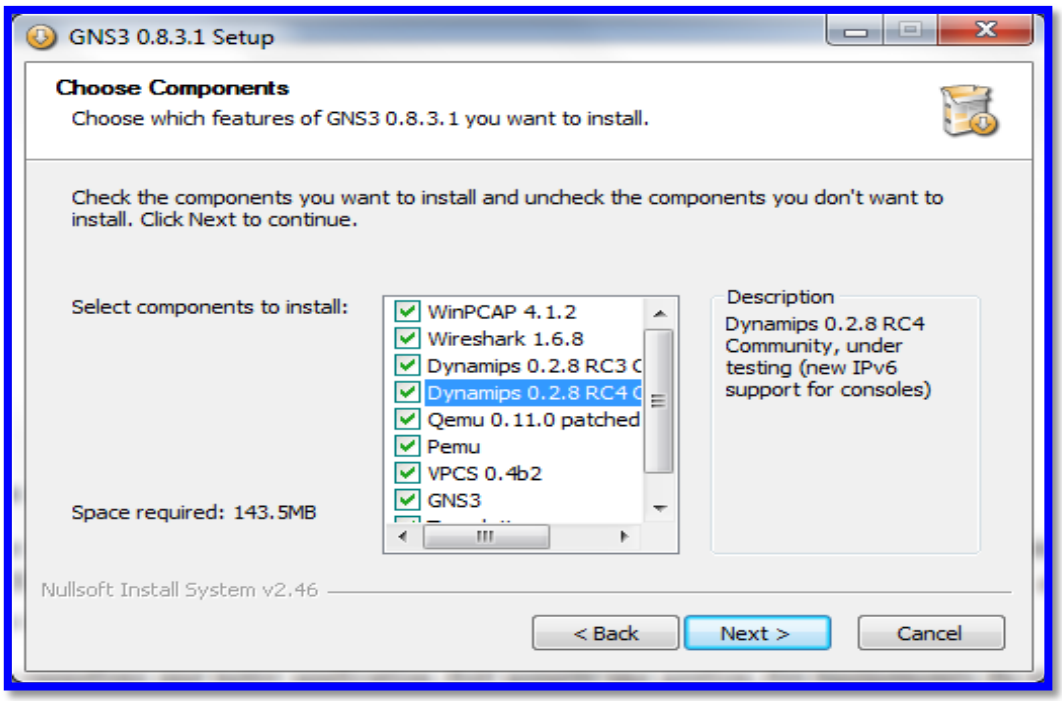


Figure 5.1: Installation de GNS3

Lorsque la fenêtre de d'installation de Wincap s'affiche, on clique aussi sur "suivant" jusqu'à la fin. Une fois l'installation terminée, on lance gns3 puis on sélectionne l'option 1 parmi celles affichées. Puis on clique sur général dans le coin gauche de la fenêtre affichée, ensuite du côte droite de la même fenêtre, on change le répertoire par défaut de destination des projets et celui des images IOS, comme nous montre la figure suivante.

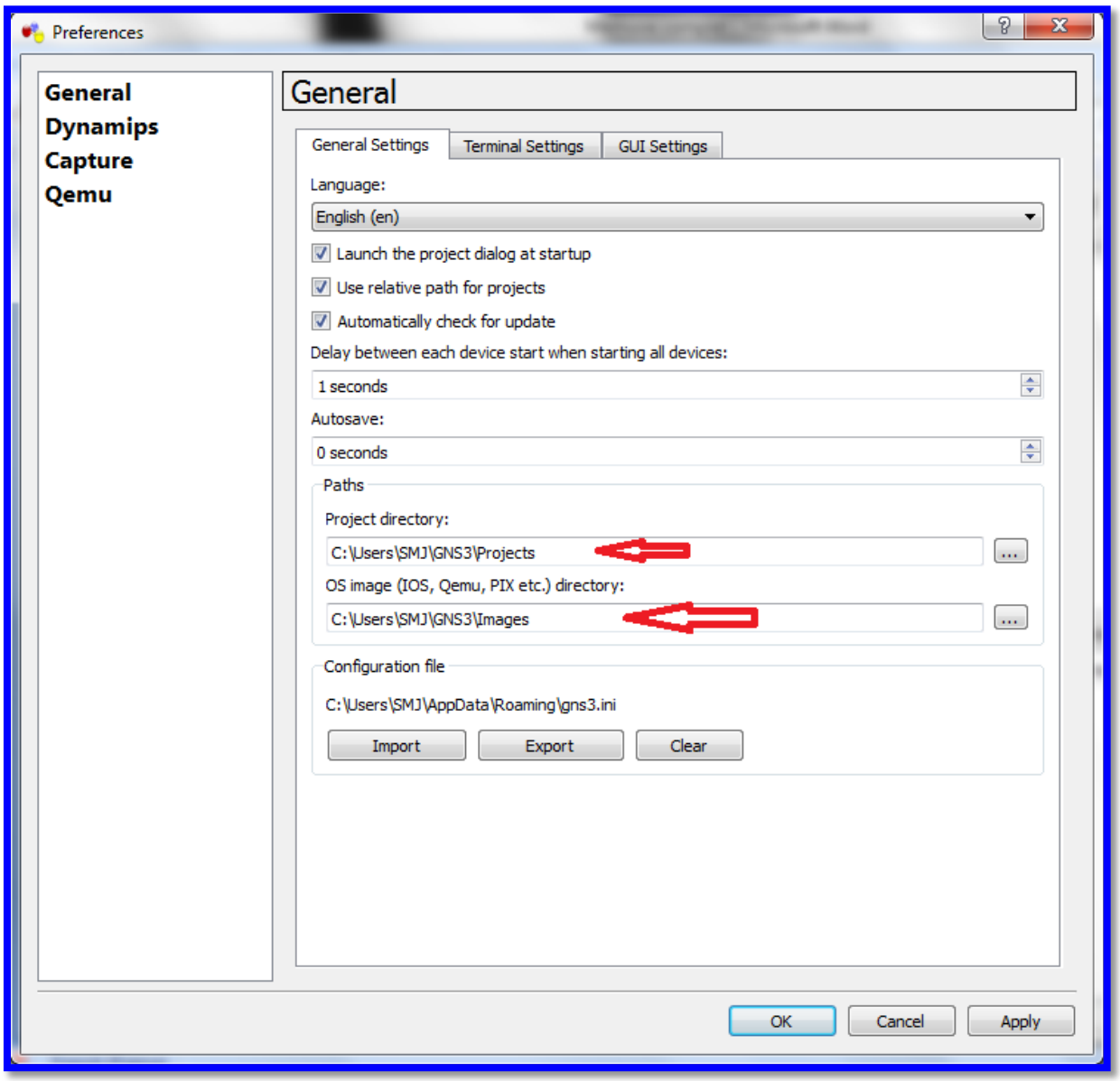


Figure 5.2: Configuration de GNS3

Ensuite on clique sur Dynamic situé au coin gauche de la fenêtre, puis sur "Test settings", il faut attendre quelques seconds et le message "successfully started" s'affichera comme illustré dans la figure suivante.

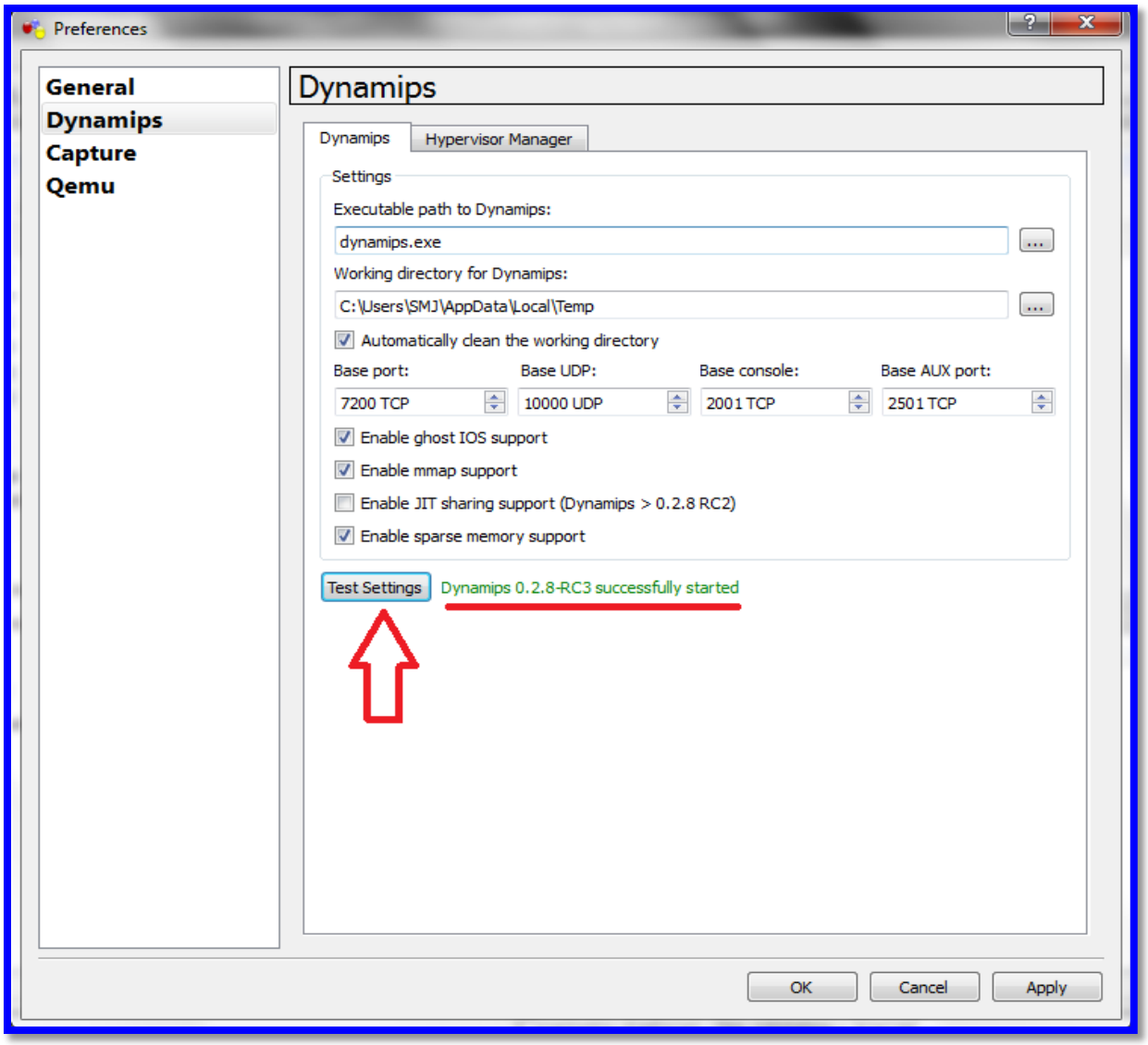


Figure 5.3: Configuration de GNS3

Si ce message ne s'affiche pas alors on doit recommencer la procédure de configuration ou d'installation.

1.2. Vmware Workstation

VMware, est une société informatique américaine fondée en 1998, filiale d'EMC Corporation depuis 2004, qui propose plusieurs produits propriétaires liés à la virtualisation d'architectures x86. C'est aussi par extension le nom d'une gamme de logiciels de virtualisation.

Le Vmware Workstation est la version station de travail du logiciel Vmware. Il permet la création d'une ou plusieurs machines virtuelles au sein d'un même système d'exploitation (généralement Windows ou Linux), ceux-ci pouvant être reliés au réseau local avec une adresse IP différente, tout en étant sur la même machine physique (machine existant

réellement). Il est possible de faire fonctionner plusieurs machines virtuelles en même temps, la limite correspondant aux performances de l'ordinateur hôte.

2. L'environnement de développement

2.1. L'environnement d'exécution

Notre application peut s'exécuter sur tous les systèmes qui peuvent intégrer une JVM. Comme le JVM est multiplateforme, notre applications est par ce fait portable.

2.2. L'environnement de programmation

Comme notre application manipule des modules du protocole TCP/IP pour les requêtes SNMP, notre choix s'est porté sur l'outil de développement NetBeans IDE version 6.7.1 de Sun Microsystem comme environnement de programmation, qui offre une souplesse de programmation, une convivialité de travail, et également la possibilité d'intégrer des plugins.

Nous avons utilisé la version 6.23 de La JDK (Java Developpement Kit).

2.3. SNMPCommunicationInterface

SNMPCommunicationInterface est une API gratuit, qui gère le protocole SNMP. C'est une API très populaire dans le monde du SNMP vu que c'est une API libre. Elle peut être employée pour développer des applications de gestion de réseau ; pour construire des applets, des composants, et des applications réparties comme DCOM, EJB, CORBA, ou RMI. La bibliothèque fournit les fonctions et les composants le plus généralement utilisés pour rendre le développement plus simple.

2.4. SNMP EXPLORER :

Le logiciel libre SNMP Explorer permet d'interroger n'importe quel périphérique compatible SNMP en utilisant le SNMPv1 ou SNMPv2. Comme une application Java peut fonctionner sur tout système ayant le Run Time Java installé. Il peut effectuer des GET, GETNEXT, WALK et des SET. Il permet aussi de regarder la hiérarchie des variables de la MIB sous forme d'arbre et fournit des informations sur chaque nœud. Nous l'avons utilisé dans le cadre des comparaisons avec notre application java.

3. Installation du service SNMP

3.1. Activation, Installation et configuration du service SNMP sous Windows

Avant de pouvoir utiliser notre application, il faut tout d'abord activer et configuré l'agent SNMP de la machine que vous souhaitez gérer. Voici les étapes à suivre pour configurer l'agent sous Windows :

3.1.1. Activation et installation du service SNMP

- **Ouvrez le Panneau de configuration**, double-cliquez sur Ajout/Suppression de programmes, puis cliquez sur Ajouter/supprimer des composants Windows.

- Si vous êtes sous Windows xp dans Composants, cliquez sur Outils de gestion et d'analyse sans activer ni désactiver la case à cocher correspondante), puis cliquez sur détails et activez la case à cocher SNMP (Protocole simplifié de gestion de réseau), puis cliquez sur OK.

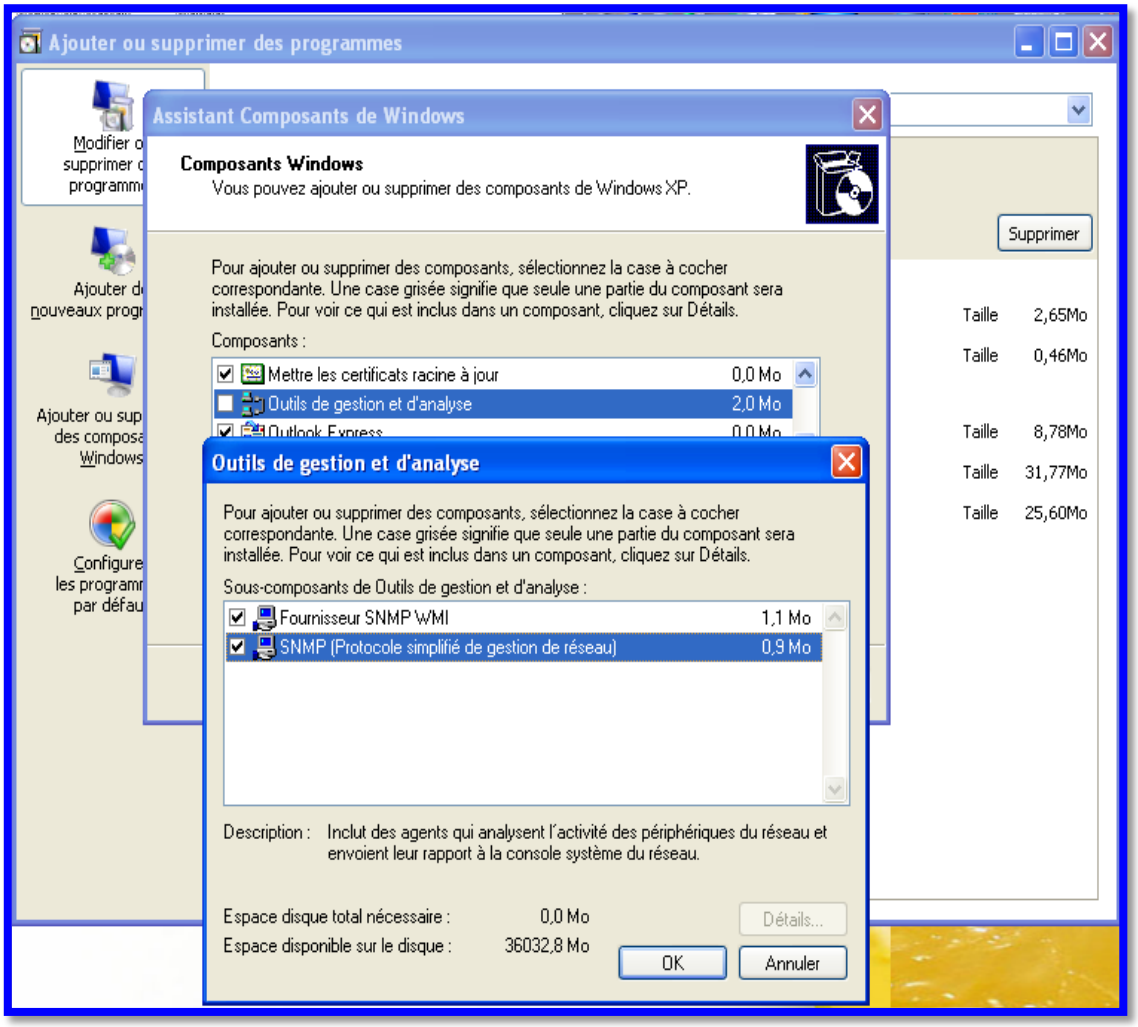


Figure 5.4: Activation et installation de SNMP sous Windows

- Si vous êtes sous Windows Vista ou Seven cliquez sur "Activer ou désactiver des fonctionnalités Windows, puis chercher dans la liste "Protocole SNMP" et cochez la case comme montré dans la figure suivante.

