La gestion réseau et le protocole SNMP

Aurélien Méré
FIIFO4

Table des matières

Première partie : Les réseaux

Qu'est-ce qu'un réseau

La notion de réseau a fortement évolué au fil des années, et qualifie aujourd'hui, dans son sens le plus strict, un ensemble ordonné d'objets ou de personnes ayant tous une même tâche bien définie. On parle ainsi de réseaux d'espionnage, de réseaux ferrés, de réseaux informatiques, etc…

Le fait que ces réseaux soient ordonnés nous permet de considérer aisément ceux-ci d'un point de vue algorithmique, et on les représente typiquement sous forme de graphes. Chaque objet ou personne constituant ce réseau est alors un nœud du graphe.

Les réseaux informatiques

L'objet de notre étude portant sur l'informatique, nous nous intéresserons spécialement aux réseaux informatiques dans lesquels chaque nœud a pour tâche le traitement d'informations.

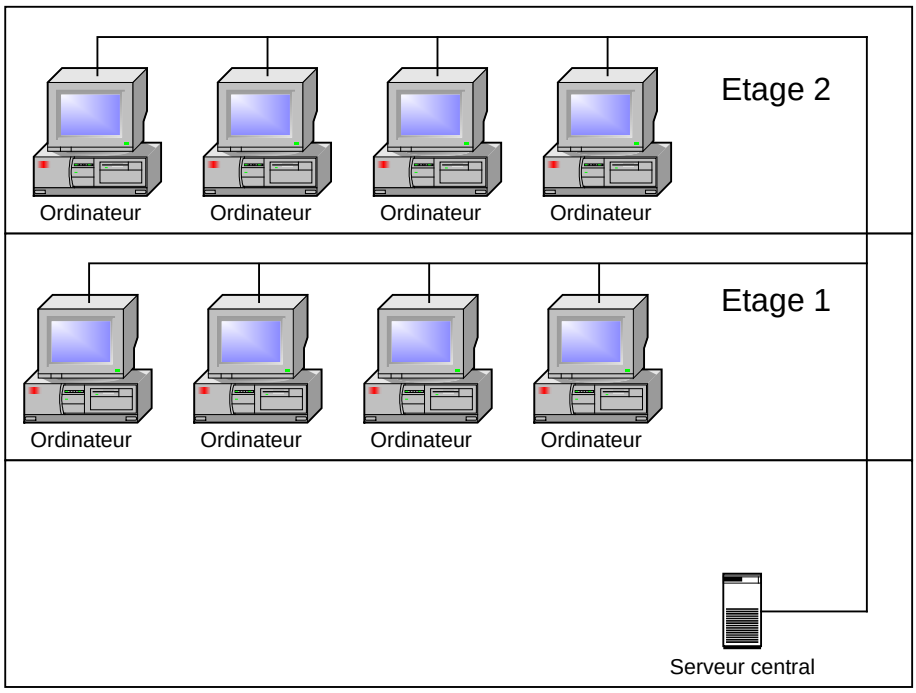

Fig.1 : Exemple de réseau informatique

Dans la représentation graphique d'un réseau informatique, un nœud peut parfois représenter un réseau en lui-même : on parle alors d'un sous-réseau de ce réseau, et on le représente par un nuage. On obtient ainsi plusieurs niveaux de représentation des réseaux, chacun plus étendu que le précédent. Les deux schémas suivants présentent ces différents niveaux d'abstraction : le premier montre le réseau dans son ensemble, et le second le montre totalement déroulé.

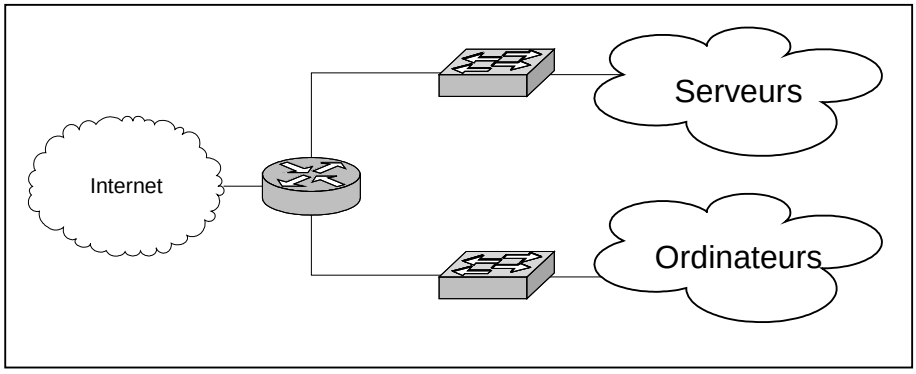

Fig.2 : Epine dorsale du réseau informatique

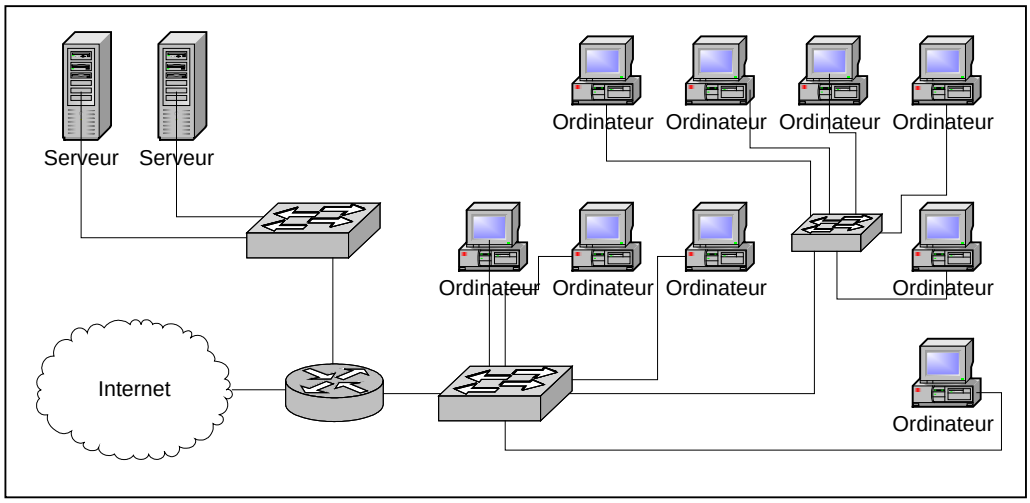

Fig.3 : Le même réseau au niveau de représentation le plus complet

Comme nous pouvons le constater sur ces précédents schémas, les réseaux informatiques sont souvent constitués par ce qu'on appelle une épine dorsale, constituant le cœur du réseau et représentant les éléments essentiels à son fonctionnement (figure 1). Il s'agit généralement de matériel réseau : les routeurs, qui représentent les ponts entre les différents réseaux logiques, et les concentrateurs qui permettent à plusieurs nœuds, tel un groupe d'ordinateurs, d'être interconnectés.

