Faculté des Sciences et Techniques de Nantes
Maîtrise d'informatique 1999/2000

Travaux d'Etude et de Recherche

Conception d'un outil d'administration réseau

(Projet NaONeT)

CABOTSE Aurélien
COUSIN Marc
HARFOUCHE Karim
HARFOUCHE Mehdi

Encadrement : Christian ATTIOGBE

Table des matières

Introduction **4**
Etat du projet . . . 4
Projet A . . . 4
Projet B . . . 5

1 Cahier des Charges **6**
1.1 Contexte . . . 6
1.2 Les modes d'apprentissage . . . 6
1.3 Carte du réseau . . . 6
1.3.1 Visualisation . . . 6
1.3.2 Construction graphique d'un réseau . . . 7
1.3.3 Administration de réseaux . . . 7
1.4 Affichage de données . . . 7
1.5 Surveillance . . . 7
1.6 Compilateur de MIB . . . 7

2 Généralités **8**
2.1 Les réseaux . . . 8
2.1.1 Types de réseaux . . . 9
2.1.2 Les normes . . . 9
2.1.3 Le modèle ISO . . . 10
2.1.4 Topologie des réseaux . . . 11
2.1.5 Les méthodes d'accès . . . 14
2.2 L'interconnexion . . . 15
2.2.1 Les moyens mis à disposition . . . 15
2.2.2 Les équipements d'interconnexion . . . 16
2.3 Les Clients/Serveurs . . . 19
2.4 Les sockets . . . 20
2.5 Les protocoles . . . 21
2.5.1 IP . . . 21
2.5.2 TCP . . . 22
2.5.3 UDP . . . 22

3 Administration de réseaux 24
3.1 Le protocole SNMP (Simple Network Management Protocol) . . . 24
3.2 Le protocole SNMPV2 . . . 25
3.3 Les RFCs (Request For Comments) . . . 26
3.4 La syntaxe ASN.1 . . . 26
3.5 Le modèle ISO . . . 27
3.6 La MIB . . . 27
3.7 La MIB RMON . . . 28

4 L'outil d'administration 29
4.1 L'administration d'interconnexion réseau . . . 29
4.2 Modes d'apprentissage . . . 30
4.2.1 Le mode Virtuel local . . . 31
4.2.2 Le mode Virtuel distant . . . 31
4.2.3 Le mode Physique . . . 31
4.3 Interface graphique . . . 31
4.3.1 Analyse . . . 33
4.3.2 L'ajout et la modification d'éléments (Réseaux, Machines) . . . 34
4.3.3 L'interconnexion de réseaux . . . 34
4.3.4 Mise en place de la structure de données . . . 36
4.3.5 Spécification . . . 36
4.3.6 Conception . . . 39
4.3.7 Gestion des listes . . . 40
4.3.8 Gestion des événements . . . 40
4.3.9 Chargement/Sauvegarde . . . 41
4.4 L'interrogation . . . 41
4.4.1 Surveillance . . . 41
4.4.2 Création et affectation d'une MIB . . . 42
4.4.3 Diversité des types d'agents . . . 42

5 Evolutions 43
5.1 Algorithme de découverte . . . 43
5.2 Utilisation des données SNMP . . . 43
5.3 Agents intelligents . . . 44

Conclusion 45

Glossaire 46

Références 47

Table des figures

2.1 Modèle ISO . . . 11
2.2 Topologie en Bus . . . 12
2.3 Topologie en Etoile . . . 12
2.4 Topologie en Anneau . . . 13
2.5 Le répéteur . . . 16
2.6 Le pont . . . 17
2.7 Le routeur . . . 18
2.8 La passerelle . . . 18
2.9 Le modèle centralisé . . . 19
2.10 Le modèle serveur de fichiers . . . 20
2.11 Le modèle Client/Serveur . . . 20
2.12 Couches des protocoles, Voir [Rose 95] . . . 22

3.1 Arbre des OID . . . 26

4.1 Concepts . . . 30
4.2 Hiérarchie des classes . . . 32
4.3 Première possibilité . . . 33
4.4 Seconde possibilité . . . 33
4.5 Interconnexion de réseaux . . . 34
4.6 Un exemple d'interconnexion . . . 34
4.7 Un exemple d'utilisation . . . 35
4.8 Diagramme UML . . . 39

Introduction

Nous présentons dans ce rapport, le travail effectué dans le cadre du module TER (Travaux d'Etude et de Recherche) de la maîtrise d'informatique. Le sujet de notre étude est la contribution au développement d'un outil logiciel pour l'administration de réseaux interconnectés. Le développement de cet outil se fait dans le cadre du projet NaONeT (Nantes Open Network Tools) piloté par Mr Christian Attiogbé

Cet outil a un but pédagogique et expérimental. Avec la généralisation des réseaux et de l'internet, la diversité des périphériques est devenue de plus en plus grande. Il est devenu inconcevable de ne pouvoir gérer qu'un type de réseau (Unix, Windows NT, Linux ...) et l'interconnexion des réseaux et des protocoles est devenu un problème majeur si l'on veut assurer la sécurité, la configuration et les performances. L'outil d'administration doit être simple à utiliser car il sera employé par des étudiants. Il doit être modulaire et ergonomique afin de fournir une aide à l'apprentissage et ainsi éviter des opérations néfastes pour le réseau. Dans un cadre expérimental, l'outil doit assurer des fonctions d'administration ; l'administrateur doit pouvoir avoir rapidement une vue globale de l'état du réseau et pouvoir intervenir sur les différents éléments du réseau.
Nous allons, tout d'abord, décrire l'état dans lequel se trouvait le projet lors de sa reprise.