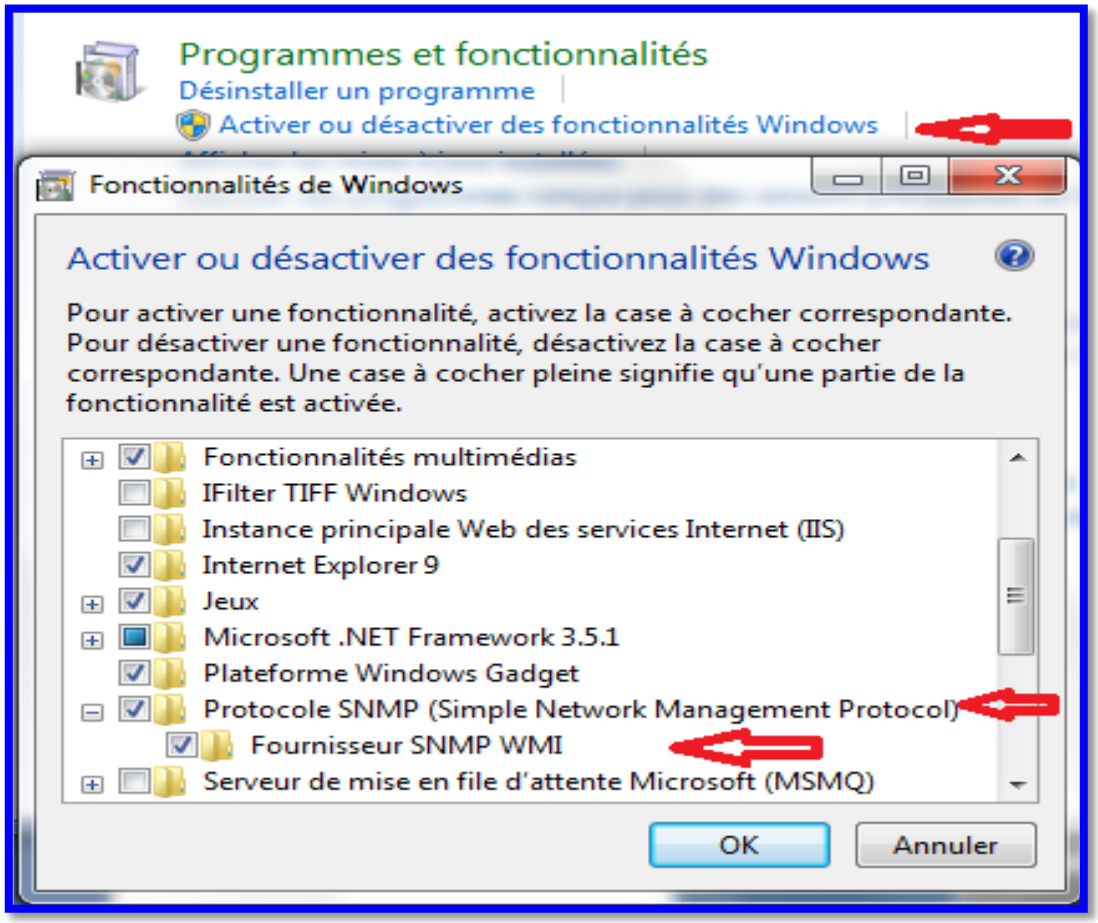


Figure 5.5: Installation et activation de SNMP sous Windows

- Ensuite cliquez sur Suivant et l'installation du service SNMP démarre automatiquement (elle peut prendre plusieurs minutes).

3.1.2. Configuration des propriétés de l'agent SNMP

Les étapes de configuration du protocole SNMP sous Windows, est la même; que ce soit Windows XP, Vista ou Seven. Pour ce faire vous devez suivre les étapes suivantes:

- Ouvrez le Panneau de configuration, puis cliquez sur système.
- Ensuite cliquez sur "Outils d'administration" et ouvrez le Gestionnaire des Services, et rechercher le service SNMP.
- En cliquant avec le bouton droit sur le service, un menu apparait et cliquez sur Propriétés.

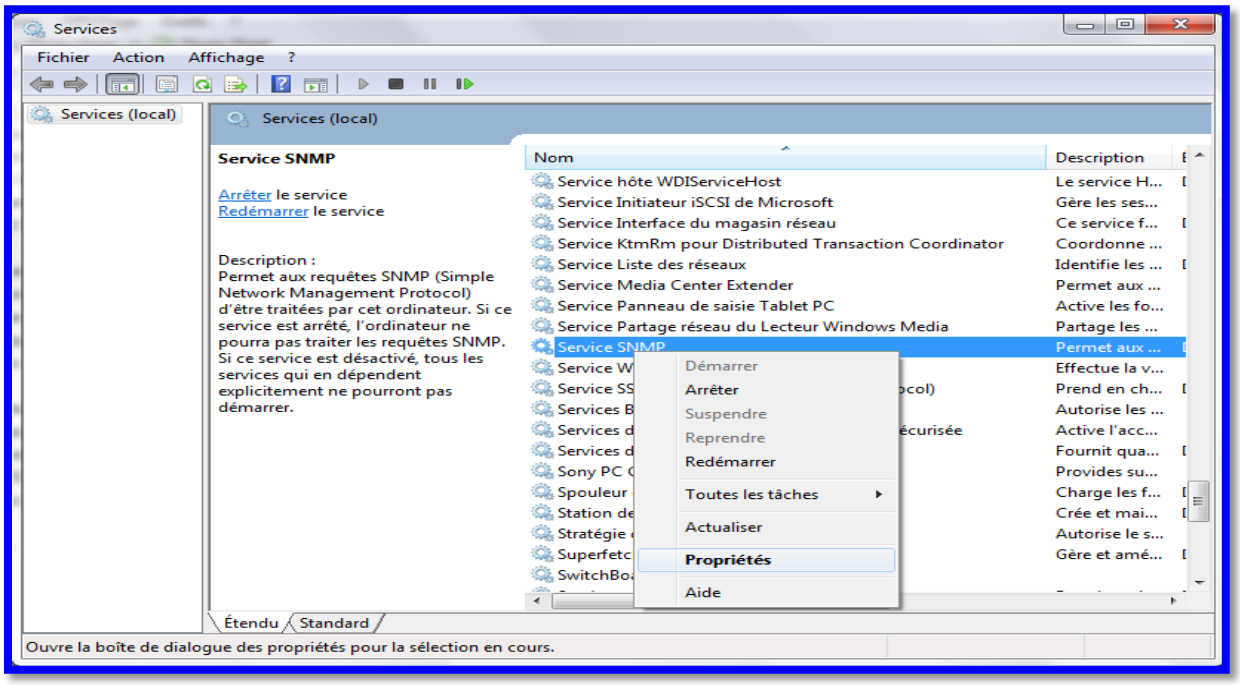


Figure 5.6: Configuration du protocole SNMP sous Windows

Gestion de l'agent (dans les propriétés du service SNMP **):**

- **Sous l'onglet Agent** : Dans Contact, tapez le nom de l'utilisateur ou de l'administrateur de cet ordinateur.
- Dans Emplacement, tapez l'emplacement physique de l'ordinateur.
- Sous Service, activez les cases à cocher appropriées pour cet ordinateur, puis cliquez sur OK.

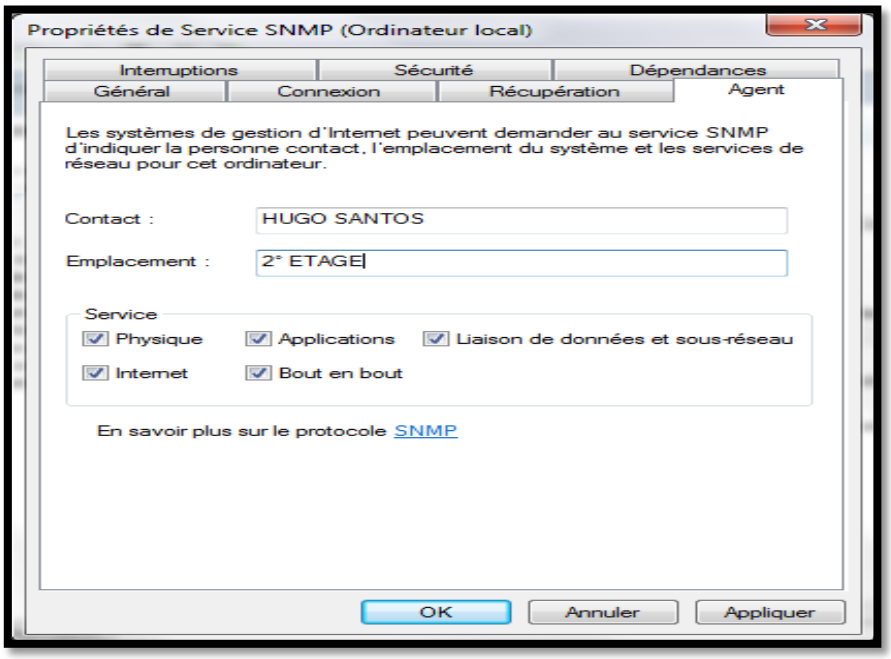


Figure 5.7: configuration de SNMP sous Windows

Gestion de la sécurité : Toujours dans les propriétés du service SNMP :

- **Sous l'onglet Sécurité** : Activez Envoyer une interruption d'authentification, si vous souhaitez qu'un message d'interruption soit envoyé à chaque échec d'authentification.
- Sous Noms de communautés acceptés, cliquez sur Ajouter, puis tapez un nom de communauté en respectant la casse, puis Sous Droits de communauté, sélectionnez un niveau d'autorisation ("lecture seul", "lecture et écriture" ou "lecture et création") pour permettre à l'hôte de traiter les requêtes SNMP en provenance de la communauté choisie. puis cliquez sur Ajouter.
- Dans Nom de communauté sur Propriétés de service SNMP, indiquez si les paquets SNMP en provenance d'un hôte sont acceptés ou non : Pour accepter les requêtes SNMP provenant d'un hôte du réseau, quelle que soit son identité, cliquez sur Accepter les paquets SNMP provenant de n'importe quel hôte. Pour limiter l'acceptation de paquets SNMP, cliquez sur Accepter les paquets SNMP provenant de ces hôtes, sur Ajouter, tapez le nom d'hôte ou l'adresse IP ou IPX appropriés, puis cliquez à nouveau sur Ajouter.

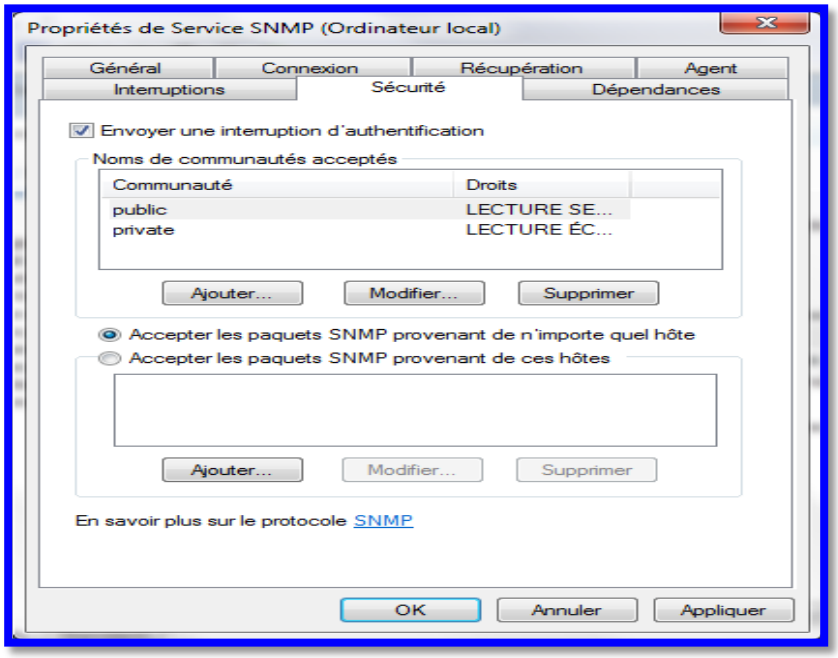


Figure 5.8: Configuration de SNMP sous Windows

Configuration des interruptions SNMP

- Pour configurer des interruptions : Cliquez sur l'onglet **Interruptions**. Puis dans la zone Nom de la communauté, tapez le nom de communauté vers laquelle l'ordinateur enverra les messages d'interception (en respectant la casse), puis cliquez sur Ajouter à la liste.
- Sous Destinations des interruptions, cliquez sur Ajouter. Puis dans la zone Nom d'hôte, adresse IP ou IPX, tapez le nom, l'adresse IP ou l'adresse IPX de l'hôte, puis

cliquez sur Ajouter. Le nom ou l'adresse de l'hôte apparaît dans la liste Destinations des interruptions.

- Répétez les étapes si vous voulez ajouter d'autres communautés et des destinations des interruptions. Puis cliquez sur OK.

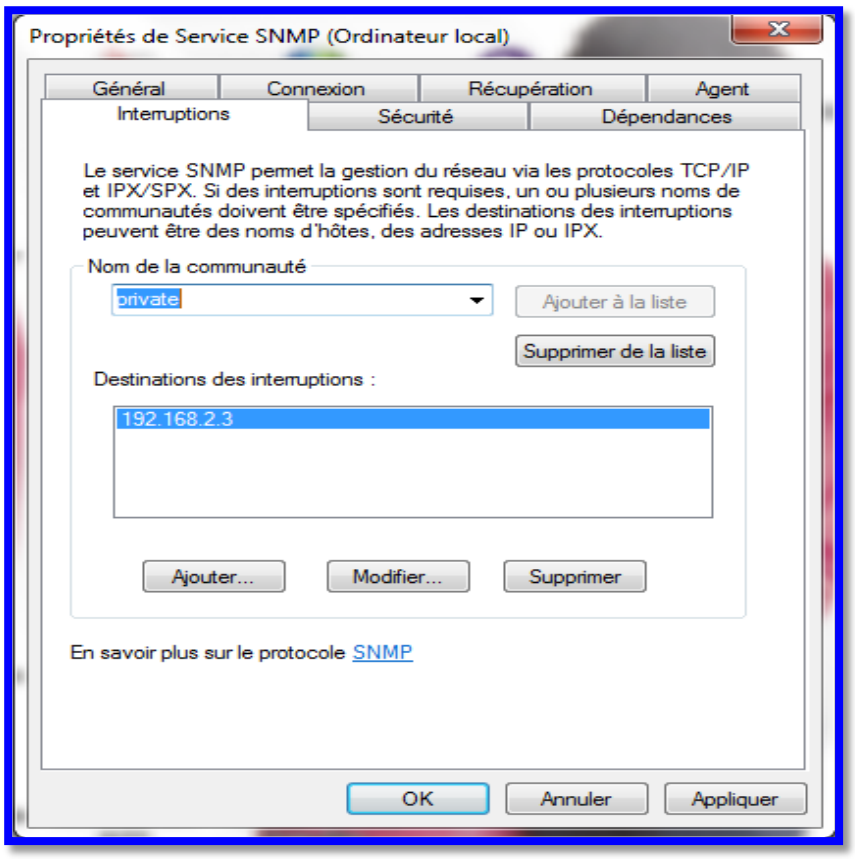


Figure 5.9: Configuration de SNMP sous Windows

3.2. Installation du service SNMP sous Linux UBUNTU

Les étapes suivantes nous montrent comment installer le service SNMP sous Linux UBUNTU. Pour le faire, nous avons utilisé le package NET-SNMP qui est un standard libre pour la mise en œuvre de SNMP sous Linux.

3.2.1. Le package NET-SNMP :

Le projet NET-SNMP a été développé par l'université américaine Carnegie Mellon University puis amélioré et maintenant maintenu par l'université américaine University of California Davis. NET-SNMP est un ensemble d'applications utilisées pour implémenter le protocole SNMP (v1, v2c & v3) utilisant à la fois l'IPv4 & IPv6. Il possède notamment les outilles et les fonctionnalités suivantes :

- Une API d'accès à SNMP.
- Un agent SNMP extensible.
- Des commandes en ligne pour interroger des agents SNMP.
- Des commandes en ligne pour gérer et générer des TRAPs SNMP.
- Un browser de MIB SNMP (TKMIB).