L'évolution permanente des réseaux

Depuis la création des premiers réseaux de communication informatisée dans les années 70 aux Etats-Unis, ceux-ci ont énormément évolué. Premièrement, les matériels d'infrastructures réseaux (routeurs, concentrateurs, modems, etc..) ont des capacités de traitement de plus en plus importantes et offrent des fonctions de plus en plus complexes : qualité de service, réseaux virtuels, sécurisation des communications, débit accru, etc…

Par ailleurs, outre l'évolution technique, l'évolution la plus flagrante est l'importance des réseaux informatiques et le nombre de nœuds interconnectés. Il suffit de prendre l'exemple d'Internet : alors qu'il était réservé aux militaires dans un premier temps, ce réseau s'est ouvert aux universités dans les années 70 puis aux commerciaux dans les années 80. Aujourd'hui, de plus en plus de produits ont un support de connexion à Internet.

Ordinateurs, téléphones, assistants personnels, tous ces outils sont aujourd'hui reliés en réseau et accroissent l'importance des réseaux de télécommunications mondiaux. Alors qu'au début des années 90 Internet était constitué d'environ 400.000 nœuds, il y en aurait plus de 40.000.000 aujourd'hui.

La gestion réseau

L'évolution des réseaux dont nous venons de parler pose un problème majeur : la manière dont ceux-ci sont gérés pratiquement. En effet, le nombre de nœuds ne cessant de s'étendre, il est important de pouvoir gérer tout ceux-ci le plus facilement possible, mais en conservant certaines contraintes techniques.

Nous l'avons dit précédemment, le matériel est de plus en plus sophistiqué et permet d'être contrôlé à distance : c'est là un des points fondamentaux de la gestion réseau, il est aujourd'hui nécessaire, étant donnée l'étendue des réseaux, de pouvoir le gérer à distance depuis son poste de travail et n'avoir à se déplacer qu'en dernier recours, lorsqu'une opération physique est nécessaire. Cela peut sembler exagéré, voire être perçu comme un luxe pour l'administrateur réseau, toutefois lorsque l'on a ne serait-ce qu'une centaine de nœuds à sa charge, il est important de pouvoir agir à distance dessus.

La centralisation de la gestion réseau permet également d'effectuer simultanément une même opération sur plusieurs matériels : si l'on souhaite appliquer une modification à un ensemble de matériels données, il suffit d'utiliser la liste de ce matériel dans notre programme et de l'exécuter depuis la machine de gestion. En l'absence de central, il aurait fallu se déplacer sur chaque matériel afin d'y appliquer ladite modification : la centralisation offre un gain de temps considérable.

L'évolution des réseaux entraîne une contrainte intéressante aux protocoles de gestion : ceux-ci doivent être évolutifs et prendre en compte uniformément les évolutions des différents matériels. Dans un souci d'efficacité, il est important que ses évolutions complètent les capacités précédentes du matériel, afin d'éviter tout alourdissement de sa gestion, et pour éviter des surcoûts de recherche ou d'apprentissage pour les administrateurs et les programmeurs.

Une autre contrainte pour l'administrateur réseau est la fiabilité des transactions : il est essentiel que lorsqu'une commande est envoyée à un matériel, celui-ci approuve cette modification afin que l'administrateur soit certain que la commande ait été exécutée correctement. Dans le cas contraire, la commande doit être réexpédiée jusqu'à ce qu'elle ait été acceptée par le matériel. Un état incohérent (le matériel n'a pas reçu la commande par exemple) pourrait conduire à des problèmes de fonctionnement du réseau. Par ailleurs, par le terme de « fiabilité », nous considérons également le fait que le matériel géré doit pouvoir répondre immédiatement à toutes nos requêtes et être constamment disponible.

La diversité du matériel présent dans les réseaux a imposé une nouvelle contrainte pour la gestion : l'homogénéisation des moyens d'administration. Il est impératif que les matériels se paramètrent de manière similaire afin d'éviter l'apprentissage de toutes les interfaces spécifiques à chaque constructeur et parfois à chaque modèle. Cette contrainte est malheureusement peu respectée et la solution à celle-ci consiste à acheter du matériel de même marque et/ou de même modèle suivant les besoins.

Enfin, la dernière contrainte de l'administrateur réseau est la sécurité : le matériel d'infrastructure réseau est généralement protégé au maximum car perdre l'accès à celui-ci implique généralement une interruption prolongée du service, ce qui, dans les réseaux courants, n'est pas tolérable. Ainsi, les transactions entre l'administrateur et le matériel doivent être sécurisées au maximum afin d'éviter toute tentative de détournement des droits d'accès au matériel.

Les applications de la gestion réseau

La gestion réseau est la tâche quotidienne de tout administrateur de réseau : il s'agit de vérifier le fonctionnement optimal de chacun du matériel, de paramétrer celui-ci, de le mettre à jour régulièrement, etc… Ainsi, dans les réseaux informatiques d'aujourd'hui, la gestion (ou *management*) est un problème de tous les jours.

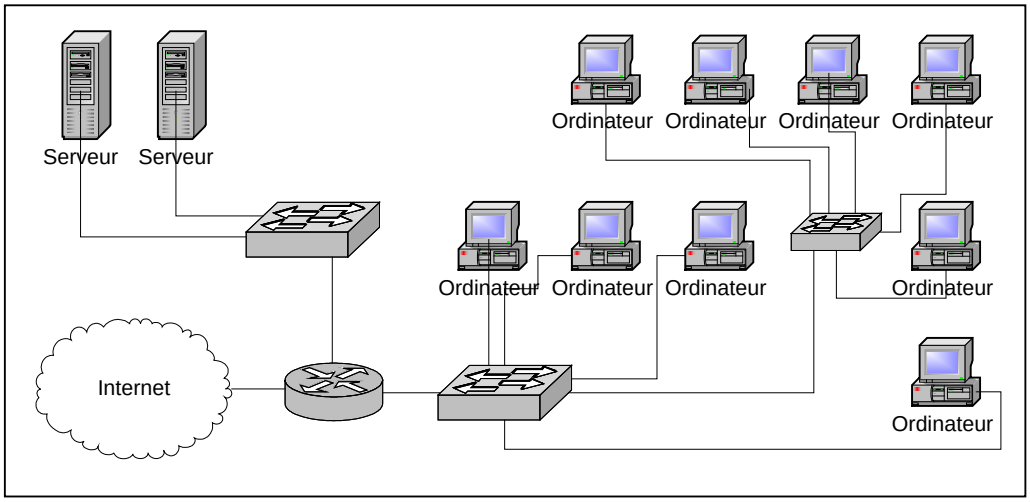

Fig.4 : Un réseau d'entreprise simplifié

Si nous reprenons le réseau présenté précédemment, la tâche de l'administrateur peut par exemple être d'interdire à l'ordinateur en haut à gauche de communiquer avec celui en bas à droite ; il peut s'agir d'interdire ou d'autoriser à tel ou tel ordinateur l'accès à Internet ; il peut s'agir de distinguer deux groupes de travail différents et de créer deux « réseaux virtuels », ceux-ci ne pouvant communiquer entre eux ; il peut s'agir de vérifier qu'il n'y a pas d'erreur de transmission sur les différents câbles réseaux ; etc… le nombre de possibilités applicatives de la gestion réseau est incalculable, sur un simple réseau d'entreprise.

Les réseaux ayant tendance à évoluer, nous approchons de plus en plus vers une « électronisation » et une informatisation du quotidien. Le futur nous réserve certainement quelques surprises, mais nous pouvons dors et déjà faire une projection de la gestion réseau dans une maison futuriste. L'alliance de la domotique et des réseaux informatiques provoquera certainement l'extension ultime de ces réseaux.