Etat du projet (1999)

Pour réaliser notre projet, nous avons dû examiner et comprendre le travail effectué par deux groupes de Maîtrise dans le cadre du T.E.R. de 1998/1999. Ce projet ayant débuté en septembre 1997.

Projet A [TER M. CHATTON, D. MAURANGE, X. PAQUET]

L'outil qui a été réalisé est capable de générer une vue MIB reflétant l'état d'un système, d'utiliser des requêtes SNMP sur cette vue et de donner les résultats dans une interface graphique. Cette application se compose de quatre grandes parties :

La vue MIB

La réalisation d'une copie de MIB nécessite un fichier tampon, contenant les différents variables du système, et un RFC, répertoriant le type, l'accès et le statut de chacune. La mise en relation de ces deux fichiers permet d'obtenir la vue MIB contenant les variables et leurs caractéristiques.

Les requêtes SNMP

Le projet 1999 ayant pour but de réaliser un outil pédagogique d'administation de réseaux, il était nécessaire de réaliser un "remplacant" au démon snmpd. Ce pseudo-démon permet de réaliser des modifications sur la vue MIB, de la même façon que sur une MIB réelle, mais sans affecter de quelques manières que ce soit les registres internes des agents.

La communication entre les entités

Le système réalisé s'appuie sur le principe client-serveur. Il utilise le protocole SNMP qui impose un fonctionnement des communications en mode non connecté. Le protocole UDP a donc été retenu pour la communication entre les entités, car il est beaucoup plus fiable que le protocole TCP dans le cas d'un réseau surchargé. Or, c'est dans cette situation que la gestion du réseau est importante. Des sockets Unix sont chargées de la circulation des informations : une socket à l'écoute des clients possibles, et un processus fils pour la communication avec l'un d'eux ; soit un processus par client communicant.

L'interface graphique

L'interface réalisée permet de sélectionner un agent. Une fois cette étape efféctuée, il est possible d'envoyer des requêtes à un agent. Une fenêtre permet alors la visualisation de la réception des données ou traps. Nous avons réutilisé une grande partie de cette interface, en la réintégrant dans la notre.

Projet B

Dans le même contexte que le projet précédent, celui-ci se base sur le protocole SNMP et son démon sous Linux pour servir d'agent.

Les statistiques du commutateur

Selon le type de l'agent, les fonctionnalités diffèrent et il est possible dans le cas du switch, d'obtenir de la MIBRMON un maximun de statisques sous forme de graphiques. Ce qui permet de visualiser en temps réel la charge du réseau.

L'interface graphique

Comme dans l'interface du projet précédent, il est possible de sélectionner l'agent désiré. Les requêtes SNMP sont construites manuellement et lui sont envoyées. De même, chaque variable est expliquée en reprennant le RFC de la MIB. Il est possible d'effectuer une recherche dans l'arbre des variables.

Nous avons eu à charge de poursuivre les projets A et B en concevant un outil graphique intégrant les résultats A et B.

Chapitre 1

Cahier des Charges

1.1 Contexte

L'outil en vue doit :
- $[CO1]$: pouvoir être utilisé dans les salles de T.P sur des réseaux de taille moyenne,
- $[CO2]$: pouvoir convenir aux étudiants apprennant l'administration réseau ainsi qu'à toute personne désirant administrer son propre système,
- $[CO3]$: pouvoir faciliter l'apprentissage des nombreuses notions entourant l'administration grâce à un outil graphique.

1.2 Les modes d'apprentissage

L'outil doit permettre de :
- $[MV]$: pouvoir travailler sur une seule machine en imitant des conditions réelles,
- $[MVD]$: pouvoir travailler avec des machines distantes mais sur des copies de MIB,
- $[MP]$: pouvoir travailler en conditions réelles avec des machines distantes.

1.3 Carte du réseau

L'outil doit être doté d'une bonne interface graphique.

1.3.1 Visualisation

L'outil doit :
- $[CVD]$: pouvoir dessiner (construire) automatiquement ou manuellement une carte de réseau.
- $[CVS]$: offrir la possibilité de sauver dans un fichier une structure construite,
- $[CVC]$: offrir la possibilité de charger un fichier.

1.3.2 Construction graphique d'un réseau

Fonctionnalités attendues :
- [CCA] : possibilité d'ajouter un hôte à gérer,
- [CCS] : possibilité de supprimer un hôte,
- [CCM] : possibilité de modifier l'adresse et le nom de l'hôte.

1.3.3 Administration de réseaux

Fonctionnalités attendues :
- [CAI] : interroger la MIB et récupérer les valeurs des variables,
- [CAM] : changer la valeur d'une ou plusieurs variable(s),
- [CAS] : récupération des traps générées par l'agent.

1.4 Affichage de données

- [AG] : affichage des données sous forme de graphiques.

1.5 Surveillance

- [SNT] : créer de nouvelles traps sur l'hôte distant,
- [SMT] : modifier un seuil sur un hote distant.