NET-SNMP supporte SNMPv1, SNMPv2 et SNMPv3 que ce soit côté agent SNMP comme du côté manager SNMP via les commandes en ligne NET-SNMP.

3.2.2. Installation de NET-SNMP :

Avant de commencer l'installation de Net-SNMP assurez vous, qui vous êtes connecté à l'internet. Puis ouvrez une fenêtre Terminal et exécuter la commande suivante :

```
# sudo apt-get install snmp snmpd
```

Tapez le mot-de-passe administrateur et l'installation démarre toute suite après.

3.2.3. Configuration de NET-SNMP :

Pour configurer NET-SNMP, il suffit d'éditer le fichier « /etc/snmp/snmpd.conf » comme suit :

```
syscontact Contact_de_la_machine
syslocation Localisation_de_la_machine
# sec.nom    source          community
com2sec readonly  default        public
com2sec readwrite default        private
# Creation des vues
view all    included  .1
view system included  .1.3.6.1.2.1.1
```

Une fois ces lignes ajoutez au fichier snmpd.conf, il faut redémarrer le service avec la comande: **/etc/init.d/snmpd restart.** Et la configuration SNMP sur la machine Linux maintenant est active.

3.3. Activation et configuration de SNMP sur le Routeur 3600 de Cisco

Les étapes à suivre pour l'installation et la configuration d'un routeur Cisco sont les suivantes:

- Pour des raisons de sécurité il est conseillé d'utiliser une liste d'accès contenant le(s) réseau(x) autorisé à se connecter sur le router avant toute autre configuration. Dans notre cas voici la liste de contrôle d'accès utilisé est: " **#access-list 1 permit 192.168.0.0 0.0.255.255**".

- Pour l'activation de l'agent (serveur) SNMP sur le routeur il suffit de créer des communauté en suivant cette syntaxe: " **snmp-server community nom_communauté [view nom_vue] [RO|RW] [access_list number]** ". Dans notre cas, on a utilisé pour la création des communautés les commandes suivantes:
    - **snmp-server community public view view1 RO 1**
    - **snmp-server community lecture view view3 RO 1**
    - **snmp-server community private view view2 RW 1**
    - **snmp-server community ecriture view view4 RW 1**
- Ensuite, il faut ajouter à chacune des vue créé les parties de l'arborescence MIB que nous intéresse, avec la syntaxe suivante: " **#snmp-server view nom_vue nom_de_la_partie_mib included** ". Dans le cadre de la configuration de notre routeur on a utilisé les commandes suivantes
    - **snmp-server view view1 system included**
    - **snmp-server view view1 interfaces included**
    - **snmp-server view view1 ip included**
    - **snmp-server view view1 snmp included**
    - **snmp-server view view1 ospf included**
    - **snmp-server view view1 cisco included**
    - **snmp-server view view3 system included**
    - **snmp-server view view3 interfaces included**
    - **snmp-server view view3 ip included**
    - **snmp-server view view4 system included**
    - **snmp-server view view4 interfaces included**
    - **snmp-server view view4 ip included**
    - **snmp-server view view4 snmp included**
    - **snmp-server view view4 ospf included**
    - **snmp-server view view4 cisco included**
- Une fois que les parties de la MIB sont associés au vue des communautés on configure les interruptions (Traps) et la destination des interruptions (le manager), comme suit:
    - **snmp-server enable traps vrrp**
    - **snmp-server enable traps ds1**
    - **snmp-server enable traps tty**
    - **snmp-server enable traps eigrp**
    - **snmp-server enable traps casa**
    - **snmp-server enable traps xgcp**
    - **snmp-server enable traps isdn call-information**
    - **snmp-server enable traps isdn layer2**
    - **snmp-server enable traps isdn chan-not-avail**
    - **snmp-server enable traps isdn ietf**
    - **snmp-server enable traps icsudsu**
    - **snmp-server enable traps ds3**
    - **snmp-server enable traps hsrp**

- **snmp-server enable traps config**
- **snmp-server enable traps entity**
- **snmp-server enable traps cpu threshold**
- **snmp-server enable traps config-copy**
- **snmp-server enable traps flash insertion removal**
- **snmp-server enable traps envmon**
- **snmp-server enable traps ds0-busyout**
- **snmp-server enable traps ds1-loopback**
- **snmp-server enable traps bgp**
- **snmp-server enable traps ospf state-change**
- **snmp-server enable traps ospf errors**
- **snmp-server enable traps ospf retransmit**
- **snmp-server enable traps ospf lsa**
- **snmp-server enable traps ospf cisco-specific state-change nssa-trans-change**
- **snmp-server enable traps ospf cisco-specific state-change shamlink interface-old**
- **snmp-server enable traps ospf cisco-specific state-change shamlink neighbor**
- **snmp-server enable traps ospf cisco-specific errors**
- **snmp-server enable traps ospf cisco-specific retransmit**
- **snmp-server enable traps ospf cisco-specific lsa**
- **snmp-server enable traps pim neighbor-change rp-mapping-change invalid-pim-message**
- **snmp-server enable traps ipmulticast**
- **snmp-server enable traps mvpn**
- **snmp-server enable traps msdp**
- **snmp-server enable traps rsvp**
- **snmp-server enable traps frame-relay**
- **snmp-server enable traps frame-relay subif**
- **snmp-server enable traps syslog**
- **snmp-server enable traps rtr**
- **snmp-server enable traps bulkstat collection transfer**
- **snmp-server enable traps mpls ldp**
- **snmp-server enable traps mpls traffic-eng**
- **snmp-server enable traps mpls vpn**
- **snmp-server enable traps cnpd**
- **snmp-server enable traps snasw alert isr topology cp-cp port link dlus**
- **snmp-server enable traps stun**
- **snmp-server enable traps dlsw**
- **snmp-server enable traps bstun**
- **snmp-server enable traps pppoe**

- **snmp-server enable traps l2tun session**
- **snmp-server enable traps atm subif**
- **snmp-server enable traps dial**
- **snmp-server enable traps dsp card-status**
- **snmp-server enable traps ipmobile**
- **snmp-server enable traps vtp**
- **snmp-server enable traps director server-up server-down**
- **snmp-server enable traps isakmp policy add**
- **snmp-server enable traps isakmp policy delete**
- **snmp-server enable traps isakmp tunnel start**
- **snmp-server enable traps isakmp tunnel stop**
- **snmp-server enable traps ipsec cryptomap add**
- **snmp-server enable traps ipsec cryptomap delete**
- **snmp-server enable traps ipsec cryptomap attach**
- **snmp-server enable traps ipsec cryptomap detach**
- **snmp-server enable traps ipsec tunnel start**
- **snmp-server enable traps ipsec tunnel stop**
- **snmp-server enable traps ipsec too-many-sas**
- **snmp-server enable traps event-manager**
- **snmp-server enable traps voice poor-qov**
- **snmp-server enable traps voice fallback**
- **snmp-server enable traps dnis**
- **snmp-server host 192.168.2.3 version 2c public**

La dernière ligne sert à indiquer l'adresse IP du manager SNMP dans notre architecture réseau.

4. Topologie

La topologie que nous avons utilisé lors de la réalisation de ce travail comprend six hôtes (terminaux) deux Switchs et un routeur, comme nous montre la figure 10.

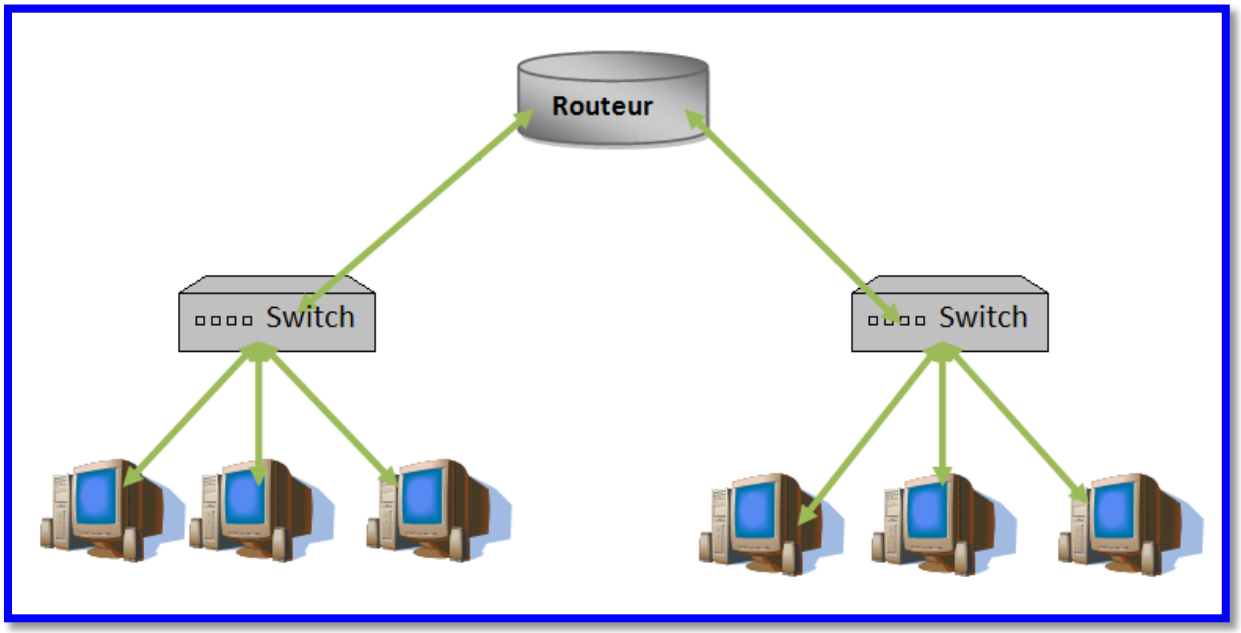


Figure 5.10: Topologie réseau utilisé

4.1.Connexion Global

Lors de la réalisation de cette topologie on s'est retrouvé avec deux petits défis:

1. Le premier consistait à: comment connecter les machines virtuelles situées au niveau de la machine physique où est installé le GNS3 au routeur dans GNS3.
2. Le deuxième consistait à: comment connecter la deuxième machine réelle ainsi que ses machines virtuelles au routeur dans GNS3 situé dans la première machine.

Pour résoudre le premier point on a procédé comme suit:

– On a ouvert le Vmware Workstation, puis on a cliqué sur le menu "Edit", ensuite sur "Virtual network editor", puis sur la liste de VMnet proposé on a configuré le VMnet2 comme nous montre la figure 11.

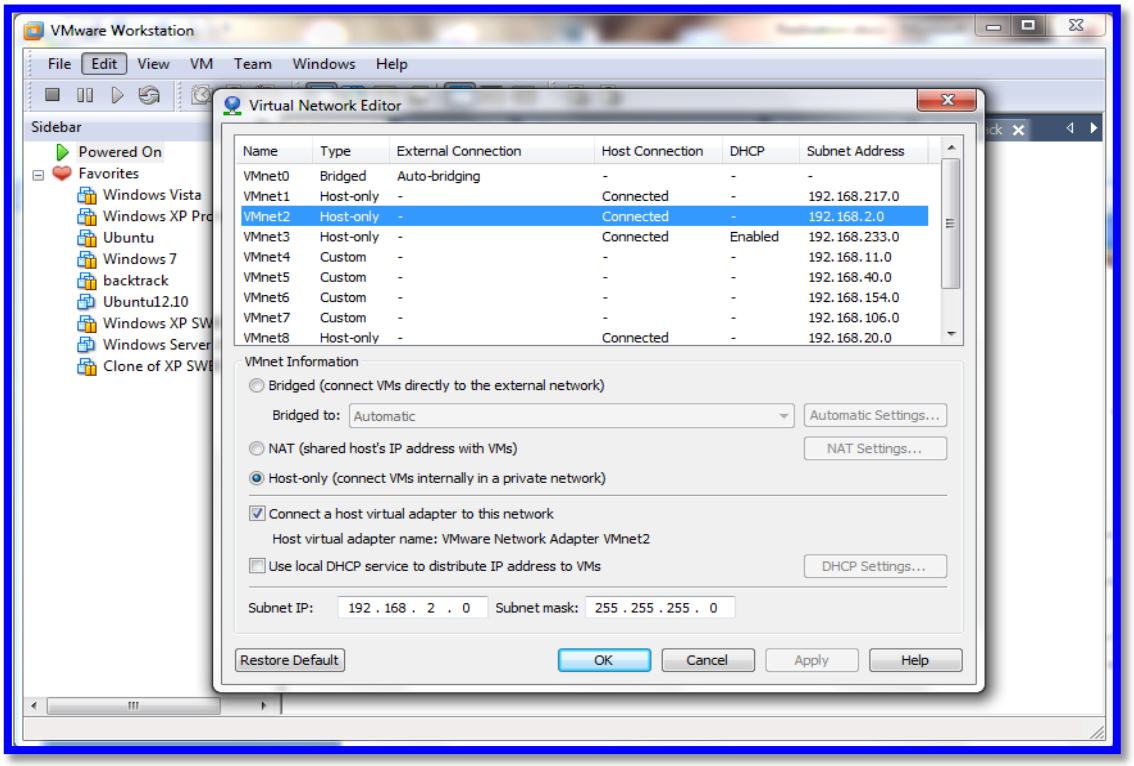


Figure 5.11: Configuration du VMnt2

– Une fois cette configuration terminée, on a configuré les OS des machines virtuelles pours qu'elles se connectent au VMnet2 comme suit:

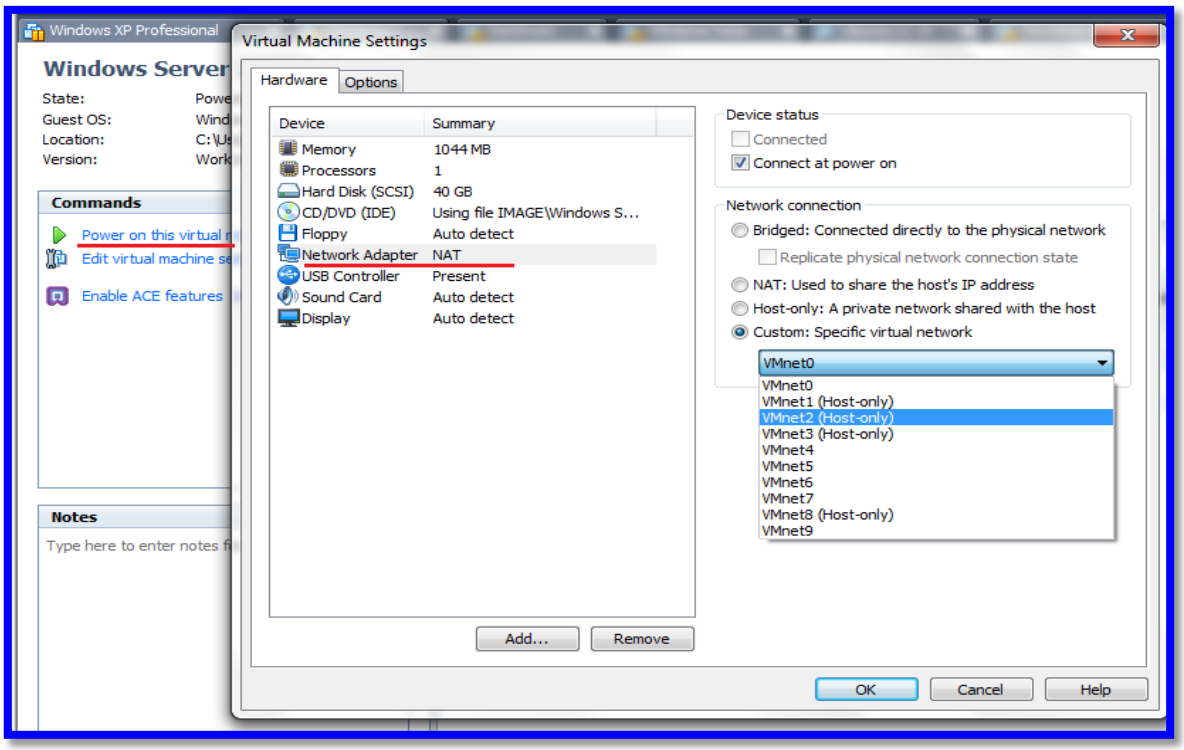


Figure 5.12: Connexion de l'hôte au VMnet2.

- On procède de la même façon pour la deuxième machine virtuelle. Ensuite on donne des adresses IP aux machines dans le réseau 192.168.2.0, qui est celui du VMnet2.
- Puis deux cliques sur GNS3, on crée un projet et on met le routeur 3600 sur l'espace en blanc avec deux Switch (les Switch ne sont là que pour une question d'esthétique), puis on met deux nuages Cloud dans l'espace en blanc, ensuite on fait un clique droit sur la souris au niveau du premier nuage et on clique sur configure, puis sous l'onglet "Nio Ethernet" on choisit dans la liste proposé le VMnet2 et on clique sur "Add". Maintenant les machines virtuelles sur la machine un, sont en mesure de communiquer avec le routeur.