Nous pourrons alors considérer que chaque appareil électrique est un élément du réseau, relié à un concentrateur qui réagit aux différentes commandes d'un ordinateur de contrôle central. On obtient ainsi un système entièrement informatisé et géré par le réseau :

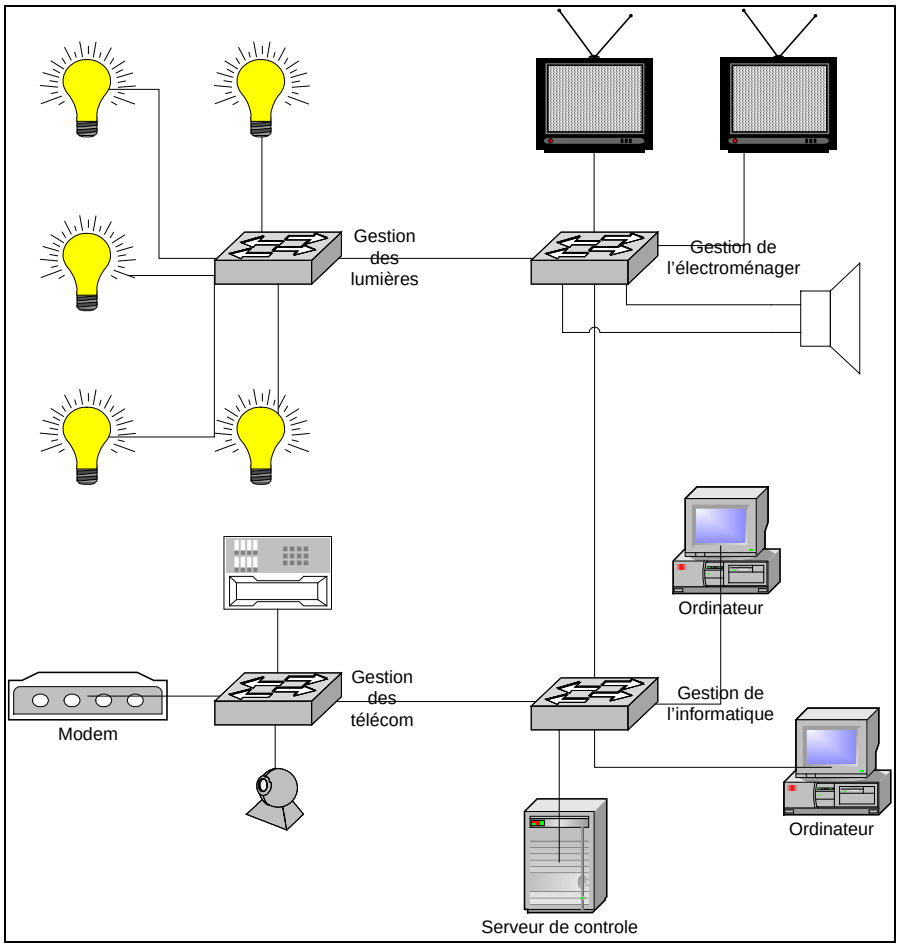

Fig.5 : Les maisons de demain

En couplant les technologies actuelles de gestion réseau permettant de contrôler à distance les appareils, il suffira ainsi d'un simple clic pour éteindre les lumières quelque part dans la maison ou pour changer de chaîne sur sa télévision. Quel intérêt me direz-vous : certes, dans la mesure actuelle des choses l'intérêt est très limité. Toutefois, si l'on couple ceci avec la technologie de la reconnaissance vocale, qui s'améliore sans cesse, il suffira alors de dire « lumière » pour que la lumière bascule de l'état allumé à l'état éteint et inversement. L'informatique et les réseaux ne sont pas forcément nécessaires, toutefois la gestion s'en trouve centralisée et de ce fait particulièrement simplifiée. Enfin, les possibilités d'évolution ne s'en retrouvent limitées que par notre imagination.

L'intérêt de la gestion réseau dans ce schéma est également accru par la diversité du matériel géré : il peut aussi bien s'agir d'un téléphone, que d'une ampoule ou d'une cafetière. La seule difficulté à surmonter est que chacun de ses appareils doit être doté d'une interface commune de communication : par exemple, un concentrateur d'ampoules, dont les caractéristiques seront la prise en charge d'un nombre déterminé de matériels sous-jacents, sur lesquels il sera possible de modifier l'état (allumé/éteint), éventuellement la puissance fournie (lumière forte/diffuse/etc…). Ce concentrateur est ensuite donné d'une interface de gestion accessible à distance.

La fusion des différentes technologies fera des miracles : Si l'on couple l'ensemble à la robotique, il suffira bientôt de dire « café » pour que la cafetière soit mise en route et que notre robot domestique nous l'apporte…

Deuxième partie : Les protocoles de gestion réseau

Il existe plusieurs protocoles de gestion réseau, actuellement tous utilisés couramment. Nous nous limiterons dans cette présentation aux protocoles basés sur TCP/IP, la norme protocolaire utilisée dans les réseaux de type Internet.

Telnet, SSH

Le premier protocole historique est *Telnet*. Ce protocole TCP est largement utilisé pour le contrôle à distance de matériel réseau. La conception est excessivement simple : une fois que l'on est connecté à la machine distante, les touches tapées au clavier sont directement transmises à la machine distante et telnet nous renvoie les réponses de ladite machine. Généralement, la machine distante commence la transaction par nous demander un mot de passe d'accès, puis nous donne accès à un shell sur lequel nous pouvons lancer nos commandes.

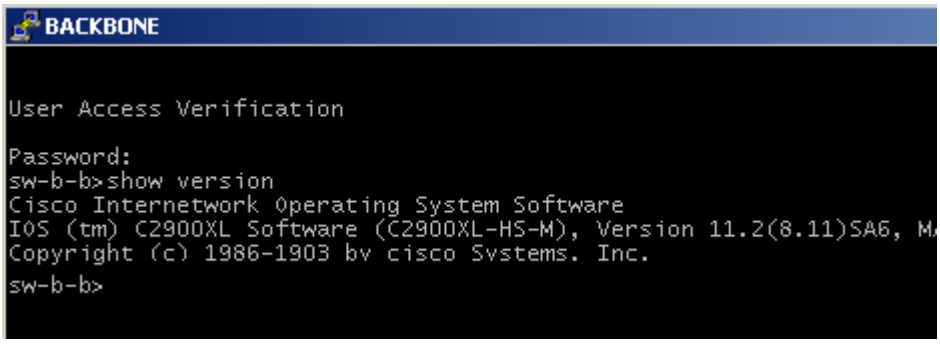

Fig.5 : Exemple de session Telnet sur un Cisco

Ci-dessus nous pouvons analyser l'exemple d'une session Telnet sur un commutateur Cisco Catalyst : celui-ci nous réclame un mot de passe d'accès, puis nous pouvons exécuter des commandes. Ici « show version » nous affiche la version du système d'exploitation du matériel.