1.6 Compilateur de MIB

- [CDM] : créer une nouvelle MIB utilisable grâce aux RFC.

Une grandee partie des besoins, exprimés dans ce cahier des charges, ont été entamés dans les précédents travaux (A et B), nous avons essentiellement à charge l'intégration de ces projets à travers une interface graphique. La partie Construction graphique de réseaux est une nouvelle fonctionnalité.

Chapitre 2

Généralités

Pour envoyer un matériel à l'autre bout du monde, vous allez emballer ce matériel, y écrire l'adresse du destinataire, le confier à un transporteur qui lui même le donnera peut être à un transporteur aérien ...

De la même manière, pour transférer une information à un destinataire distant, il faut formater cette information pour la rendre compréhensible, préciser l'adresse du destinataire, établir le chemin de transmission... C'est l'ensemble de ces tâches que les techniques réseaux s'efforcent de solutionner, à partir, notamment, de normes de communication établies par les différents acteurs du monde des télécoms. Un réseau est un ensemble de connexions entre plusieurs ordinateurs. Il permet à différentes machines d'accéder en commun, à la plupart des ressources, aussi efficacement que dans le cadre d'un système centralisé. Dans le cas où la connexion se réduit à un simple câble reliant des ordinateurs sur un même site, nous parlons de réseau local. Un réseau longue distance englobe plusieurs sites interconnectés par des liaisons plus élaborées, telles que les lignes téléphoniques spécialisées ou les satellites.

2.1 Les réseaux

Les réseaux informatiques ont eu comme première fonction de permettre l'accès à des ordinateurs distants par l'intermédiaire de modems. Le programme mis en oeuvre consiste alors, pour l'utilisateur à distance, à simuler la connexion d'un terminal local, tout en transmettant les données à travers le réseau. Par les mêmes techniques, il est possible d'accéder à des informations distantes, telles que des bases de données par exemple. La transmission de fichiers ou l'envoi de messages se font par des techniques voisines. Elles évitent les disquettes, les bandes magnétiques ou les lettres par la poste. Dans leurs versions plus élaborées, les réseaux permettent de partager des ressources telles que des imprimantes ou des disques. Ce partage suit le modèle du client et du serveur. Le serveur étant le logiciel ou le matériel qui fournit le service. Il est alors possible de construire ainsi des serveurs d'impression, de disques, de fichiers... Celui qui utilise le service est appelé le client.

Un réseau est constitué de "noeuds", tels que des ordinateurs, des imprimantes, ... Chaque noeud possède une adresse à laquelle nous pouvons associer un nom. Ces noeuds sont reliés entre eux par un support de transmission. Ce support peut être formé de paires de métal torsadées, de câbles coaxiaux, de fibres optiques, ou bien être l'atmosphère, ... Des signaux émis selon un certain codage, et regroupés en trames d'un certain nombre d'octets, transitent sur le support. Les signaux peuvent être diffusés en bande de base ou bien modulés. Les noeuds sont agencés suivant une architecture, une topologie. Cette topologie peut être un bus, une étoile ou bien un anneau. Nous devons différencier la topologie physique, qui correspond à l'agencement des noeuds, de la topologie logique, qui est le mode de circulation des données. Pour TokenRing, l'anneau à jeton d'IBM, la topologie physique est une étoile, la topologie logique est un anneau. Les noeuds accèdent au support :

- de manière aléatoire, en détectant une porteuse pour la plupart des réseaux Ethernet,
- de manière semi-répartie, en se saisissant d'un jeton, pour l'anneau d'IBM,
- ou bien par l'intermédiaire d'une commande centrale et d'une commutation de données dans le réseau téléphonique et dans les réseaux locaux modernes.

Ces noeuds sont rattachés au support par un dispositif matériel de connexion, une carte électronique, qui se charge, en général, de gérer l'accès.

2.1.1 Types de réseaux

Les réseaux inter-sites sont encore appelés WAN et nécessitent de mettre en oeuvre des moyens particuliers (modem, routeurs, commutateurs, passerelles...) pour s'affranchir des problèmes de distance et de disponibilité de liaisons dont les débits inter-site sont toujours, hélas, plus lents qu'en local. Ils offrent un service point à point entre deux interlocuteurs. Exemples : Transpac (filiale France Telecom), RATP, SNCF (fibres optiques sous les voies), Air France (via satellites), ...

Les réseaux locaux, encore appelés LAN sont constitués de moyens de communication internes à un établissement, donc entièrement maîtrisés par l'entreprise. C'est un système de communication de données limité à une zone géographique restreinte (jusqu'à 10 Km environ) et utilisant des débits moyens élevés (de 100 kbps à 50 Mbps). La zone servie peut être un simple bâtiment, un complexe de bâtiments ou un « campus ». Le réseau utilise une forme de commutation et n'emploie par les circuits des opérateurs publics, mais peut contenir des passerelles ou des ponts vers d'autres réseaux publics ou privés.

2.1.2 Les normes

Une norme est l'ensemble des règles qui doivent être respectées pour réaliser un échange entre deux ordinateurs. Il existe différents types de protocoles comme TCP/IP, les séries V et X (CCITT ou Comité Consultatif International des Téléphones et Télécommunications) et le modèle OSI.