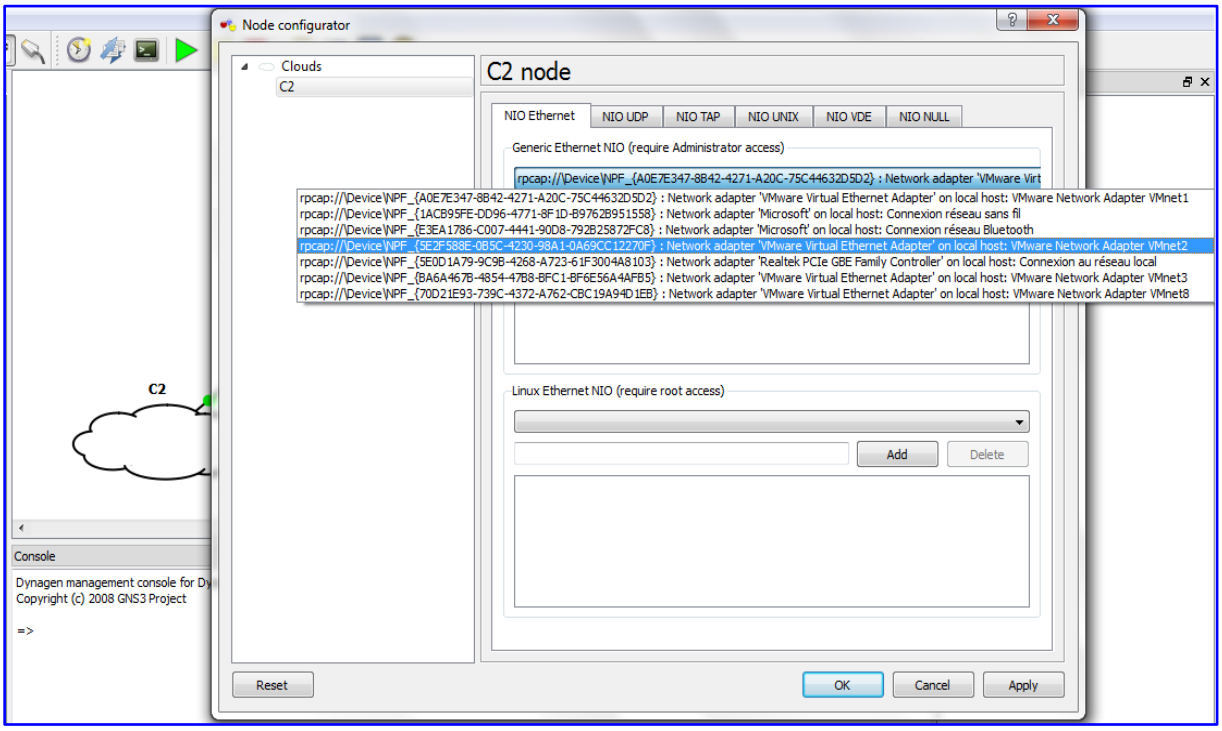


Figure 5.13: connexion du VMnet2 au routeur sur GNS3

Pour résoudre le deuxième point on a procédé comme suit:

- Puisqu'on était déjà sur GNS3, on configure le deuxième nouage Cloud presque de la même façon que le premier, la différence ici est qu'on sélectionne la carte réseau physique local.
- Ensuite on part sur la deuxième machine réelle et on configure ses machines virtuelles sur Bridge (i.e. que les machines seront directement connectées à la carte physique).

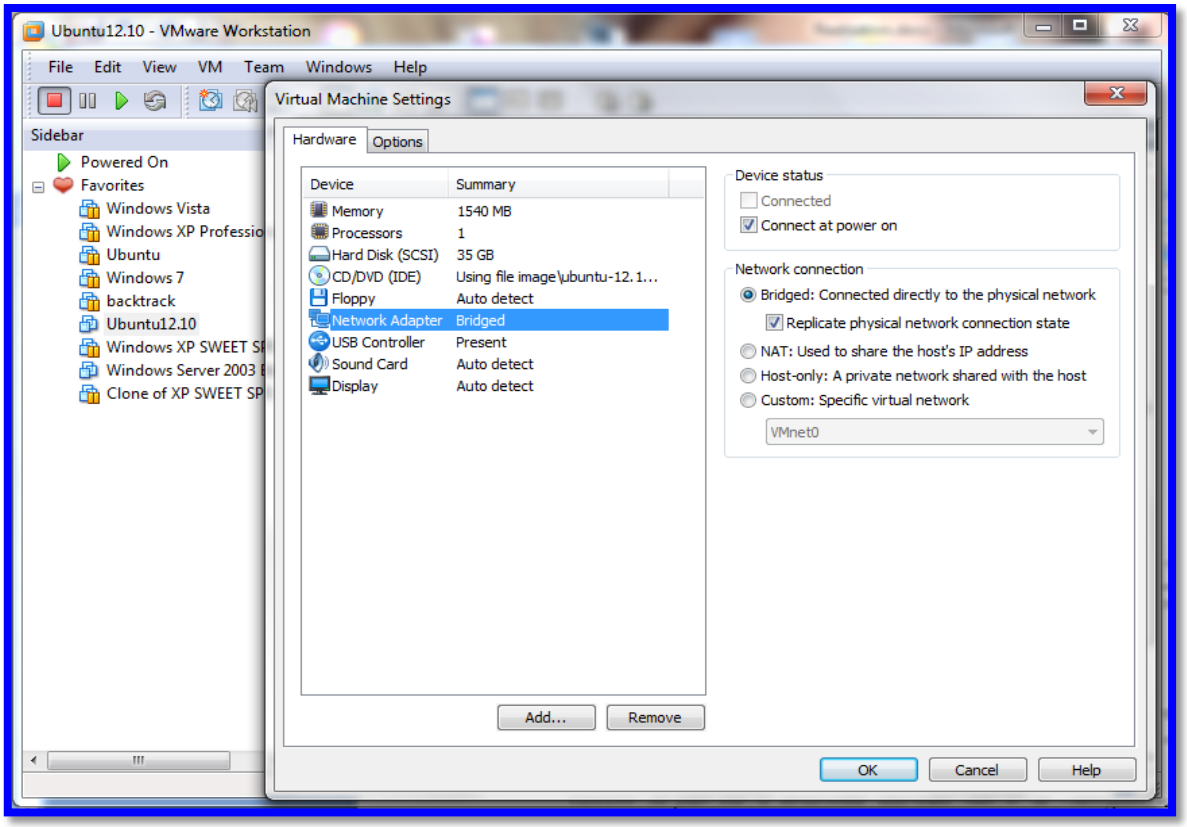


Figure 5.14: configuration des machines virtuels du deuxième Ordinateur.

– Il nous suffit maintenant de connecter les deux ordinateurs à l'aide d'un câble croisé et on aura une communication de bout en bout si un bon adressage a été fait.

Cella fait, notre topologie réseau est enfin terminée. La figure 15 donne la vue final de la configuration de la topologie sur GNS3.

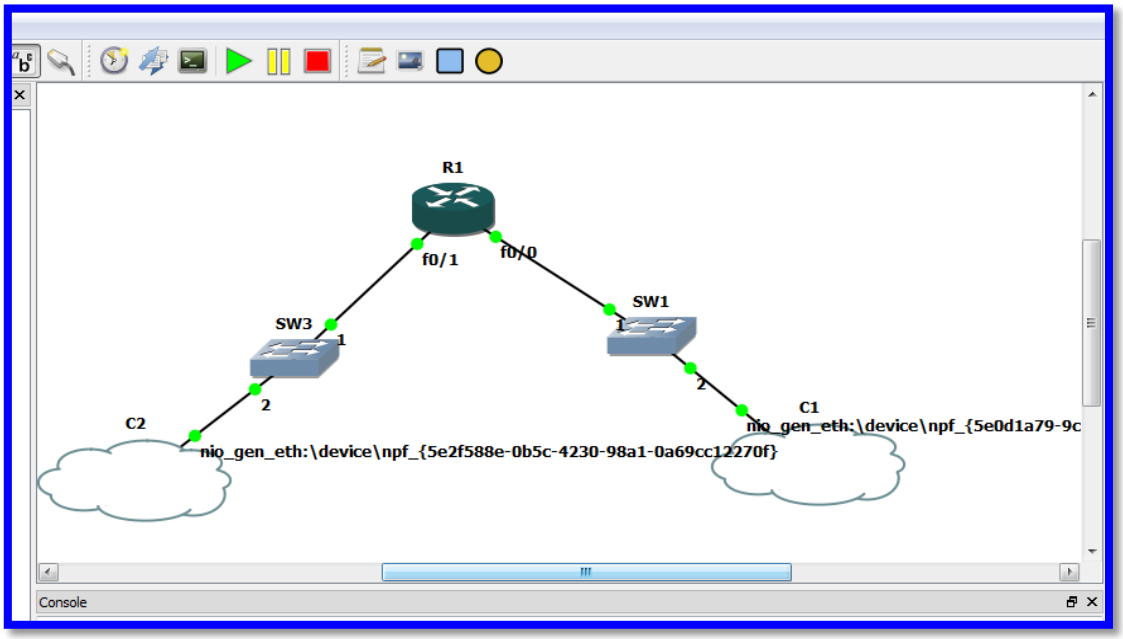


Figure 5.15: configuration final de la topologie sur GNS3.

5. Présentation de l'application SNMP

Dans ce qui suit on présente les principales classes et les interfaces graphiques de notre application ainsi que chacune des ses fonctions.

5.1.L'interface graphique SNMPGetSetRequest

L'interface SNMPGetSetRequest peut être considéré comme le cœur de notre applications, puisqu'elle permet de questionner et de configurer plusieurs valeurs de la MIB des agents (serveur) SNMP. Elle est composée de plusieurs fonctions comme nous montre la figure 16.

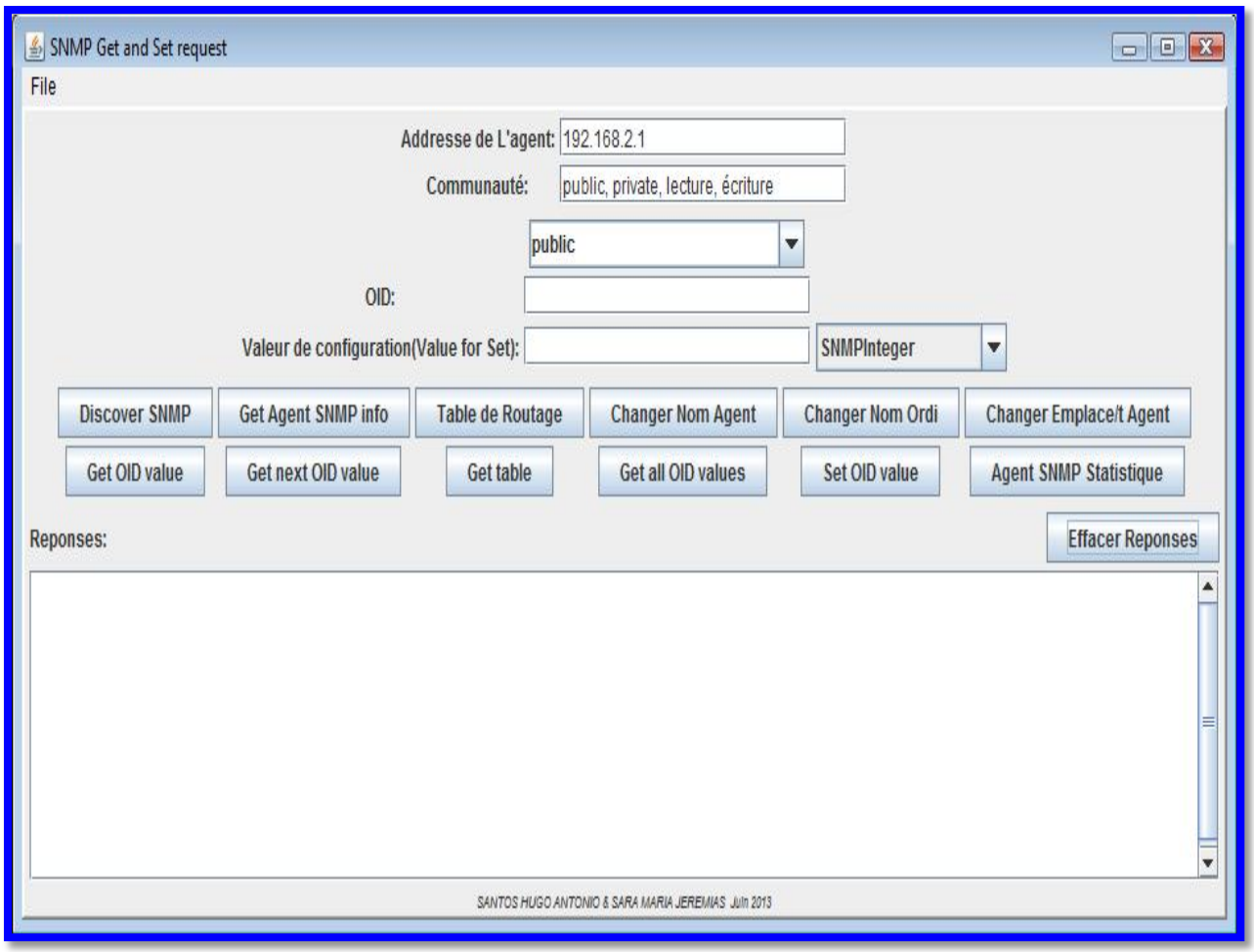


Figure 5.16: Interface SNMPGetSetRequest.

5.1.1- La fonction-Discover SNMP:

C'est le plus souvent la première commande à être exécuté lors d'une procédure de surveillance ou de gestion SNMP, puisqu'elle permet de chercher les hôtes présent sur le réseau ayant configuré le service SNMP.

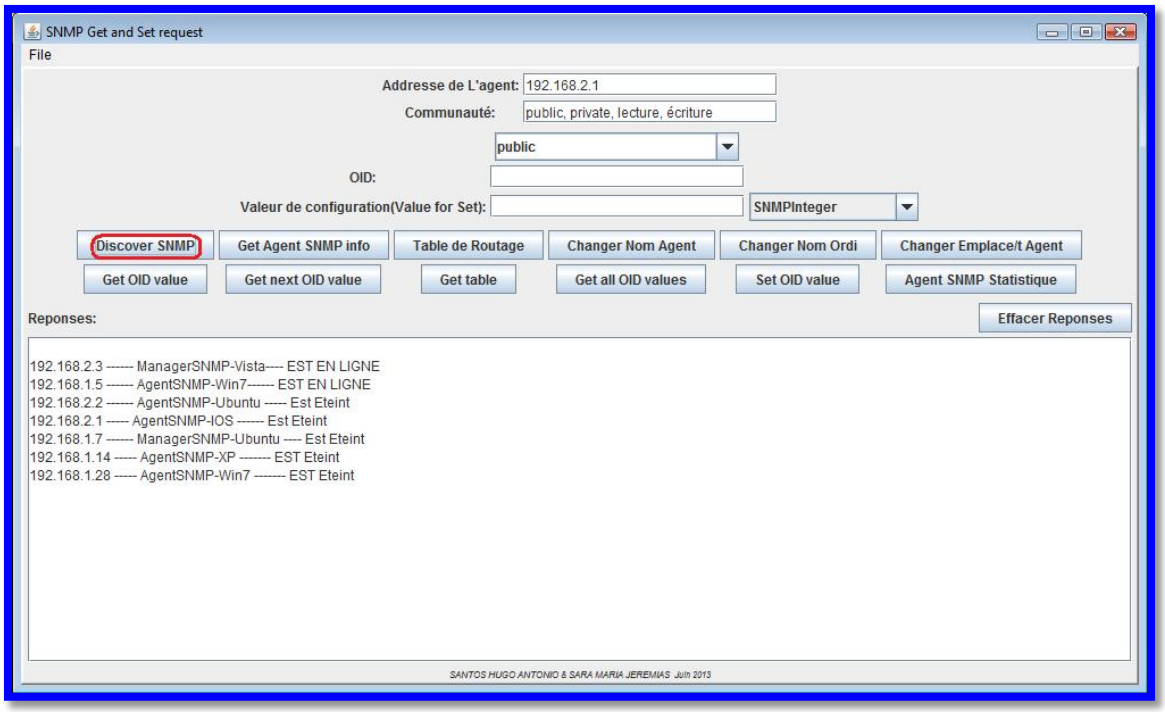


Figure 5.17: découverte des service SNMP

5.1.2- La Fonction Get-Agent-SNMP-info

Cette fonction nous permet d'obtenir les informations de base d'un Agent SNMP telles que le type et la version du système d'exploitation, le nom de l'agent, le nom de l'ordinateur sur le réseau, son emplacement, et son adresse Mac.