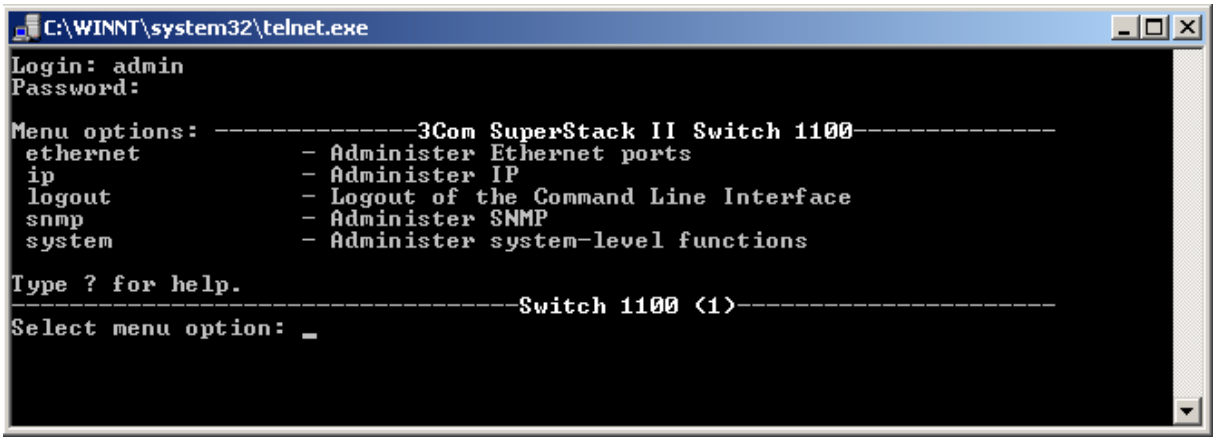

Fig.6 : Exemple de session Telnet sur un 3Com

La session Telnet sur un commutateur 3Com Superstack commence de manière similaire à la session Cisco par l'entrée d'un mot de passe. Puis nous accédons à un menu de

gestion : il faut alors choisir l'une de ces options pour naviguer à travers les différentes commandes possibles et effectuer des modifications sur le matériel.

Comme nous pouvons le constater, telnet n'est pas un protocole fournissant une interface commune à tous les matériels réseau. Chaque constructeur inclut son propre gestionnaire telnet personnalisé, et la gestion du réseau n'est donc pas uniforme suivant le type de matériel. Telnet assure intrinsèquement la fiabilité de la transaction de par l'utilisation du protocole TCP, toutefois la communication entre l'administrateur et le matériel n'est pas cryptée et la seule sécurité apportée est l'authentification initiale.

Le protocole SSH comble cette lacune en cryptant la transaction via le protocole SSL. Il permet également d'effectuer des transferts de fichiers entre les deux hôtes (protocole SCP). Toutefois l'interface reste propre à chaque matériel et ne permet pas d'effectuer des transactions parallèles ou une gestion uniforme de ceux-ci.

Interface Web

La gestion réseau par interface Web s'est développée durant ces dernières années, car celle-ci fournit une interface plus intuitive et plus agréable à utiliser que l'interface Telnet. Toutefois, à l'instar de Telnet, l'interface reste propre à chaque matériel réseau et ne permet aucune homogénéisation de la gestion.

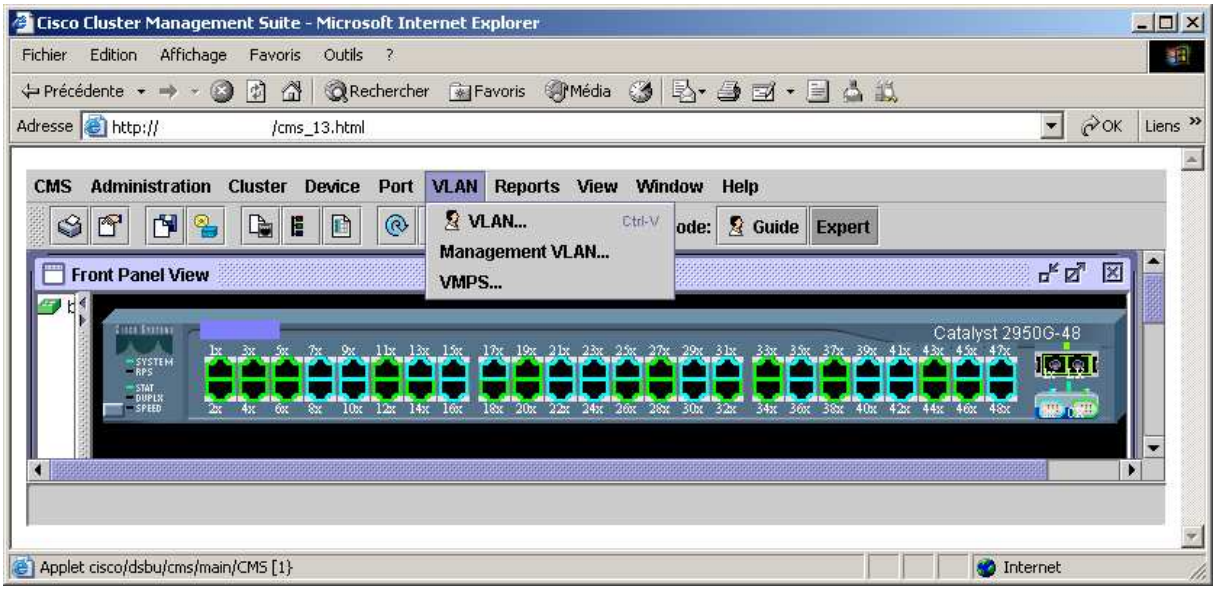

Fig.7 : Exemple de session web sur un Cisco Catalyst.

De plus, la gestion peut vite s'avérer fastidieuse s'il y a un grand nombre de nœuds au sein de notre réseau et qu'il soit nécessaire d'appliquer plusieurs modifications sur chaque. Enfin, les serveurs Web consomment souvent un important nombre de ressources sur le matériel, apportant un risque de baisse de performances.

Le protocole SNMP

Le protocole SNMP (Simple Network Management Protocol, ou autrement dit « protocole de gestion réseau simplifié »), que nous allons étudier plus en détails dans la partie suivante, a pour rôle exclusif la gestion réseau, et offre en conséquence un grand nombre d'avantages que n'ont pas les autres protocoles. SNMP propose une interface de transaction commune à tous les matériels, et donc la plus homogène possible. Son utilisation reste peu importante suite à des débuts difficiles, mais SNMP tend à devenir à terme l'une des références en matière de gestion réseau.

Comparaison des protocoles

Chaque protocole que nous avons présenté a ses propres objectifs et en conséquence ses propres avantages.

- *Telnet* est un standard en matière de communication à distance et son objectif est d'exécuter par l'intermédiaire d'une interface des commandes de gestion.
- SSH est similaire à Telnet, tout en offrant une couche supplémentaire permettant de crypter le contenu de la transaction.
- L'interface Web permet de gérer intuitivement le matériel et d'effectuer des modifications basiques de la manière la plus simplifiée possible.
- Enfin, SNMP est un protocole proposant une interface commune à tous les matériels et donc une homogénéité accrue. Nous allons étudier ce dernier plus en détail, car il respecte la plupart des contraintes que nous avons posées au début de cette synthèse.