Pour garder une certaine cohérence, une structure et une organisation dans le domaine des réseaux, il faut des normes. Ces normes sont crées et approuvées par un certain nombre d'organismes de normalisation. Pour n'en citer que quelques uns :

- ISO (International Organisation for Standardisation)
- CEI (Commission Electrotechnique Internationale)
- CCITT

A ces organismes internationaux s'ajoutent encore des organismes de différents continents comme l'Europe ou les Etats-Unis :

- AFNOR (Association Française de NORmalisation)

- ECMA (European Computer Manufacturer Association)
- ANSI (American National Standart Institute)
- IEEE (Institute of Electrical and Electronics Engineers)
- SEV (Schweizerischer Electrotechnischer Verein)
- DIN (Deutsches Institut für Normung)

Nous allons nous concentrer sur les normes de l'IEEE qui est un des organismes les plus actifs dans le domaine des réseaux locaux. Voici en résumé ces différentes normes :

- 802.1 GESTION DE RESEAU décrit les relations entre les normes ci-dessous.
- 802.2 LIEN LOGIQUE (LLC) définit la partie LLC (Logical Link Control) de la couche 2.
- 802.3 RESEAU CSMA/CD pour les réseaux à topologie bus et méthode d'accès CSMA/CD.
- 802.4 RESEAU TOKEN BUS pour les réseaux à topologie bus avec méthode d'accès à jeton.
- 802.5 RESEAU TOKEN RING réseaux en anneau avec méthode d'accès à jeton.
- 802.6 METROPOLITAN AREA NETWORK réseaux à l'échelle d'une ville. Il ne semble pas à l'heure actuelle qu'il y ait besoin de réseaux purement métropolitains.
- 802.7 TRANSMISSION LARGE BANDE c'est une norme qui se base sur les réseaux 802.3 et 802.4.
- 802.8 RESEAUX FIBRE OPTIQUE.
- 802.9 VOIX + DONNEES concerne l'utilisation d'un seul support physique pour transporter la voix et les données.
- 802.10 SECURITE DES RESEAUX LOCAUX étudie les problèmes de sécurité dans les réseaux.
- 802.11 RESEAUX LOCAUX SANS FIL transmission infrarouges, micro-ondes, ondes hertziennes, etc.

2.1.3 Le modèle ISO

Le modèle ISO-OSI, de loin le plus répandu, décrit des niveaux de transmission mais pas des normes. Il divise l'ensemble des protocoles en sept couches indépendantes entre lesquelles sont définis deux types de relations ; les relations verticales entre les couches d'un même système (interfaces) et les relations horizontales relatives au dialogue entre deux couches de même niveau :

N°	Nom de la couche	Exemples de protocoles
7	Application	FTP, TELNET, SMTP
6	Présentation	XDR : eXternal Data Representation
5	Session Remote	Procedure Call
4	Transport	TCP (mode circuit virtuel) UDP (mode datagramme)
3	Réseau	IP (Internet Protocol)
2	Liaison	
1	Physique	ETHERNET

La couche physique (1) s'occupe de la connexion physique sur le réseau (modulation, câbles, synchrone ou pas...). Les plus connues sont le V24 (modem) et le X21 (ETTD et ETCCD : terminal...). La couche liaison (2), contrôle de flux, a pour but de transmettre les données sans

erreur. Elle utilise, par exemple, le protocole HDLC ou le LAP-B. La couche réseau (3) assure la commutation et le routage des paquets entre les noeuds du réseau, par exemple l'IP. Deux méthodes d'acheminement se distinguent :
- la commutation de circuits
- la commutation de paquets.