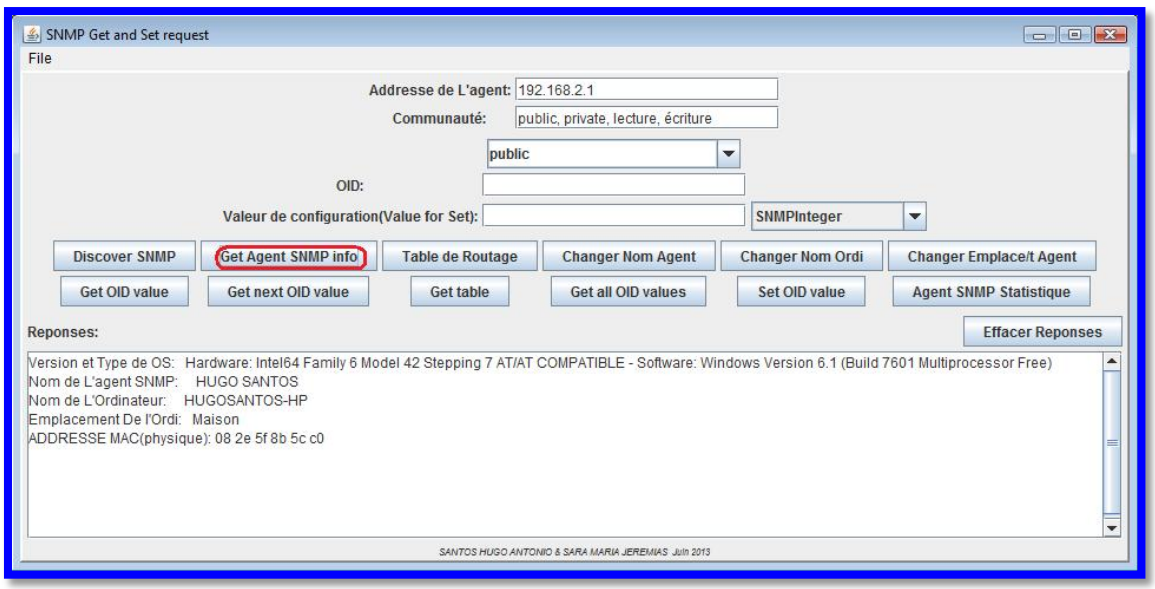


Figure 5.18: La fonction Get-Agent-SNMP-info

5.1.3- La Fonction Table-de-Routage

La fonction Table-de-Routage nous permet d'avoir la liste d'adresses IP que l'agent possède. Si l'agent ne contient aucun logiciel d'émulation d'OS, la liste d'adresses retournées est composée de 2 adresses, sinon elle aura les adresses des VMnet ainsi que l'adresse de la passerelle par défaut en plus.

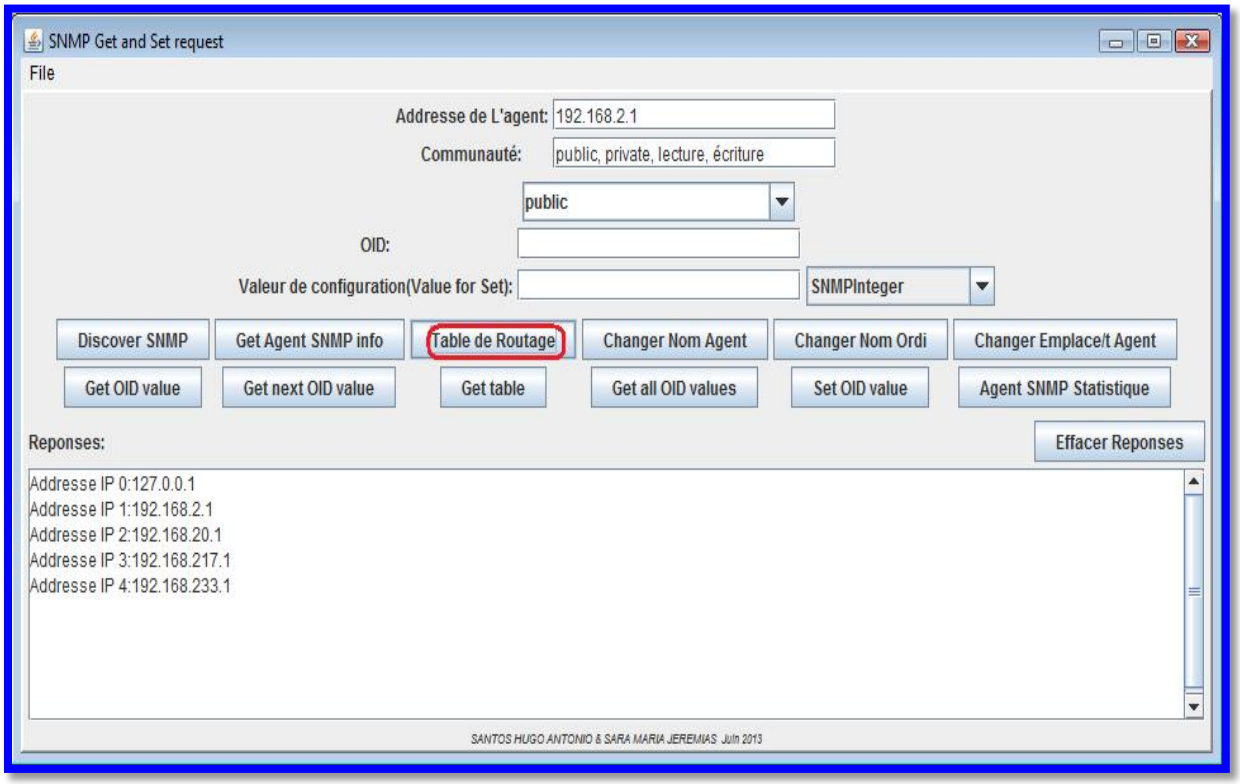


Figure 5.19: La fonction Table-de-Routage

5.1.4- Les fonctions de configuration de base.

Les fonctions de configuration de base, Changer-Nom-Agent, Changer-Emplace/t-Agent et Changer-Nom-Ordi, permettent successivement de configurer le nom de l'agent, l'emplacement et le nom de l'ordinateur, sans avoir besoin de spécifier le OID dans la MIB.

5.1.5- La fonction Get-OID-Value

Elle nous permet d'avoir n'importe quelle élément de la MIB de l'agent, à condition de donner l'OID de l'élément en paramètre. Elle ne retourne qu'une seule valeur.

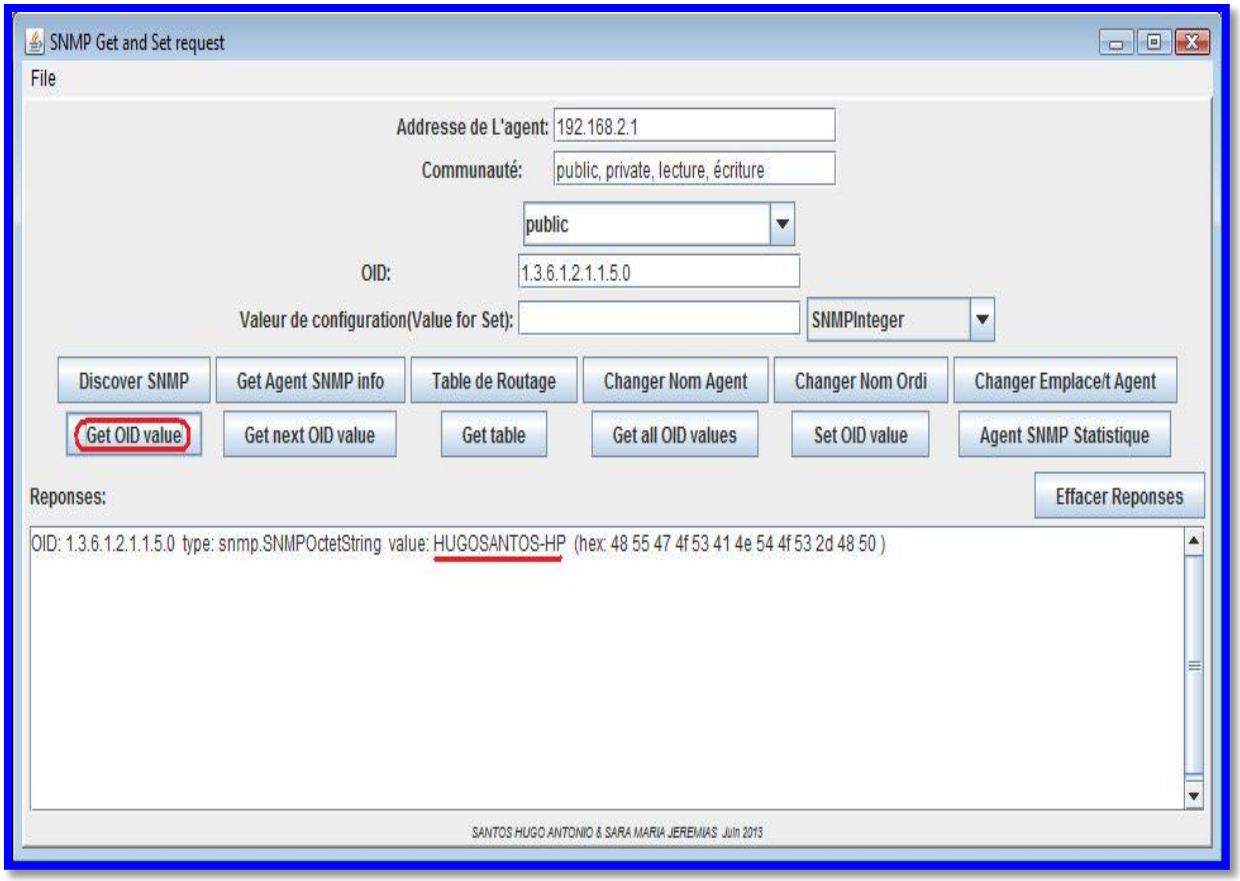


Figure 5.20: La fonction Get-OID-Value.

5.1.6- La fonction Get-Next-Value

Elle fonctionne presque comme la fonction Get-OID-Value, à la différence qu'elle retourne la valeur suivante dans la MIB de l'agent, par rapport à l'OID donnée en paramètre.

5.1.7- La fonction Get-Table

A la différence de la fonction Get-OID-Value, que ne permet de retourner qu'une seule valeur, la fonction Get-Table permet de retourner n'import quelle sous arbre de la MIB d'un agent.

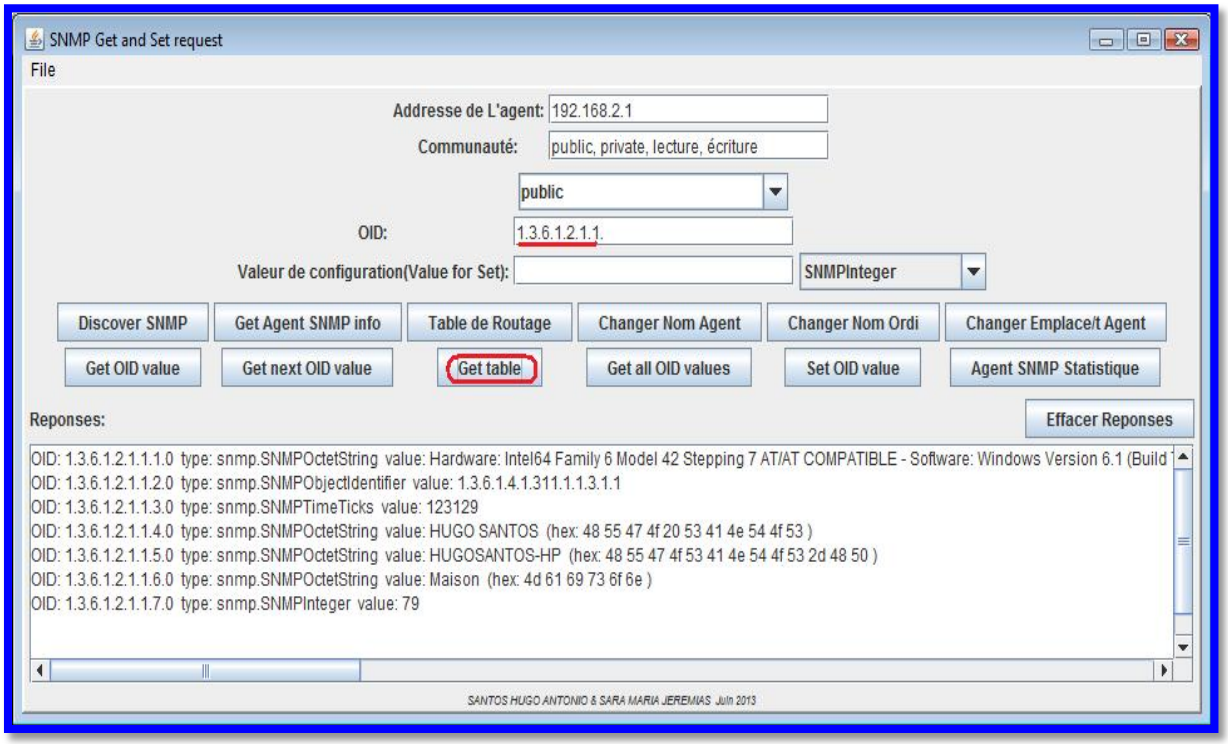


Figure 5.21: Fonction Get-Table

5.1.8- La fonction Get-All-OID-Values

Elle retourne tous les éléments de la MIB de l'agent.

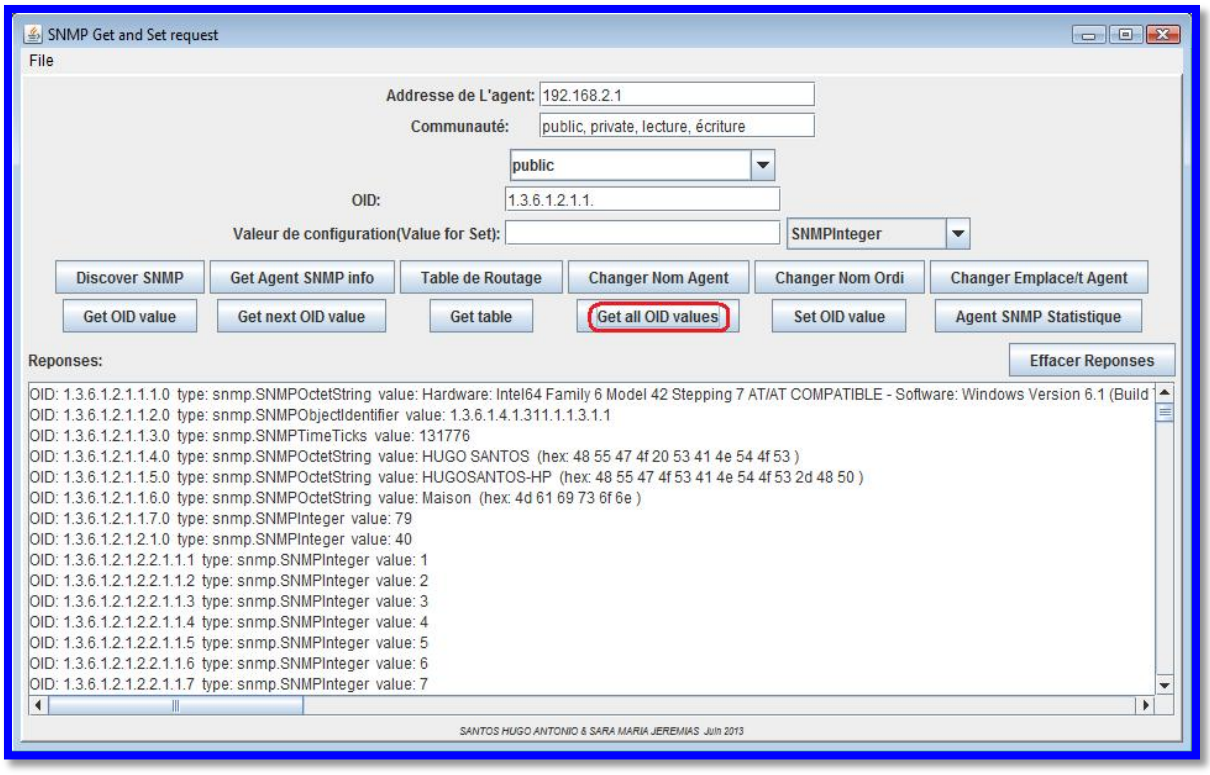


Figure 5.22: La fonction Get-All-OID-Values

5.1.9- La fonction Set-OID-Values.

Elle permet de configurer dans la MIB de l'agent les OID que sont en lecture et écriture grâce à une communauté configuré en lecture et écriture sous l'agent.