Troisième partie : Le protocole SNMP

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 plupart des contraintes que nous avons définies dans la partie précédente, mais 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 (historique)**: 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 (expérimental)**: 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 (expérimental)**: Cette version du protocole utilise les opérations, les types de données de SNMPv2c et la sécurité basée sur les usagers. Cette version est décrite dans les RFC 1905, RFC 1906, RFC 1909 et RFC 1910.

- **SNMPv2* (expérimental)**: 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 1997 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.

Présentation

SNMP est un protocole de la famille TCP/IP (Internet protocol), et peut donc être utilisé sur tous les réseaux de type Internet. Il exploite les capacités du protocole de transport UDP :

• Chaque trame possède une adresse source et une adresse destination qui permettent aux protocoles de niveaux supérieurs comme SNMP de pouvoir adresser leurs requêtes.

• Le protocole UDP peut utiliser un checksum optionnel qui couvre l'en-tête et les données de la trame. En cas d'erreur, la trame UDP reçue est ignorée : gage de fiabilité.

• Le protocole UDP fonctionne en mode non connecté, c'est-à-dire qu'il n'existe pas de lien persistant entre la station d'administration et l'agent administré. Cela oblige les deux parties à s'assurer que leurs messages sont bien arrivés à destination, ce qui apporte également un important gage de fiabilité pour la gestion réseau.

• Deux ports sont désignés pour l'utilisation de SNMP :

- Port 161 pour les requêtes à un agent SNMP.
- Port 162 pour l'écoute des alarmes destinées à la station d'administration.

Principe de fonctionnement

Le protocole SNMP se base sur le fait qu'il existe une station de gestion réseau, le manageur, dont le rôle est de contrôler le réseau et de communiquer via ce protocole avec un agent. L'agent est de manière générale une interface SNMP embarquée sur le matériel destiné à être administré à distance.

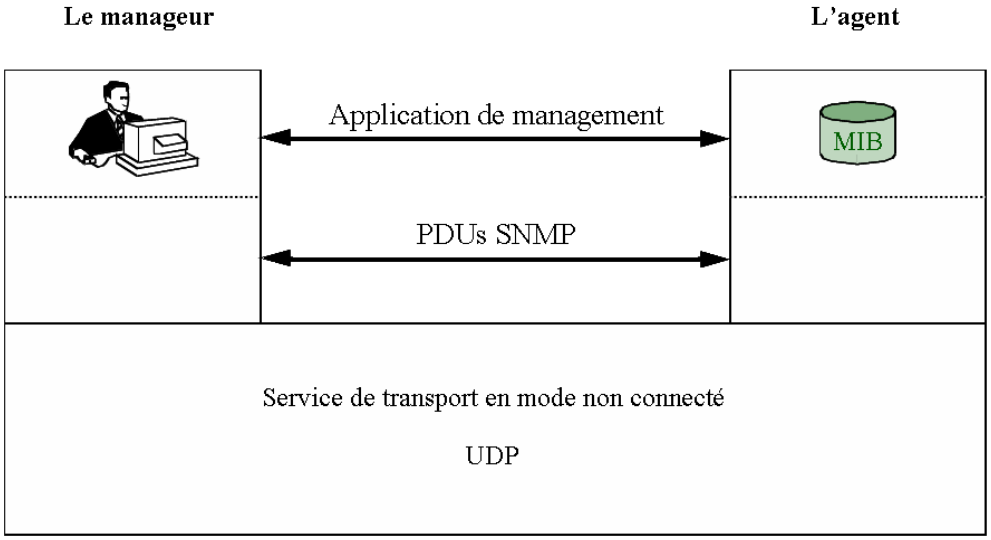

Fig.8 : Base de SNMP

Le protocole SNMP est constitué de plusieurs commandes différentes :

- Get : Cette commande, envoyée par le manageur à l'agent, a pour objectif de demander une information à l'agent. Celui-ci, dans le cas où la validité de la requête est confirmée, renvoie au manageur la valeur correspondant à l'information demandée.
- Getnext : Cette commande, envoyée par le manageur à l'agent, a pour objectif de demander l'information suivante à l'agent : il arrive qu'il soit nécessaire de parcourir toute une liste de variables de l'agent. On utilise alors cette commande, à la suite d'une requête 'get', afin d'obtenir directement le contenu de la variable suivante.
- Getbulk : Cette commande, est envoyée par la manageur à l'agent pour connaître la valeur de plusieurs variables : cela évite d'effectuer plusieurs requêtes Get en série, améliorant les performances (implémenté dans SNMPv2).

- Set : Cette commande, envoyée par le manageur à l'agent, a pour objectif de définir la valeur d'une variable de l'agent administré. Cela permet d'effectuer des modifications sur le matériel.
- Trap : Lorsqu'un événement particulier survient chez l'agent (connexion, modification de la valeur d'une variable donnée, etc…), celui-ci est susceptible d'envoyer ce que l'on appelle une « trap », à savoir un message d'information destiné à la station d'administration : celle-ci pourra alors la traiter et éventuellement agir en conséquence. S'il s'agit par exemple de la coupure d'un lien réseau, cela permet à l'administrateur réseau d'en être immédiatement informé.
- Inform : Dans certains cas, il peut être intéressant pour l'agent d'obtenir une réponse à une Trap qu'il a émise, afin d'obtenir confirmation que celle-ci a bien été reçue et analysée : c'est l'objectif d'une commande « inform ». (Implémenté dans SNMPv2).

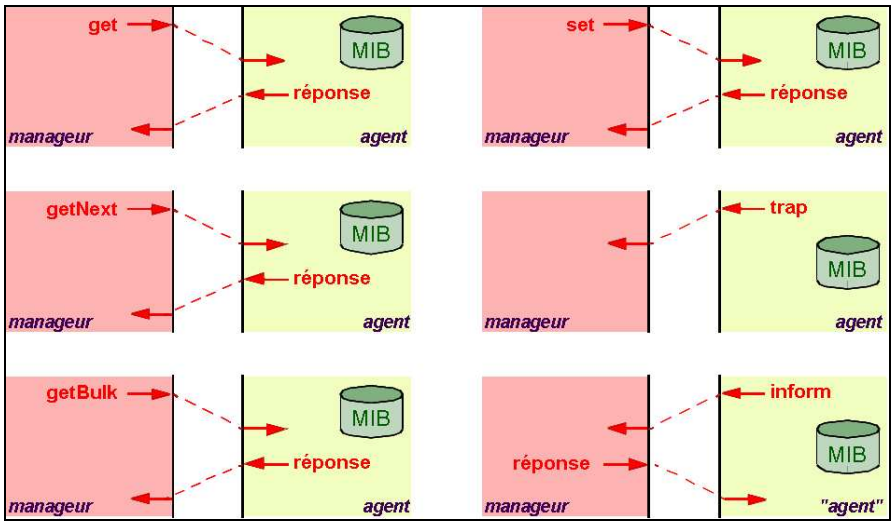

Fig.9 : Résumé des commandes SNMPv2

Les variables SNMP et le modèle SMI

L'objectif de SNMP est donc d'obtenir ou de modifier la valeur d'une ou plusieurs variables du matériel : Il peut s'agir par exemple de l'état d'une interface réseau (allumé/éteint). Les variables SNMP exploitent le modèle SMI (Structure of Management Information) qui définit un modèle hiérarchique des variables. Le modèle est défini dans les RFC1155 et 2578.