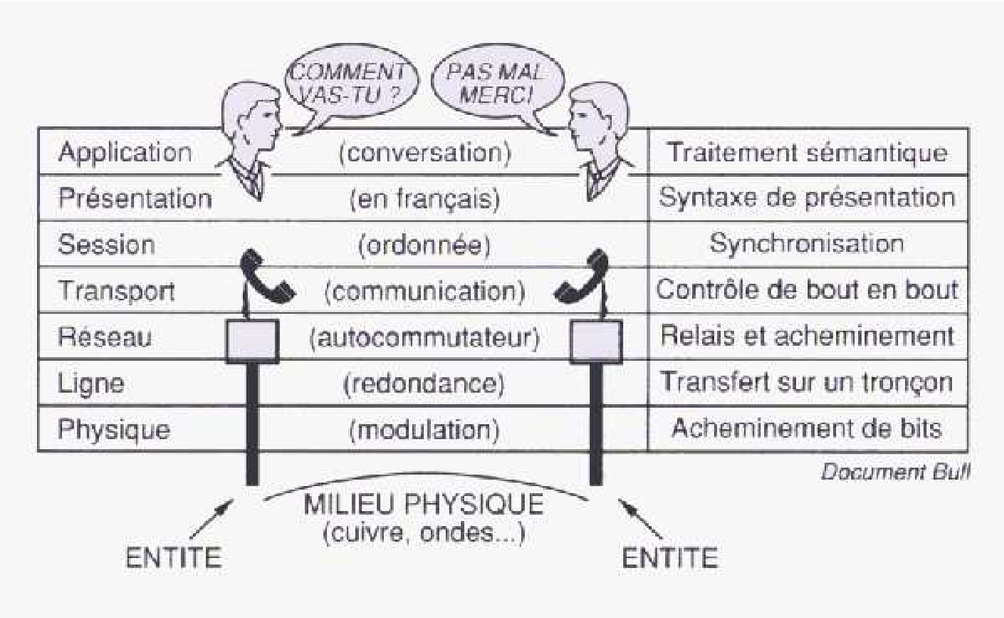

FIG. 2.1 – Modèle ISO

2.1.4 Topologie des réseaux

Depuis sa création il y a presque quinze ans, la micro-informatique n'a cessé de s'améliorer, de se démocratiser. Elle n'est plus réservée à une élite de savants. Mais pour atteindre cette maturité, elle a dû s'installer dans les milieux professionnels. Dans l'industrie, la micro-informatique a et aura de plus en plus d'importance. Dans une entreprise, l'informatique permet d'archiver, de créer des produits, ainsi que de tester, mais elle permet surtout de communiquer. Pour cela, il faut posséder un réseau informatique. Ce réseau doit être fiable, rapide et surtout facile à gérer. Le type de réseau peut-être déterminant pour la bonne marche de l'entreprise. Il devra être adapté aux besoins de l'entreprise. Dans un sondage réalisé il y a quelques années déjà aux Etats-Unis, il était dit que si une entreprise était brutalement privée de son informatique, suite à un incendie ou un tremblement de terre par exemple, il y avait 95 % de risque que cette entreprise fasse faillite. C'est pourquoi aujourd'hui, le bon fonctionnement d'une entreprise dépend avant tout de son informatique, et son implémentation est une étape relativement difficile et minutieuse.

Nous allons nous concentrer sur les réseaux avec un, ou plusieurs, serveur(s) dédié(s). Par topologie, nous entendons la façon dont les machines sont connectées au serveur. Il y en a trois principales :

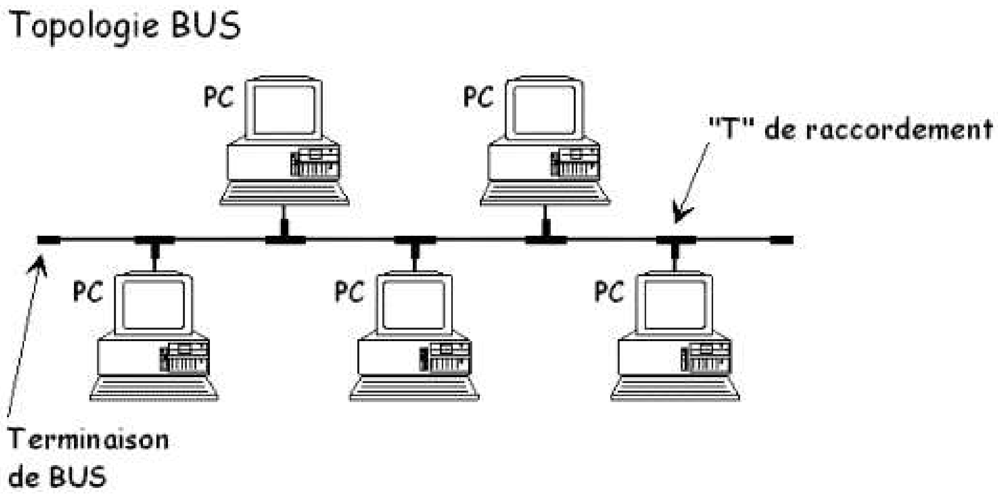

FIG. 2.2 – Topologie en Bus

Bus

Tous les équipements sont branchés en série sur le serveur. Chaque poste reçoit l'information, mais seul le poste pour lequel le message est adressé traite l'information. Un câble coaxial est utilisé pour ce type de topologie. L'avantage du bus est sa simplicité de mise en oeuvre et sa bonne immunité aux perturbations électromagnétiques. Par contre, si le câble est interrompu, aucune communication n'est possible sur le réseau.

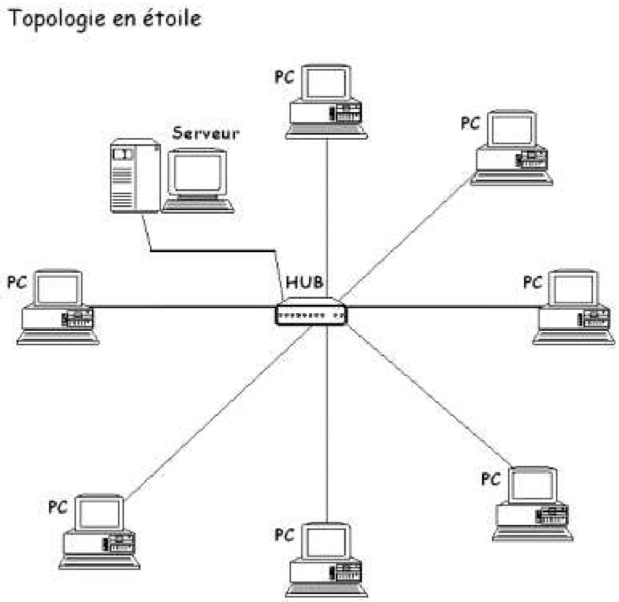

FIG. 2.3 – Topologie en Etoile

Etoile

Dans cette topologie, toutes les liaisons sont issues d'un point central. C'est une liaison dite « point à point », c'est à dire que les équipements sont reliés individuellement au noeud central et ne peuvent communiquer qu'à travers lui. Des câbles en paires torsadées ou en fibre optique sont utilisés pour ce type de topologie. L'avantage est que les connexions sont centralisées et facilement modifiables en cas de défectuosité. Si un câble est interrompu, le reste du réseau n'est pas perturbé. L'inconvénient de taille de cette topologie est l'importante quantité de câbles nécessaire.