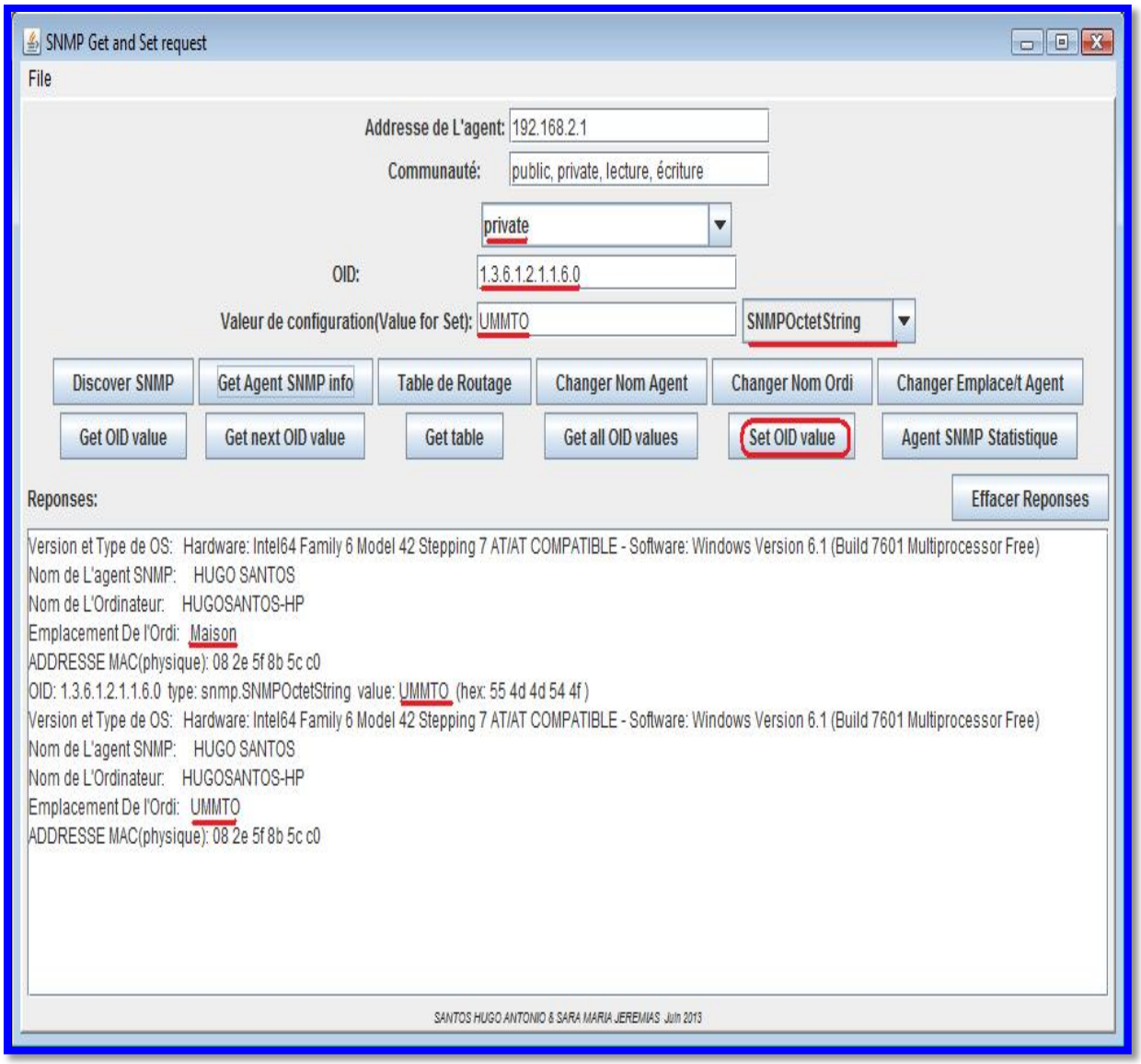


Figure 5.23: La fonction Set-OID-Value.

5.1.10- La fonction Agent-SNMP-Statistique

Cette fonction permet d'avoir des informations concernant le trafic SNMP sur l'agent, avec elle on peut avoir le nombre de Get, de GetNext , de Set, de Traps, d'accès refusé et le nombre de processus en exécution sur l'agent.

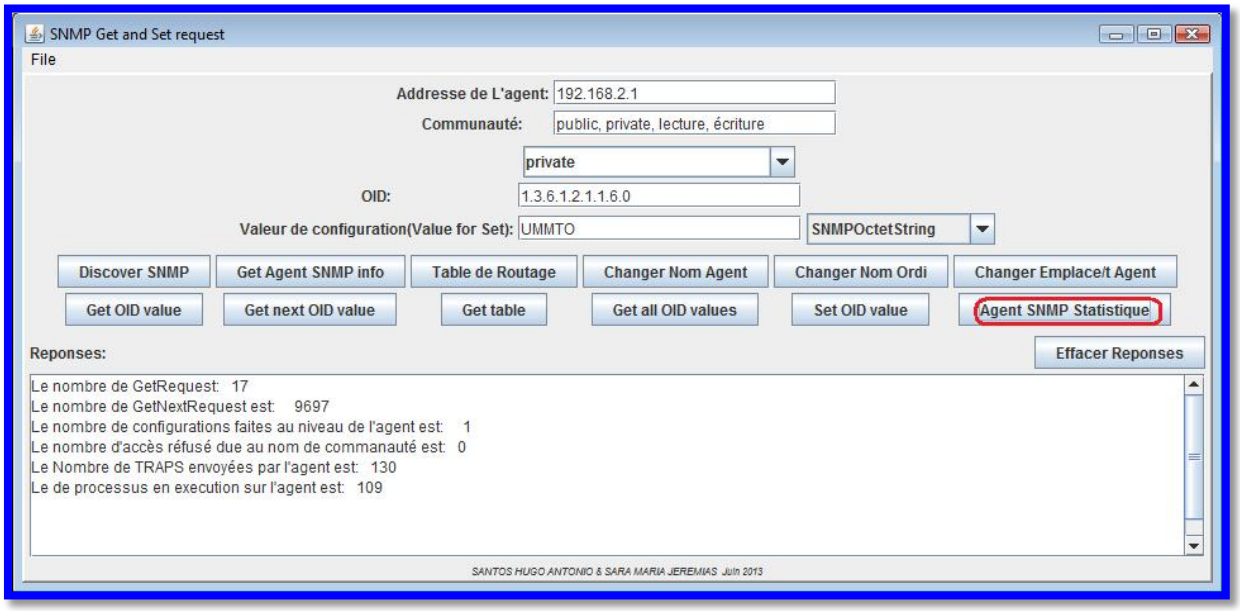


Figure 5.24 : La fonction Agent-SNMP-Satistique

5.2.L'interface TrapSurveillance

Comme sont nom l'indique, cette classe s'occupe de la réceptions des Traps émises par les agents SNMP.

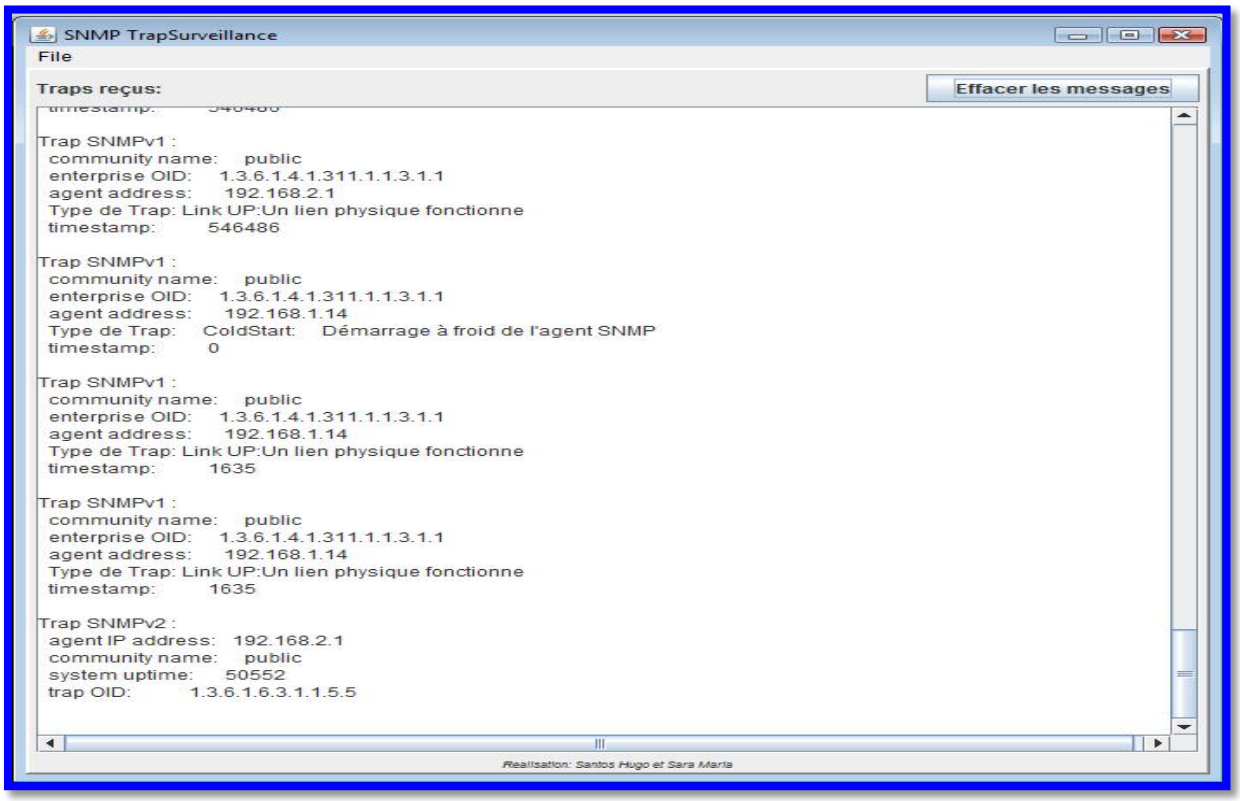


Figure 5.24: Réceptions des Traps.

Comme nos montre la figure ces Traps peuvent être de la version 1 de SNMP (envoyés par les hôtes terminaux) ou de la version 2 de SNMP (envoyés par le Routeur).

5.3. La classe Surveillance

La classe Surveillance s'exécute en arrière plan, elle s'occupe de surveiller la présence des hôtes sous le réseau. Pour ce faire, elle utilise une sonnerie et une boite de dialogue pour informer le manager dès qu'un hôte est déconnecté du réseau. Cette classe utilise les boites de dialogue illustré dans la figure 25 et 26.

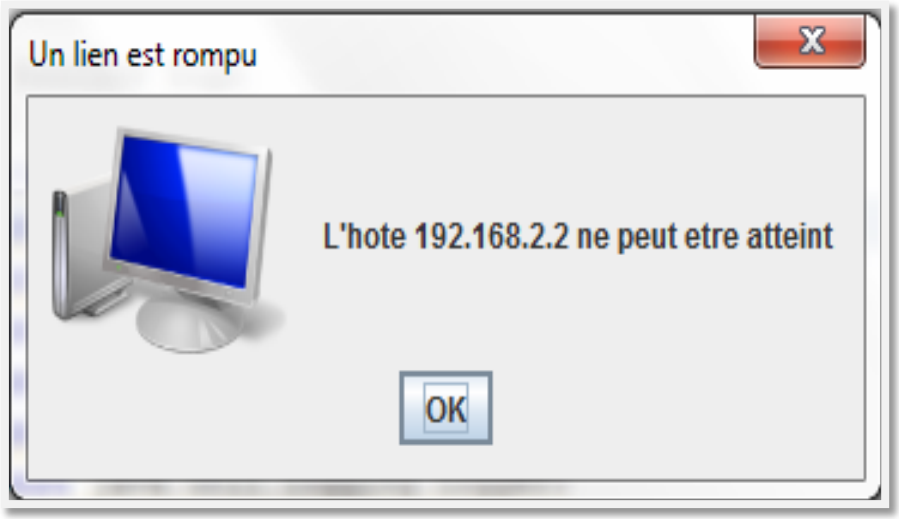


Figure 5.25: Boite de dialogue utilisé par les ordinateurs

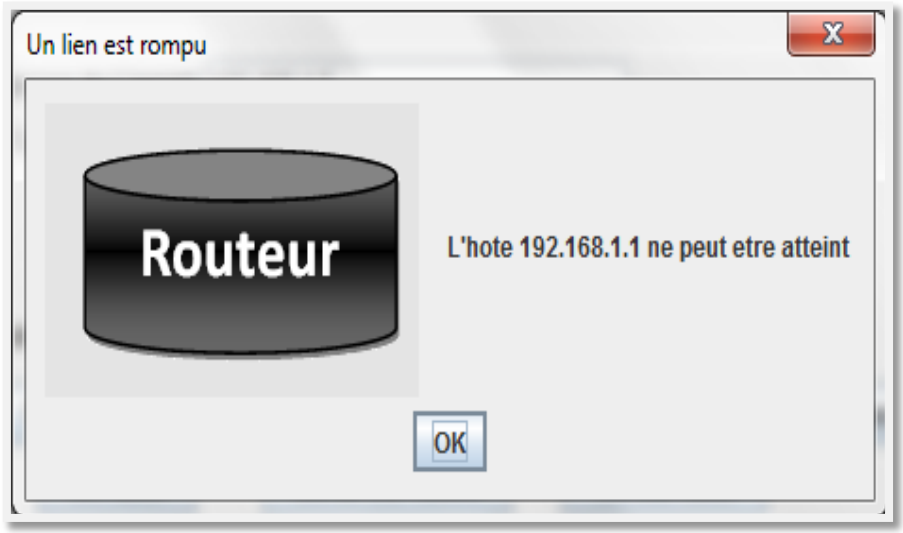


Figure 5.26: Boite de dialogue utilisé par le routeur

6. Phase de test

Nous avons testé l'application pendant 5 heures.

Pour tester notre application nous avons utilisé 6 hôtes et un Routeur Cisco. Ces six hôtes, n'utilisait pas tous le même système d'exploitation pour meilleur mettre en évidence la portabilité de l'application SNMP. Ils étaient repartis comme suit:

- Un hôte sous Windows XP
- Un hôte sous Windows Vista
- Deux hôtes sous Windows Seven
- Deux hôtes sous Linux Ubuntu

On a exécuté l'application Manager SNMP au niveau de l'hôte Windows Vista. Mise à part un des hôtes sous Linux Ubuntu, tous les autres ont été configuré comme agent SNMP.

La répartirions des hôtes

La répartirions des hôtes pour les deux segments réseau était la suivante

1- Segment Réseau 192.168.1.0 /24

- Windows Seven avec l'adresse IP 192.168.1.5
- Linux Ubuntu avec l'adresse IP 192.168.1.7
- Windows XP avec l'adresse IP 192.168.1.14
- Windows Seven avec l'adresse IP 192.168.1.28

2- Segment Réseau 192.168.2.0 /24

- Windows XP avec l'adresse IP 192.168.2.2
- Windows Vista avec l'adresse IP 192.168.2.3

Connexion

Pour la connexion des deux segments réseau, nous avons utilisé un câble réseau croisé et un routeur sous GNS3 configuré avec l'adresse 192.168.1.1 sous l'interface fastethernet 0/0 et 192.168.2.1 sous l'interface fastethernet 0/1.

Déroulement du test

Lors du lancement de l'interface SNMPGetSetRequest, il a fallut deux Discover-SNMP pour connaitre le nombre total de machine présent sous le réseau. Dès la Découverte des hôtes présents sous le réseau on a lancé l'interface TrapSurveillance (qui s'occupe des réceptions de Traps des agents) et la classe Surveillance (qui s'occupe de surveiller les agents et d'alerter le manager des qu'un agent se déconnecte du réseau) et nous avons laissé tourner l'application pendant une heure sans envoyer aucune requête sous le réseau.

Une heure passé, aucune déconnexion se produit dans le réseau, mais nous avons pu constater que le routeur à envoyez 8 Traps(alarmes) avec l'OID 1.3.6.1.4.1.9.9.41.2.0.1 qui concerne la MIB privé de Cisco.