Dans ce modèle, les variables sont répertoriées dans une hiérarchie d'objets. Chaque objet est identifié par ce que l'on appelle un OID (Object IDentifier). La hiérarchie de ces objets se représente sous la forme d'un arbre. Les branches constituent les différents OIDs et les feuilles les variables. Une variable peut donc être référencée par la liste ordonnée des différents OIDs parcourus à partir de la racine de l'arbre.

Le modèle SMI définit également les types de données utilisables pour les variables : entier, réel, durée, compteur, etc…

Un ensemble d'objets d'un même module est appelé une MIB (Module Information Bases). Il s'agit d'une base de données référençant la liste des objets et des variables associées, des types de données utilisés pour chaque variable et d'un descriptif de cette variable. La base contient éventuellement des types de données personnalisés

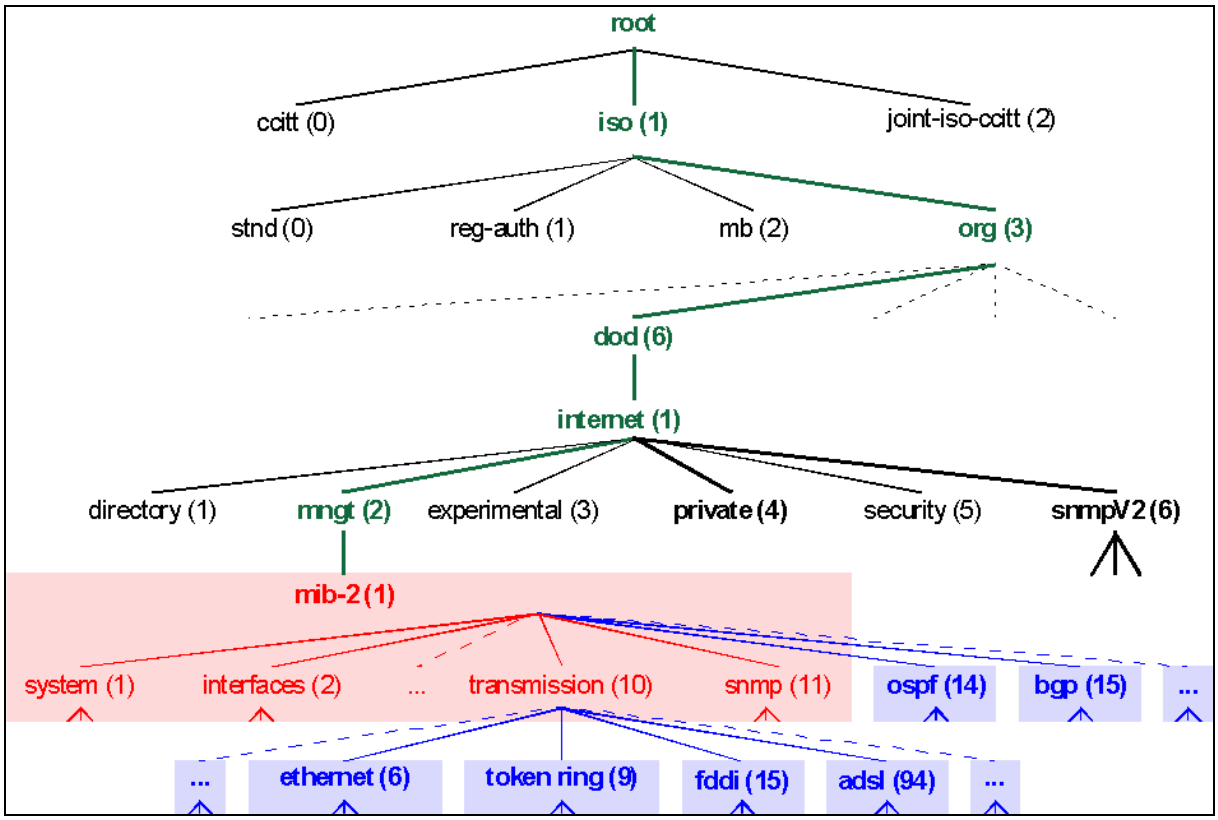

Fig.10 : L'arborescence des OIDs en SNMP

La figure 10 présente l'arborescence des OIDs, constituant les MIBs. En SNMP, on utilise communément deux branches :

- iso.org.dod.internet.mngt.mib-2 (1.3.6.1.2.1) : il s'agit de la branche contenant tous les objets standards, définis précisément dans les RFC. Ainsi, tout agent SNMP doit pouvoir reconnaître cette branche et les variables qui y sont définies.
- iso.org.dod.internet.private.enterprises (1.3.6.1.4.1) : cette branche est l'origine de tous les objets propres au matériel et définies par le constructeur. Ainsi, chaque constructeur se voit attribué un identifiant (VendorID), qui lui fournit un espace de données au sein de l'arbre des MIBs. Si nous prenons l'exemple de Cisco, dont l'identifiant est 9, toutes les variables propres à Cisco ont une clé débutant par 1.3.6.1.4.1.9.

Les fichiers MIBs

Les fichiers MIBs décrivent précisément chaque chiffre (OID) de la liste identifiant une variable (la clé), et sa signification. Prenons l'exemple simple de la variable contenant le nom du matériel interrogé. Il s'agit d'une propriété de l'objet standard « system » que nous pouvons voir dans l'arborescence de la figure 10. La propriété s'appelle « sysName ». On en déduit que la variable s'appelle alors : iso.org.dod.internet.mngt.mib-2.system.sysName.

Analysons le fichier MIB décrivant l'objet « system » (RFC 1213) :

```
sysName OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..255))
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
            "An administratively-assigned name for this
            managed node.  By convention, this is the node's
            fully-qualified domain name."
    ::= { system 5 }
```

Le fichier fournit toutes les informations relatives à la propriété « sysName » :

- Syntaxe : il s'agit d'une chaîne de caractères de taille variant entre 0 et 255.
- Accès : l'accès à cette variable se fait en lecture ou en écriture
- Etat : cette variable existe et est toujours utilisable.
- Description : il s'agit du nom complet du nœud.
- Sa place dans l'arborescence : 5e propriété de l'objet « system » : On en déduit que cette variable a pour clé la valeur 1.3.6.1.2.1.1.5.

Ainsi, nous avons la description de toutes les variables, leur méthode d'accès et la clé que nous devons utiliser pour lire ou écrire sa valeur. La majorité des constructeurs fournit des fichiers MIBs contenant des informations sur les variables propres à leur matériel, ne faisant pas partie des informations standards.

Il existe un grand nombre d'outils permettant de visualiser l'arbre des MIBs et de rechercher une variable au sein de celui-ci. L'exemple de la figure 11 est tiré du site *www.snmplink.org*.