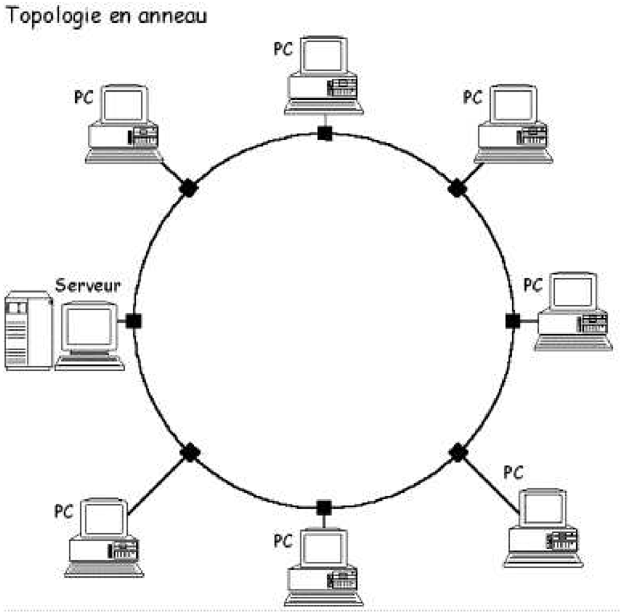

FIG. 2.4 – Topologie en Anneau

Anneau

Les équipements sont reliés entre eux en formant une boucle. La liaison entre chaque équipement est point à point. L'information est gérée comme dans la topologie bus. Chaque station reçoit le message, mais seule la station à qui le message est adressé le traite. Pour le câblage, un câble en paires torsadées ou une fibre optique est nécessaire. L'avantage est que l'anneau offre deux chemins pour aller d'un point à l'autre. Ceci permet à l'information de passer malgré une coupure sur le câble. Cette topologie est associée aux réseaux de type Token Ring. L'utilisation d'un double anneau (si le premier anneau est interrompu, les données passent sur l'anneau secondaire le temps de réparer le premier anneau) permet d'augmenter la sécurité.

Hybride

C'est un mélange des différentes structures de réseau.

2.1.5 Les méthodes d'accès

Ethernet

Le principe de base du réseau Ethernet CSMA/CD est la diffusion à toutes les stations de toutes les trames. La propagation de ces trames est bidirectionnelle. Les débits prévus par la norme sont de 1Mb/s, 10Mb/s et maintenant 100Mb/s.

Pour qu'une station puisse émettre, elle doit d'abord « écouter », vérifier que le média est libre ; c'est-à-dire qu'aucune autre station n'émette au même moment (Carrier Sense). Grâce au droit d'accès multiple (Multiple Access) la station émettrice émet sa trame. Si une trame est en circulation, alors l'émetteur continue la phase de détection jusqu'à ce que le média soit libre. Mais si deux ou plusieurs stations tentent de communiquer au même moment, il se crée une collision. Pour gérer ce problème, la norme prévoit une technique de détection de collision (Collision Detect). Cette technique commence par détecter la collision, puis les machines à l'origine de la collision réémettent des trames de façon à indiquer à toutes les stations qu'une collision a eu lieu. Les stations arrêtent d'émettre et attendent pendant un certain laps de temps aléatoire, puis toutes les stations se remettent en mode d'émission.

Token Ring (anneau à jeton)

Un jeton (token) circule sur l'anneau de station en station. Lorsqu'une station désire émettre, elle prend le jeton, y place sa trame, son adresse et l'adresse du destinataire, puis remet le jeton en circulation. Le destinataire détecte que le jeton lui est adressé et le prend. S'il veut émettre aussi, il suit la même procédure, sinon il remet le jeton en circulation sans rien lui adjoindre.

Etoile

Chaque poste est relié au serveur ou un concentrateur (HUB) si le réseau est grand. Le serveur est le point central du réseau. Toutes les communications passent par le serveur ou le HUB. Si une station désire émettre, elle envoie sa trame au concentrateur qui va se charger d'envoyer le message au bon destinataire.

2.2 L'interconnexion

2.2.1 Les moyens mis à disposition

Des réseaux de différents types, possédant des topologies différentes, peuvent, par l'intermédiaire d'un ou plusieurs autres noeuds, être reliés entre eux ; ils sont alors dits interconnectés. La nécessité d'interconnecter plusieurs réseaux locaux apparaît dans les entreprises qui disposent de plusieurs établissements ou dans celles qui désirent communiquer entre elles et s'échanger des informations. En effet, la plupart des réseaux ont des environnements de communication indépendants, destinés à satisfaire les besoins spécifiques (transmission de données, téléphonie...). Malheureusement, il est impensable de trouver sur le marché un réseau unique et universel satisfaisant tous les besoins. Assurer l'interconnexion consiste donc à établir un dialogue entre des machines d'origines diverses ; celles-ci peuvent être, en outre, reliées par des modèles différents de réseaux. De plus, l'interconnexion de plusieurs sous-réseaux est souvent préférée à l'utilisation d'un réseau unique pour délimiter différents domaines d'activité, accroître la sécurité et optimiser les taux de transfert à l'intérieur de chaque sous-réseau. Ainsi, certains utilisateurs vont disposer de réseaux à haut débit pour interconnecter leurs ordinateurs, tout en sachant que ces réseaux ne pourront couvrir que des espaces géographiques limités. D'autres, pour relier des ordinateurs éloignés de plusieurs milliers de kilomètres, n'auront à disposition que des réseaux étendus, offrant de faibles débits. L'objectif est alors d'assurer aux utilisateurs une parfaite transparence de l'interconnexion, quels que soient les divers protocoles mis en oeuvre par les équipements de communication, cf OSI 2.1.3.