Pendant les 4 heures restantes nous avons effectué plusieurs actions sous le réseau et testé le comportement de chaque hôte. Les actions effectués sont:

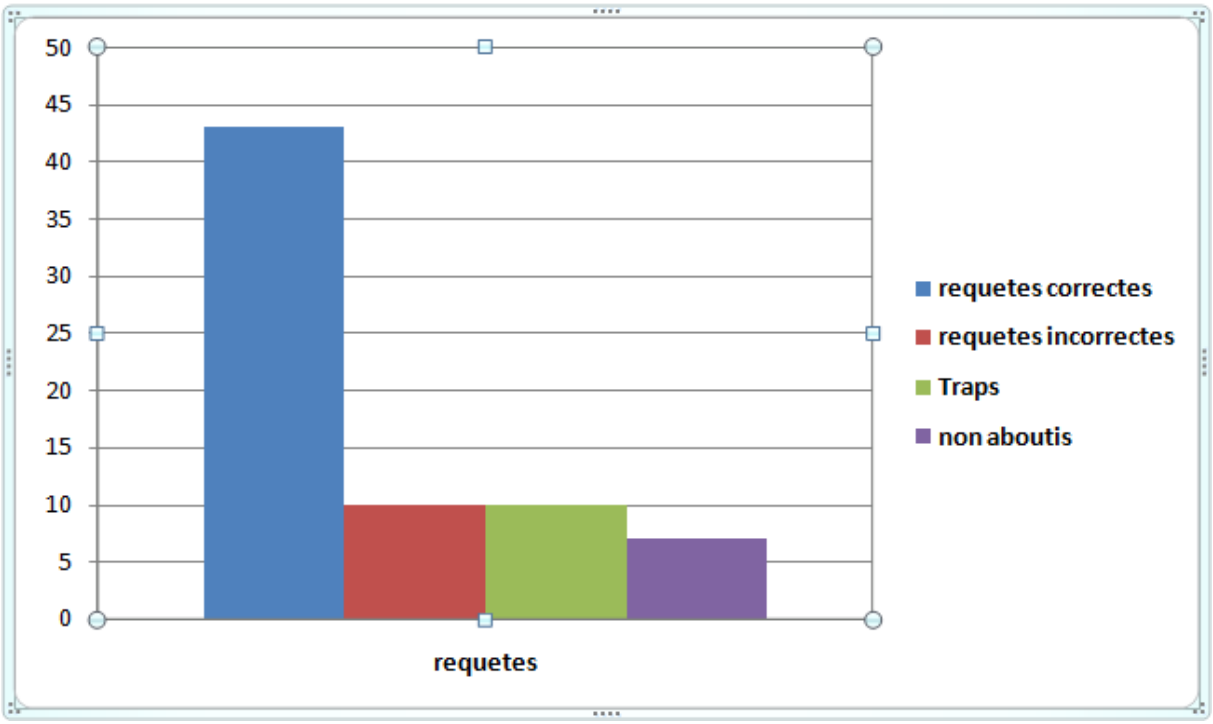
- Consultation de la MIB des tous les agent
- Modifications ou configurations de la valeur des OID de la MIB avec la communauté **private** pour les hôtes et **ecriture** pour le Routeur.
- Redémarrage du service SNMP et des agents.
- Déconnexion du câble réseau.

Consultation de la MIB des tous les agent

On à consulté la MIB de chaque agent du réseau 10 fois; deux fois pour chaque requête du type Get avec un nom de communauté correcte, et 2 fois avec une chaine de communauté erroné.

L'utilisations de chaines de communautés erroné a engendré un alarme (Trap) du type 4 pour chaque requête émise par le manager.

En somme lors de la consultation de la MIB des 5 agents on a émis sous le réseau 60 requêtes dont 10 erronés qui ont engendrée 10 alarmes et 7 qui n'ont pas pu aboutir car les agent se trouvé dans le segment réseau différente du manager.



Graphe 5.1: statistiques de la consultation de la MIB

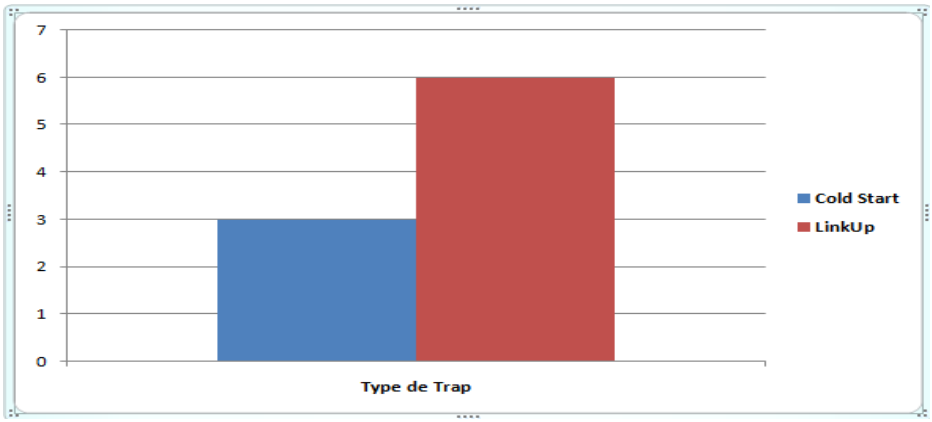
Modifications / configurations

Etant donnée que la plupart des OID de la MIB sont en lecture seule, surtout les types numériques, et parmi ceux en lecture et écriture tous ne sont pas très significatifs, nous avons modifié et configuré les OIDs qui concerne le nom de l'agent, le nom de l'ordinateur sur le réseau et l'emplacement de l'ordinateur.

Nous avons effectué trois requêtes pour chaque agent, ce qui fait un total de 15 requêtes de configurations.

Redémarrage du service SNMP et des agents

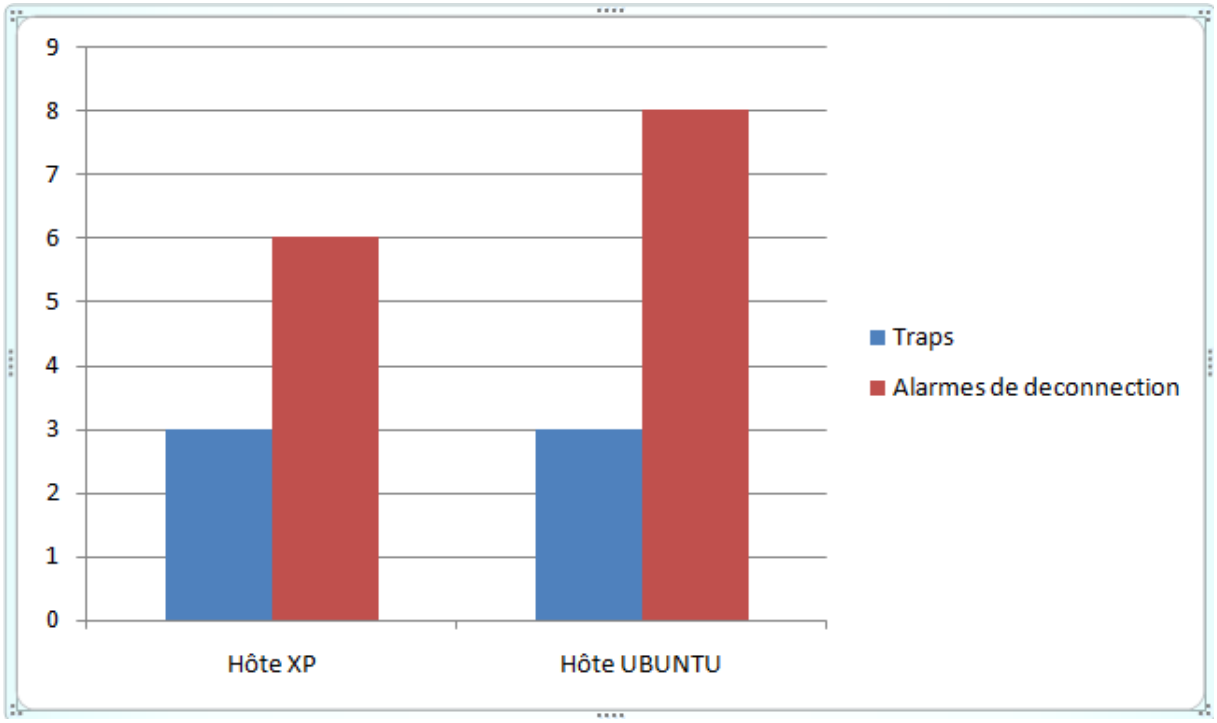
Lorsqu'on redémarre le service SNMP au niveau de l'agent, on reçoit une Trap de Coldstart (démarrage à froid de l'agent) et deux Traps de LinkUp (un lien physique fonctionne) . Lors de notre test nous avons redémarré le service SNMP trois fois sur l'agent, ce qui a produit 9 Traps, 3 de coldStart et 6 de LinkUP.



Graphe 5.2: Redémarrage du service SNMP

Lors du redémarrage de l'agent on reçoit les même nombre de Traps que lors du redémarrage du service SNMP, la différence ici est que la classe Surveillance signale la déconnection de l'hôte du réseau. Ainsi, en plus des Traps nous avons la classe Surveillance qui s'exécute en boucle qui donne le nombre de déconnection de chaque agent.

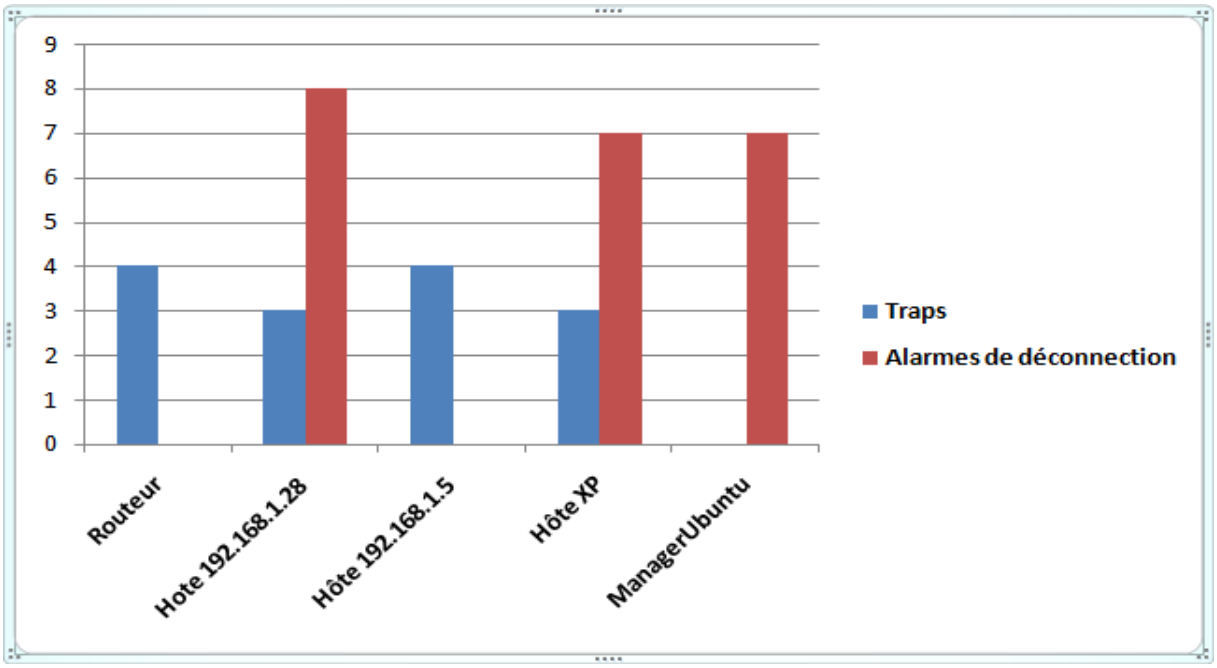
Nous avons tout d'abord redémarré l'hôte XP sous le réseau 192.168.1.0/24, ceci a engendré 3 Traps (1 coldStart et deux LinkUp) et 6 alarmes de déconnection. Ensuite nous avons redémarré l'hôte Ubuntu sous le réseau 192.168.2.0/24, ceci a engendré 3 Traps (1 coldStart et deux LinkUp) et 8 alarmes de déconnection.



Graphe 5.3 : Redémarrage des Hôtes

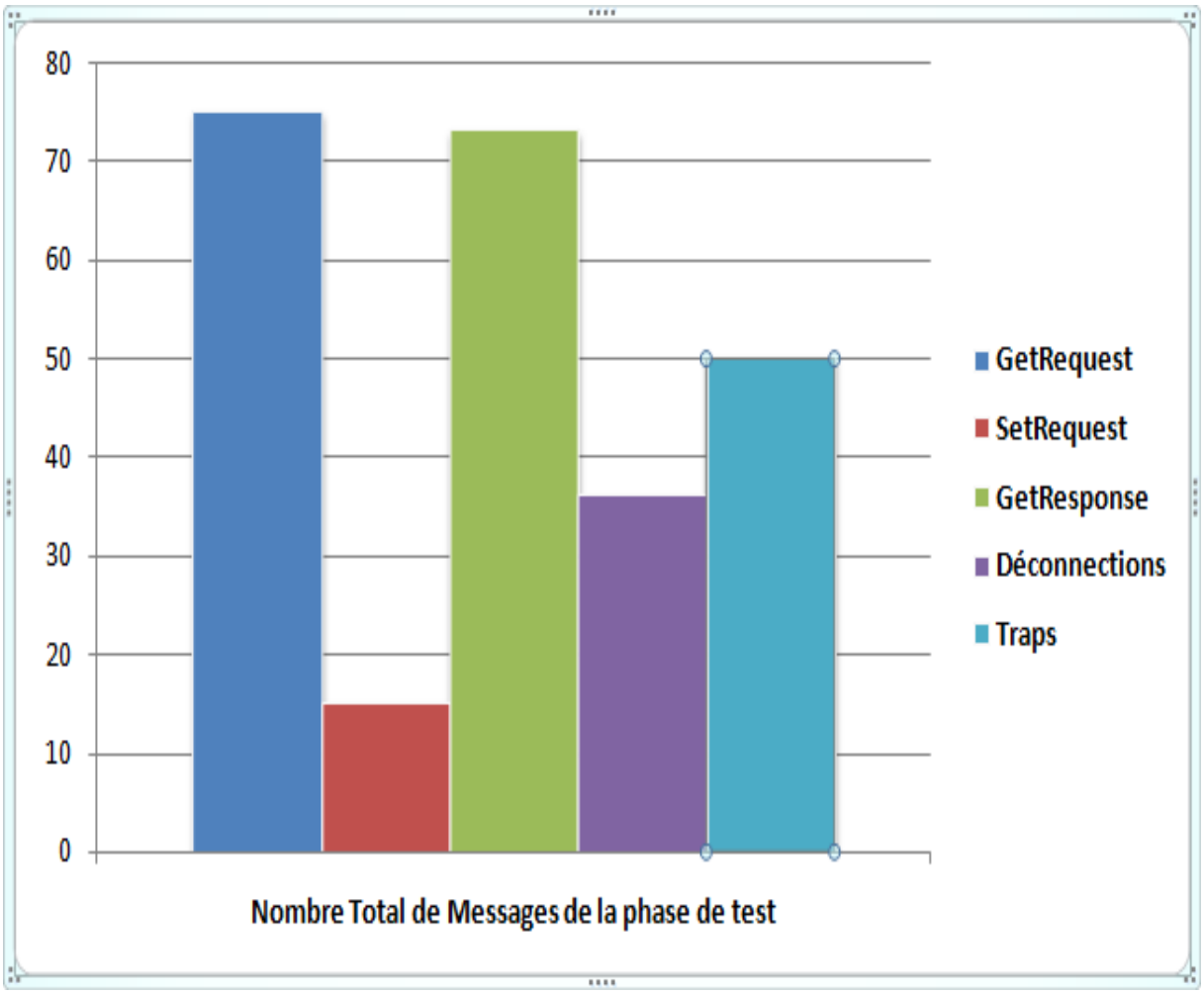
Déconnexion du câble réseau

Lorsqu'on a déconnecté le câble réseau pendant 2 minutes nous avons reçu 2 Traps LinkDown (un lien physique est rompu) et 22 alarmes de déconnection. Lorsqu'on a reconnecté on a reçus 12 Traps (4 coldStart et 8 LinkUp).



Graphe 5.4: résultats de la déconnection du câble

Les résultats des toutes les requêtes Get, Set, GetNext, et de messages d'alertes (Traps et déconnections) effectuées lors du test sont résumé dans le graphe 5.5.



Graphe 5.5: Nombre Total de messages de la phase de test

Conclusion

Le protocole SNMP est très puissant et très intéressant. Lors de ce chapitre, nous avons présenté l'implémentation de notre application dans différents systèmes d'exploitation et nous avons décrit les différents logiciels utilisés tout au long de notre implémentation. Nous avons aussi présenté les différentes interfaces et fonctionnalités de notre application. Mais elle n'est qu'une petite application que n'est pas en mesure d'exploiter toute la puissance du protocole SNMP.

Conclusion générale

L'installation et l'accès aux réseaux informatiques, augmente d'une façon exponentielle. La mise au point de systèmes de gestion de tels réseaux ainsi que des dispositifs qui leur sont connectés est indispensable. Ceci permet de faciliter la tache des administrateurs qui surveillent le bon fonctionnement de leur réseau et de fournir des meilleurs services aux utilisateurs. Aujourd'hui la question à « faut-il administrer les réseaux informatiques? » n'est plus d'actualité, c'est plutôt « comment administrer les réseaux informatiques » qui est devenue une des plus grandes préoccupations du monde scientifiques. C'est à la recherche de ce « comment » que le protocole SNMP a vu le jour.

Notre travail pratique sur SNMP nous a permis de découvrir ce protocole simple de surveillance et contrôle de périphériques réseaux à distance. Ce protocole existant depuis les années 1990, est implémenté sur de nombreux périphériques réseaux et peut être installé sur la plupart des systèmes d'exploitations d'ordinateurs ou des routeurs afin d'en permettre une gestion.

Le protocole SNMP offre de nombreux avantages et de nombreuses possibilités comme la gestion semi-automatique d'un parc informatique, la configuration réseau à distance, l'envoi d'alertes en cas de problèmes sur un périphérique et ainsi qu'une prise de décision manuelle (en avertissant un administrateur) ou automatique (en exécutant automatiquement une procédure de récupération sur le poste superviseur).