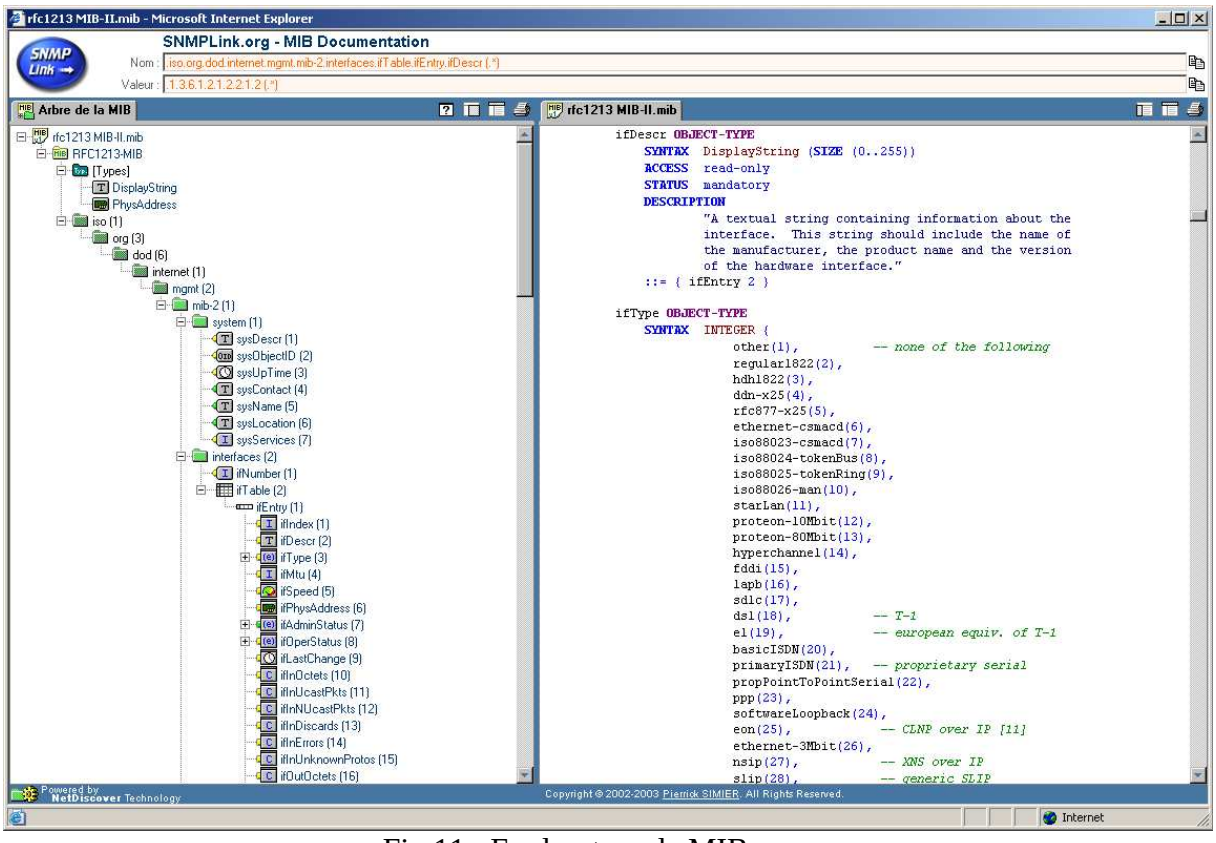

Fig.11 : Explorateur de MIBs

La sécurité

Sous SNMPv1 et SNMPv2c, la sécurité est assurée par deux choses :

- Dans sa requête, il faut envoyer une chaîne de communauté, qui correspond en quelque sorte à un mot de passe, et dont les droits varient suivant cette chaîne : il est ainsi possible d'autoriser certaines personnes un accès en lecture seule, et à d'autres personnes un accès complet suivant la communauté qu'ils utilisent.
- L'agent peut vérifier et contrôler l'origine des données, afin de vérifier que la personne en question a accès aux informations. Il s'agit généralement d'une vérification basée sur l'adresse IP source.

Nous constatons toutefois que la sécurité est particulièrement lacunaire pour deux raisons : le contenu de la transaction n'est pas crypté, et il suffit que la communauté soit connue de n'importe qui pour que cette personne puisse lire les informations.

Ces différents problèmes ont été résolus dans SNMPv3. En effet, celui-ci propose plusieurs modèles de sécurités différents :

- Le premier modèle est dépourvu de sécurités et est comparable à SNMPv1/v2c.
- Le second modèle offre des capacités d'authentification par utilisateur, c'est-à-dire que chaque utilisateur dispose d'un mot de passe d'accès, ainsi que d'une clé publique de cryptage permettant de sécuriser le contenu de la transaction.
- Le troisième modèle ajoute au précédent un niveau de cryptage supplémentaire en utilisant le principe d'échange des clés privées : le contenu des paquets est ainsi totalement crypté mais ce modèle n'est applicable qu'en fonction des lois sur la cryptographie en vigueur.

Fonctionnement pratique

Lorsqu'une commande est expédiée à un agent, on attend de celui-ci une réponse. Plusieurs cas peuvent se produire :

- Aucune réponse (Temps d'attente dépassé)
- Erreur dans la requête
- La requête a réussi.

Aucune réponse

Plusieurs cas sont susceptibles de produire une absence de réponse de la part du matériel interrogé :

- SNMP est basé sur UDP et il peut arriver que les paquets n'arrivent pas à destination. Dans ce cas, le temps d'attente de réponse finit par s'écouler et il convient alors de réémettre la requête.
- Suivant l'implémentation des agents et la version de SNMP utilisée, si l'authentification échoue (mauvaise communauté, mot de passe incorrect), l'agent peut ne pas répondre à la requête.
- Le temps d'attente de réponse peut être paramétré dynamiquement et il est possible que le temps défini soit trop court pour permettre à la réponse de revenir.

- Enfin, dans le pire des cas, il est possible qu'il n'y ait pas d'agent SNMP disponible sur le matériel interrogé. Nous ne pouvons en conséquence avoir de réponse à notre requête.

Erreur

Plusieurs cas sont susceptibles de conduire au renvoi d'une erreur :

- Lorsqu'on l'on essaie d'écrire sur une variable en lecture seule
- Lorsque l'on essaie de définir la valeur d'une variable avec un type de données incorrect (si l'on essaie d'écrire une chaîne de caractères dans une variable de type entier par exemple).
- Lorsque la variable n'existe pas.
- Lorsque la trame SNMP est incorrecte (corruption, longueur non valide…)
- Lorsque l'authentification SNMPv3 a échoué.

Réussite

Lorsque la requête à l'agent SNMP réussit, celui-ci nous envoie la valeur de la variable à laquelle on a accédé (que ce soit en lecture ou en écriture).

Les implémentations existantes du protocole SNMP

Il existe des centaines d'implémentations différentes du protocole SNMP, de par le fait qu'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.

Exemple de transaction SNMP

Procédons à l'analyse d'un matériel réseau classique, par exemple un routeur. Cherchons à connaître son nom, son temps de fonctionnement et des informations sur ses interfaces. Nous considérerons que la communauté d'accès est « TdS » et que l'adresse de ce matériel est « 172.17.67.253 ».