Divers équipements permettent d'interconnecter les réseaux locaux, en intervenant au niveau des différentes couches du modèle OSI. Chaque dispositif, relié à chacun des réseaux, doit pouvoir accueillir les messages qui ne lui sont pas destinés, puis les acheminer correctement. Si un grand nombre de réseaux sont interconnectés, cette gestion peut rapidement devenir délicate : il est alors souvent nécessaire de faire transiter les messages par plusieurs réseaux intermédiaires. Cette opération de routage sous-entend que les équipements connaissent l'emplacement de chaque machine dans l'ensemble des réseaux.

Nous pouvons distinguer plusieurs types de dispositifs d'interconnexion. Les répéteurs, qui amplifient les signaux transmis afin de prolonger le support de communication, se limitent à la première couche OSI. Les ponts permettent de relier des réseaux de même technologie afin de constituer un seul réseau local logique. Opérant par conséquent dans la couche 2, ces dispositifs peuvent ainsi traiter certaines erreurs, notamment les collisions entre messages, en veillant à ne pas les propager.

Les routeurs sont utilisés pour relier des réseaux locaux de technologies différentes (par exemple Ethernet et Token Ring). Intervenant sur la couche réseau, ils sont à même d'acheminer des paquets au travers d'un vaste ensemble de réseaux interconnectés. Enfin, les passerelles, qui opèrent sur toutes les couches, sont mises en oeuvre dans les cas complexes où les systèmes interconnectés sont radicalement différents. Elles permettent notamment de relier des systèmes informatiques centralisés à des systèmes ouverts, cf équipements d'interconnexion 2.2.2.

Une palette d'équipements permet d'interconnecter toutes les machines utilisées dans les entreprises, offrant ainsi aux usagers un réseau unique dont la disparité n'apparaît pas à l'emploi.

Pour interconnecter leurs réseaux locaux sur de grandes distances, les entreprises font appel aux services des opérateurs de télécommunications. Ces opérateurs peuvent leur proposer des liaisons spécialisées, fort chères, mais surtout différents réseaux publics : le réseau téléphonique commuté classique, dont la liaison s'établit à l'aide de modems, le réseau numérique à intégration de services ou encore des réseaux de transmission de données spécialement voués à l'interconnexion, comme le réseau Transpac.

Les opérateurs de télécommunications offrent plusieurs services réservés à l'interconnexion de réseaux à haut débit : relais de trames, SMDS, ATM ou encore commutation de paquets (norme X25). Ainsi, le réseau Transpac, ouvert en 1978, constitue le premier réseau mondial de commutation par paquets. Au début des années 1990, plus de 100 000 abonnés y étaient reliés afin d'assurer leurs échanges de données informatisées.

De tels services d'interconnexion ont mis longtemps à se mettre en place, du fait des difficultés techniques, mais surtout à cause des problèmes de normalisation. C'est notamment pour cette raison que s'est développée, à la fin des années 1970, la famille de protocoles appelée TCP/IP, du nom des deux principaux protocoles. Ils constituent un modèle d'interconnexion quelque peu différent mais surtout plus simple que le modèle OSI à sept couches, cf TCP/IP 2.5

Comme ces protocoles sont faciles à mettre en place et permettent d'interconnecter tous les types de réseaux, ils se sont rapidement développés, grâce à la gratuité de leur usage, puisque n'appartenant pas à une compagnie en particulier. C'est ainsi qu'est apparu le réseau Internet qui désigne un ensemble, devenu très important et couvrant toute la planète, de machines reliées entre elles selon ces protocoles ; chacune pouvant communiquer avec toutes les autres.

2.2.2 Les équipements d'interconnexion

L'interconnexion de réseaux peut être locale : les réseaux sont sur le même site géographique. Dans ce cas, un équipement standard (répéteur, routeur, ...) suffit à réaliser physiquement la liaison. L'interconnexion peut aussi concerner des réseaux distants. Il est alors nécessaire de relier ces réseaux par une liaison téléphonique (modems, ...).