Bien entendu, les considérations concernant la sécurité et l'utilisation du protocole SNMP posent certains problèmes, dans la mesure où seules les dernières versions du protocole peuvent être considérées comme réellement sécurisée. Ainsi, se baser sur un nom de communauté partagé entre plusieurs machines et circulant en clair sur un réseau n'est pas une sécurité suffisante.

De plus, si on n'utilise pas de protections supplémentaires comme le filtrage des adresses IP autorisées à échanger des paquets SNMP, il est aisé pour un intrus d'envoyer des commandes distantes sur le réseau ou, tout simplement, de récupérer des informations sensibles sur la machine comme son emplacement physique ou son système d'exploitation et sa version.

Ces informations peuvent permettre d'effectuer soit des attaques physiques sur la machine, soit des attaques liées à la connaissance des failles d'une version donnée d'un système.

Le protocole SNMP doit être utilisé avec précaution et être bien exploité pour être efficace. Ce travail ne donne qu'un petit aperçu de la puissance de SNMP, même si l'application présenté lors de ce mémoire permet d'exploiter la MIB des agents dans sa totalité elle n'est en mesure d'exploiter toutes ces informations à bon escient.

Perspectives

SNMP est un protocole plein d'avenir : il se développe de plus en plus ces dernières années, parallèlement à l'essor des réseaux. La seule crainte que l'on puisse avoir est que les constructeurs, plutôt que d'adopter et de continuer à faire évoluer ce protocole devenu standard, continuent d'exploiter leurs propres protocoles, détruisant un espoir d'uniformisation de la gestion réseau.

En soi, le protocole SNMP a beaucoup d'avantages indéniables que nous avons pu mettre en avant, et les implémentations de celui-ci sont de plus en plus solides et fournissent des bases de plus en plus intéressantes aux développeurs et aux intégrateurs de systèmes

Bibliographie

1. Cours Cisco CCNA 1.
2. Thèse de Françoise Baude *:* Agents Mobiles : Itinéraires pour l'administration système et réseau.
3. Thèse de MAROUANE HIMDI: Performances des systèmes distribues :proposition d'une plateforme fédératrice de la supervision.
4. Programme de l'université de Marne la Vallée, année 2006-2007, Master réseaux
5. Mémoire de Djadel Mourad: Conception et réalisation d'une application réseau multiplateforme.
6. Mémoire de Aithem Hmida: Utilisation du protocole SNMP pour la gestion à distance d'une interface radio par paquets.
7. Travail de Aurélien Méré: La gestion réseau et le protocole SNMP
8. Mémoire de Doutoum et Oumarou: Interface de gestion SNMP d'un réseau local
9. Mémoire de Alexandro Fenyo: Etude et implémentation d'outils SNMPv2.
10. www.developpez.com
11. www.ipframe.com
12. www.snmp.com
13. www.irp.nain-t.net
14. www.commentçamarche.com
15. www.formsys.net
16. www.siteduzero.com
17. www.snmp.com
18. www.ietf.org
19. www.adventnet.net
20. www.simple-times.org
21. www.teleinfo.uqam.ca/snmp
22. www.supinfo-projects.com
23. www.indexel.net
24. www.securinfo.umontreal.ca

Annexes

ANNEXE 1: OUTILS SUPPLEMENTAIRES D'ADMINISTRATION RESEAUX

L'usage des protocoles d'administrations en particulier le SNMP à permis le développement d'un certain nombre d'outils d'administration parmi lesquels nous pouvons citer les agents RMON et les proxy-agents. A ceux là s'y ajoute les analyseurs qui peuvent être indépendants des protocoles d'administrations.

1. Agents RMON (Remonte Network Monitoring)

Un agent RMON (sonde) collecte des données relatives à un segment réseau (Ethernet, Token-Ring et plus rarement FDDI) dans la limite de sa capacité mémoire. L'agent nécessite une plate-forme d'administration qui permet de l'interroger afin de récupérer les informations et de les afficher sous forme graphique ou de les enregistrer dans une base de données afin de constituer un historique (La même plate forme est utilisée pour paramétrer et réinitialiser la sonde).

L'intérêt d'un tel outil réside dans ses fonctionnalités, en effet, certains nœuds ne disposent pas d'agents suffisamment évolués pour pouvoir surveiller efficacement les réseaux. De plus, la plupart d'entre eux ne peuvent pas être programmés pour déclencher des alarmes sur dépassement de seuil, ce qui implique un polling périodique de la part de la plate-forme d'administration. La sonde permet donc de répartir l'administration du réseau, et donc d'alléger les tâches d'une plate-forme centrale.

2. Les proxy-agents

Le principe important à retenir de cet agent RMON évolutif est qu'il permet de doter l'agent (entité) SNMP d'une certaine intelligence. Initialement l'agent RMON ne devrait répondre qu'aux demandes ou au positionnement des variables MIB émise par le manager; 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 et le proxy-agent tient un rôle de gestionnaire local, ce qui permet de diminuer la charge de travail de la station d'administration.

Le proxy-agent 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. Il permet de collecter des informations localement sous un format normalisé et avec un protocole qui peut être propriétaire, et de les retransmettre vers une station d'administration. Inversement, la station envoie des requêtes SNMP à un équipement non SNMP via le proxy-agent qui les convertit aux formats appropriés. Une requête SNMP peut même générer un échange plus complexe entre le proxy-agent et le nœud final. Un exemple de proxy-agent est celui qui assure la conversion entre SNMPv1 et SNMPv2.

Conversion requêtes SNMP par un proxy-agent

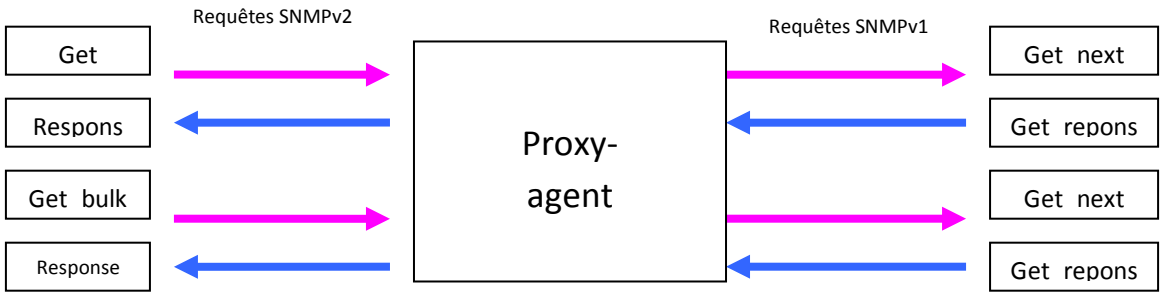


Figure 1 : exemple d'utilisation de proxy-agent

Un proxy-agent peut également surveiller un segment d'un réseau local et remonter une Trap SNMP vers la station principale sur un autre site. Ce type de montage permet de ne pas surcharger inutilement le lien à longues distances.

Une autre utilisation du proxy-agent est de détecter la plate-forme des tâches de collecte qui sollicitent les interfaces réseaux. La station ne peut analyser le trafic que sur le segment sur lequel elle est connectée, car la plupart des trames restent locales. Ce type de besoin qui a conduit à l'utilisation de la sonde RMON (Remote Monitoring), associé une MIB du même.

3. Analyseurs réseaux

Les analyseurs réseaux sont des compléments nécessaires à toute administration, ils sont autonomes et indépendants de toute plate forme. Ils sont constitués d'une carte d'acquisition du trafic à insérer dans un ordinateur portable ou non, sur lequel est installé le logiciel d'analyse.

Les fonctionnalités puissantes offertes par ces appareils les destinent à l'investigation de problèmes ponctuels. Souvent les pannes réseaux ne sont pas franches, c'est-à-dire que le réseau présente des disfonctionnements, ce qui rend la recherche de la source du problème délicate. L'analyseur doit alors être connecté sur le segment qui pose problème, des traces doivent être prélevées, et ce souvent sur plusieurs segments, de manière à remonter le problème jusqu'à sa source.

ANNEXE 2: LES RFCs EN RAPPORT AVEC SNMP

Les tableaux suivants synthétisent les différentes versions de SNMP et leurs RFC respectives :

- Version 1 – 1990

RFC	Titre de la RFC	Statut
RFC 1155	Structure and Identification of Management Information for TCP/IP-based Internets	standard
RFC 1156	Management Information Base for network management of TCP/IP-based internets	historique
RFC 1157	Simple Network Management Protocol (SNMP)	historique

- Version 2c (classique) - 1993

RFC	Titre de la RFC	Statut
RFC 1441	Introduction to version 2 of the Internet-standard Network Management Framework	historique, proposé comme standard
RFC 1442	Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)	standard proposé, remplacé par RFC-1902
RFC 1443	Textual Conventions for version 2 of the Simple Network Management Protocol (SNMPv2)	standard proposé, remplacé par RFC-1903
RFC 1444	Conformance Statements for version 2 of the Simple Network Management Protocol (SNMPv2)	standard proposé, remplacé par RFC-1904
RFC 1445	Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)	historique
RFC 1446	Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2)	historique
RFC 1447	Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2)	historique
RFC 1448	Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)	standard proposé, remplacé par RFC-1905
RFC 1449	Transport Mappings for version 2 of the Simple Network Management Protocol (SNMPv2)	standard proposé, remplacé par RFC-1906
RFC 1450	Management Information Base for version 2 of the Simple Network Management Protocol (SNMPv2)	standard proposé, remplacé par RFC-1907
RFC 1451	Manager-to-Manager Management Information Base	historique
RFC 1452	Coexistence between version 1 and version 2 of the Internet-standard Network Management Framework	standard proposé, remplacé par RFC-1908

❖ Version 2 - 1996

RFC	Titre de la RFC	Statut
RFC 1901	Introduction to Community-based SNMPv2	historique, proposé comme expérimental
RFC 1902	Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)	standard, remplacé par RFC-2578
RFC 1903	Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)	standard, remplacé par RFC-2579
RFC 1904	Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)	standard, remplacé par RFC-2580
RFC 1905	Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2) (3416)	standard, remplacé par RFC-3416
RFC 1906	Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2) (3417)	standard, remplacé par RFC-3417
RFC 1907	Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2) (3418)	standard, remplacé par RFC-3418
RFC 1908	Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework (2576)	standard, remplacé par RFC-2576

❖ Version 3 - 1999

RFC	Titre de la RFC	Statut
RFC 2571	An Architecture for Describing SNMP Management Frameworks	standard, remplacé par RFC-3411
RFC 2572	Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)	standard, remplacé par RFC-3412
RFC 2573	SNMP Applications	standard, remplacé par RFC-3413
RFC 2574	User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)	standard, remplacé par RFC-3414
RFC 2575	View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)	standard, remplacé par RFC-3415

❖ Versions 2 et 3 - 2000-2002

RFC	Titre de la RFC	Statut
RFC 2576	Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework	standard proposé
RFC 3411	An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks	standard
RFC 3412	Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)	standard

RFC 3413	Simple Network Management Protocol (SNMP) Applications	standard
RFC 3414	User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)	standard
RFC 3415	View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)	standard
RFC 3416	Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)	standard
RFC 3417	Transport Mappings for the Simple Network Management Protocol (SNMP)	standard
RFC 3418	Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)	standard

ANNEXE 3: COMMANDES CISCO ET EXEMPLES DE CONFIGURATION DE SNMP SUR LES MATERIELS CISCO

1. Configurations des communautés SNMPv2c et SNMPv3.

Syntaxe:

```
snmp-server community string [view view-name] [ro | rw] [access-list-number]
```

```
snmp-server engineID {local engineid-string | remote ip-address [udp-port port-number] engineid-string}
```

```
snmp-server group groupname {v1 | v2c | v3 {auth | noauth | priv}} [read readview] [write writeview] [notify notifyview] [access access-list]
```

```
snmp-server user username groupname [remote ip-address [udp-port port]] {v1 | v2c | v3 [encrypted] [auth {md5 | sha} auth-password [priv des56 priv-password]]} [access access-list]
```

Exemple 1:

Configuration de SNMPv2c et SNMPv3

```
! Configuration du protocole SNMPv2c
snmp-server community ciscosys1 view view1 ro 10
snmp-server community ciscosys2 view view2 rw 10

! Configuration du protocole SNMPv3

snmp-server engineID local 0123456789

!noauthNoPriv - no Authentication with no Encryption

snmp-server group ADMIN1 v3 noauth read view1 write view2 access 10
snmp-server user user1 ADMIN1 v3 access 10

! authNoPriv - Authentication with no Encryption
snmp-server group ADMIN2 v3 auth read view1 write view2 access 10
snmp-server user user2 ADMIN2 v3 auth md5 Cisco1@2012 access 10

! authPriv - Authentication and Encryption
snmp-server group ADMIN3 v3 priv read view1 write view2 access 10
snmp-server user user3 ADMIN3 v3 auth md5 Cisco2@2012 priv des Cisco3@2012 access 10

! ACL administrateurs
access-list 10 permit host 192.168.1.106
! Activer les traps
snmp-server host 192.168.1.106 traps version 2c ciscosys4
snmp-server enable traps
```

2. **Exemple de Configurations des vues sur un routeur et un Switch Cisco.**

Exemple 2:

Configuration des Views – Routeur 2801

```
snmp-server view view1 system included
snmp-server view view1 interfaces included
snmp-server view view1 ip included
snmp-server view view1 snmp included
snmp-server view view1 ospf included
snmp-server view view1 frame_relay included
snmp-server view view1 ciscoEigrpMIB included

snmp-server view view2 system included
snmp-server view view2 internet.2.1.1.5.0 excluded
```

Exemple 3:

Configuration des Views – Catalyst 2960

```
snmp-server view view1 system included
snmp-server view view1 interfaces included
snmp-server view view1 ciscoMgmt.82 included ! ciscoSTPExtensionsMIB
snmp-server view view1 ciscoVlanIfTableRelationshipMIB included
snmp-server view view1 ciscoMgmt.173 included ! ciscoPrivateVlanMIB
snmp-server view view1 ciscoIfExtensionMIB included
snmp-server view view1 ciscoMgmt.315 included ! ciscoPortSecurityMIB
snmp-server view view1 ciscoMgmt.362 included ! ciscoStormControlMIB
snmp-server view view1 ciscoMgmt.380 included ! CiscoDhcpSnoopingMIB

snmp-server view view2 system included
snmp-server view view2 internet.2.1.1.5.0 excluded
```

3. **Configurations des Traps sur un routeur et/ou un Switch Cisco.**

Syntaxe:

```
snmp-server host host-address [traps | informs] [version {1 | 2c | 3
[auth | noauth | priv]}] community-string [udp-port port]
[notification-type]
```

```
snmp-server enable traps notification-types
```

```
snmp-server trap-source interface-id
```

```
snmp-server trap-timeout seconds
```

Exemple 4:

Configuration des Traps – Routeur 2801

```
snmp-server enable traps snmp authentication linkdown linkup coldstart
warmstart

snmp-server enable traps eigrp

snmp-server enable traps flash insertion removal

snmp-server enable traps envmon

snmp-server enable traps config-copy

snmp-server enable traps config

snmp-server enable traps frame-relay

snmp-server enable traps frame-relay subif

snmp-server enable traps ospf [state-change | state-change | errors |
retransmit | lsa]
```

Exemple 5:

Configuration des Traps – Catalyst 2960

```
Switch(config)# snmp-server enable traps config

! Génère des traps suite aux changement de configuration SNMP

Switch(config)# snmp-server enable traps vlancreate

! Génère des traps suite à la création de VLANs.

Switch(config)# snmp-server enable traps vlandelete

! Génère des traps suite à la   supression de VLANs.

Switch(config)# snmp-server enable traps vtp

! Génère des traps suite au changement de protocole VTP (VLAN
Trunking Protocol).
```

4. Commandes de débogage et de vérification

a) Commandes de débogage:

debug snmp detail	Détail de SNMP
debug snmp headers	Entêtes des paquets SNMP
debug snmp mib	Commandes MIB
debug snmp packets	Paquets SNMP
debug snmp requests	Requêtes SNMP
debug snmp sessions	Sessions SNMP

b) Commandes de vérification:

show snmp	Affiche l'état de SNMP
show snmp community	Affiche les chaînes de communautés de SNMP
show snmp engineID	Affiche SNMP engineID.
show snmp group	Affiche les groupes SNMP.
show snmp host	Affiche des informatons concernant la configuration des hôtes SNMP.
show snmp session	Affiches les sessions SNMP.
show snmp source-interface	Affiche des informations concernant les interfaces sources.
show snmp trap	Affiche les notifications SNMP notifications activées et désactivées.
show snmp user	Affiche les utilisateurs SNMPv3.
show snmp views	Affiche les vues SNMP (views).
show snmp objects	Affiche les objets de la MIB.
show snmp pending	Affiche les requêtes SNMP en attente (pending requests)
show snmp sessions	Affiche les sessions SNMP
show snmp stats	Affiche des statistiques SNMP