Reprenons tout d'abord l'exemple présenté dans une partie précédente, concernant le nom de l'hôte. Nous en avions conclu que la clé de la variable était 1.3.6.1.2.1.1.5. Effectuons une requête de type GET sur ce matériel et sur cette clé afin de voir s'il répond conformément à nos attentes :

```
$ snmpget -c TdS 172.17.67.253 .1.3.6.1.2.1.1.5
system.sysName = router-exemple
$
```

La syntaxe de la commande est très simple : on appelle *snmpget* en définissant le nom de la communauté (-c TdS). On lui fournit l'adresse IP du matériel à interroger ainsi que la clé que l'on cherche à obtenir. La commande a réussi et l'agent nous a renvoyé la valeur « router-exemple ». Le nom de ce système est donc « router-exemple ».

Nous pouvons constater que le principe de sécurité basé sur la communauté est opérationnel. Essayons d'interroger l'agent en utilisant un nom de communauté différent, par exemple « test » :

```
$ snmpget -c test 172.17.67.253 .1.3.6.1.2.1.1.5
Timeout: No Response from 172.17.67.253.
$
```

Snmpget nous signale alors qu'il n'a pas eu de réponse : la sécurité fonctionne. De même, si l'on essaye d'accéder à une variable inexistante, l'agent renvoie une erreur :

```
$ snmpget -c TdS 172.17.67.253 .1.3.6.1.99
Error in packet
Reason: (noSuchName) There is no such variable name in this MIB.
$
```

L'agent renvoie une erreur similaire lorsque l'on interroge une branche et non une feuille de l'arbre des OIDs. Essayons une requête snmpget sur l'objet « system » lui-même :

```
$ snmpget -c TdS 172.17.67.253 .1.3.6.1.2.1.1
Error in packet
Reason: (noSuchName) There is no such variable name in this MIB.
$
```

Il est nécessaire, pour parcourir toute la branche, d'effectuer une requête de type « walk », qui constitue en fait une série de requêtes get et get-next. On obtient ainsi la liste de toutes les variables de la branche :

```
$ snmpwalk -c TdS 172.17.67.253 .1.3.6.1.2.1.1
system.sysDescr = Cisco Internetwork Operating System Software
system.sysObjectID = OID: enterprises.9.1.172
system.sysUpTime = Timeticks: (9360890) 1 day, 2:00:08.90
system.sysContact = KindMan@fleming.u-psud.fr
system.sysName = router-exemple
system.sysLocation = Résidence Fleming
system.sysServices = 78
$
```

On obtient ainsi un ensemble d'informations relatives au système : sa description, le temps depuis quand il fonctionne, l'endroit où il se trouve, la personne qui s'en occupe, etc… Il faut bien sûr que ces paramètres aient été précédemment définis dans la configuration de l'agent.

Nous pouvons également modifier les informations directement depuis notre console, par l'utilisation de commandes SET. Essayons par exemple de modifier la description de l'endroit où le matériel se trouve (system.sysLocation, .1.3.6.1.2.1.1.6). Cela se fait par le biais de la commande snmpset. Il est également nécessaire de lui fournir le type de données utilisé (ici « s » pour « string, chaîne de caractères ») :

```
$ snmpset -c TdS 172.17.67.253 .1.3.6.1.2.1.1.6 s "chez moi"
system.sysLocation = chez moi
$
```

Nous pouvons vérifier que la commande a bien été prise en compte :

```
$ snmpget -c TdS 172.17.67.253 .1.3.6.1.2.1.1.6
system.sysLocation = chez moi
$
```

La MIB standard fournit des informations sur l'état des interfaces réseaux du matériel interrogé. L'OID « set iso.org.dod.internet.mgmt.mib-2.interfaces.ifNumber » nous permet tout d'abord de connaître le nombre d'interfaces :

```
$ snmpget -c TdS 172.17.67.253 .1.3.6.1.2.1.2.1.0
interfaces.ifNumber.0 = 4
```

Les interfaces sont alors regroupées dans un tableau « .interfaces.ifTable » contenant la liste des propriétés de chaque interface. Si nous désirons par exemple connaître le type des interfaces, il nous faudra interroger l'OID « interfaces.ifTable.ifEntry.ifType » :

```
$ snmpwalk -c TdS 172.17.67.253 .1.3.6.1.2.1.2.2.1.3
interfaces.ifTable.ifEntry.ifType.1 = ethernetCsmacd(6)
interfaces.ifTable.ifEntry.ifType.2 = ethernetCsmacd(6)
interfaces.ifTable.ifEntry.ifType.3 = propPointToPointSerial(22)
interfaces.ifTable.ifEntry.ifType.4 = other(1)
```

On constate que l'agent a ajouté à l'OID le numéro de l'interface : nous avions vu qu'il y en avait 4, aussi le tableau est indexé de 1 à 4. Nous constatons que les interfaces n°1 et n°2 sont de types « ethernet », alors que celle de type n°3 est de type « pppSerial ». Enfin la 4e interface est de type inconnue (ou de type défini par le constructeur).

Nous pouvons connaître l'état de chacune des interfaces en interrogeant la propriété ifOperStatus (état courant de l'interface, ix=8) de l'objet ifEntry :

```
$ snmpwalk -c TdS 172.17.67.253 .1.3.6.1.2.1.2.2.1.8
interfaces.ifTable.ifEntry.ifOperStatus.1 = up(1)
interfaces.ifTable.ifEntry.ifOperStatus.2 = up(1)
interfaces.ifTable.ifEntry.ifOperStatus.3 = down(2)
interfaces.ifTable.ifEntry.ifOperStatus.4 = up(1)
```

Le programme nous apprend ainsi que les interfaces n°1, 2 et 4 sont actives, mais que l'interface n°3 est inactive. Il nous serait possible, compte tenu de ces informations, de désactiver l'une des 3 interfaces actives en définissant la valeur via une commande de type set.

Il existe plusieurs centaines de milliers d'informations récupérables via SNMP, en raison de la quantité impressionnante du nombre d'objets. De plus, ceux-ci sont en constante évolution, aussi ce nombre ne cesse d'augmenter, chaque constructeur ajoutant ses propres spécificités.

Par ailleurs, pour montrer que l'interface de communication est réellement standard et fonctionne avec n'importe quel agent SNMP, nous allons essayer de communiquer avec une station Windows et de s'assurer que le résultat est similaire :

```
$ snmpwalk -c public 172.17.64.28 .1.3.6.1.2.1.1
system.sysDescr = Windows 2000 Version 5.0 (Build 2195)
system.sysObjectID.0 = OID: enterprises.311.1.1.3.1.2
system.sysUpTime.0 = Timeticks: (7541868) 20:56:58.68
system.sysContact.0 =
system.sysName.0 = SERVEUR
system.sysLocation.0 =
system.sysServices.0 = 76
```

La chaîne d'informations qualifie cet agent de « Windows 2000 ». Nous constatons que cet agent fournit beaucoup moins d'informations que le précédent, toutefois même si les champs sont vides, les variables existent, ce qui nous conforte dans l'idée que SNMP est un protocole gérant tout type de matériel réseau.

```
$ snmpget -c public 172.17.64.28 .1.3.6.1.2.1.2.1
interfaces.ifNumber = 3
```

Nous constatons que cette machine a 3 interfaces réseau : il y a bien un respect de la norme, et nous pouvons interroger indépendamment un serveur Windows 2000 ou un matériel de gestion réseau central, sachant que les résultats sont similaires ;

Avantages et inconvénients

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 équipement 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.

Avenir de SNMP

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.