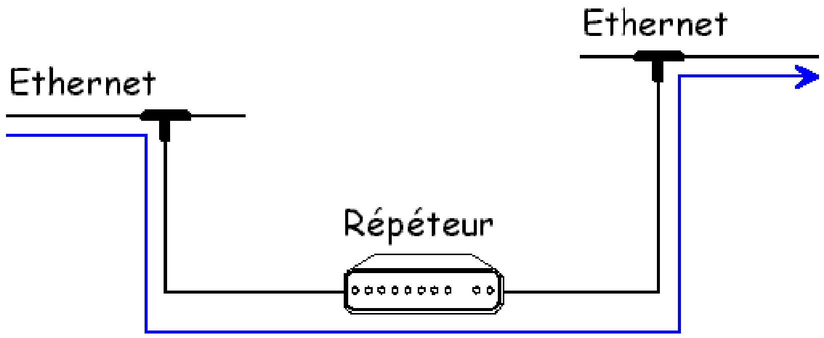

Fig. 2.5 – Le répéteur

Le répéteur

Il permet d'interconnecter deux segments d'un même réseau. Il travaille au niveau de la couche OSI 1. Ces fonctions sont :

– la répétition des bits d'un segment à l'autre,
– la régénération du signal pour compenser l'affaiblissement,
– changer de média (passer d'un câble coaxial à une paire torsadée).

Le pont (Bridge)

Ce sont des équipements qui décodent les adresses machines et qui peuvent donc décider de faire traverser ou non les paquets. Le principe général du pont est de ne pas faire traverser les trames, dont l'émetteur et le destinataire sont du même coté, afin d'éviter du trafic inutile sur le réseau. Il permet d'interconnecter deux réseaux de même type et travaille au niveau de la couche OSI 2. Il permet aussi de filtrer les trames. Si les stations émettrices et réceptrices se trouvent du même côté du pont, la trame ne le traversera pas pour aller polluer le deuxième segment.

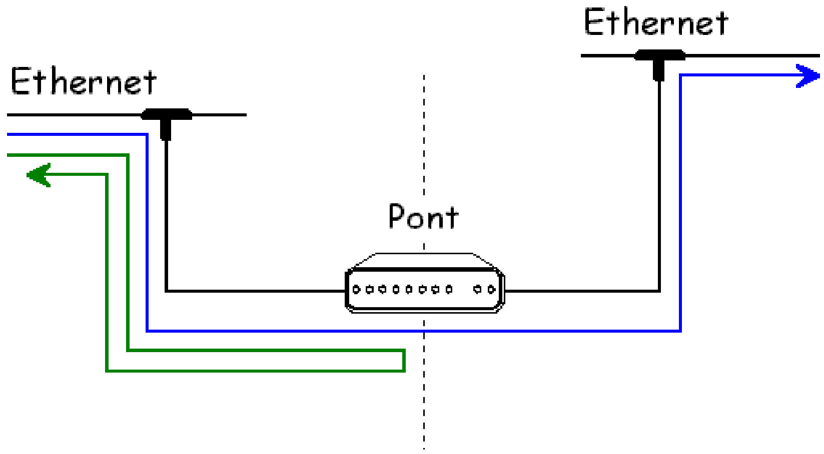

FIG. 2.6 – Le pont

Le routeur (Router)

Les routeurs manipulent des adresses logiques (ex : IP) et non physiques (ex : MAC). Ils ne laissent pas passer les broadcasts[1] et permettent un filtrage très fin des échanges entre les machines, grâce à la mise en oeuvre de listes de contrôle d'accès. Les droits de chaque machine vont être décrits. C'est un équipement qui couvre les couches 1 à 3 du modèle OSI. Il est généralement utilisé pour l'interconnexion à distance. Il est surtout employé pour l'interconnection de plusieurs réseaux de types différents (Ethernet, Token ring). Un routeur est multi-protocoles : IP, IPX, DECnet, OSI, Appletalk, ...Le routeur est capable d'analyser et de choisir le meilleur chemin pour véhiculer la trame à travers le réseau. Il optimise ainsi la transmission des paquets.

[1]trame envoyée à toutes les machines du réseau

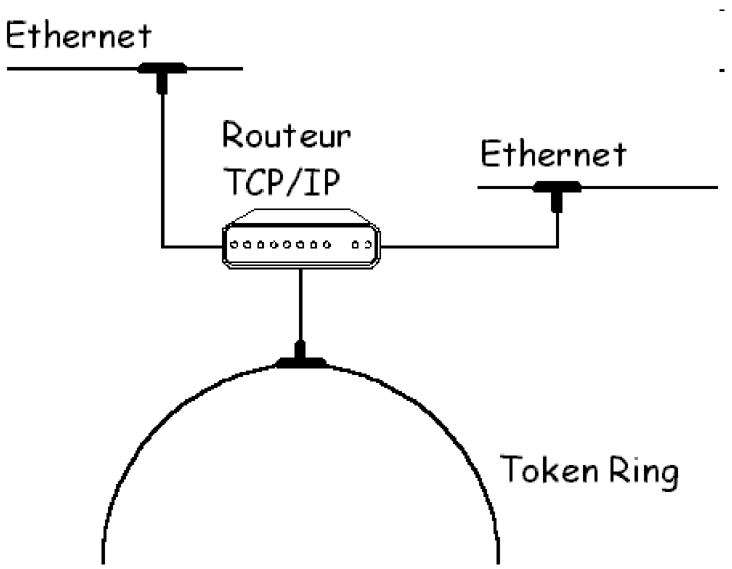

FIG. 2.7 – Le routeur

La passerelle

C'est un système complet du point de vue de la connexion. C'est généralement un ordinateur. C'est le seul qui travaille jusqu'à la couche OSI 7 des différents protocoles qu'il utilise. Une passerelle est utilisée pour la connexion entre un réseau local et un système informatique qui ignore totalement le réseau local, par exemple pour relier un réseau PC sous NT avec un AS/400 ou un VAX.

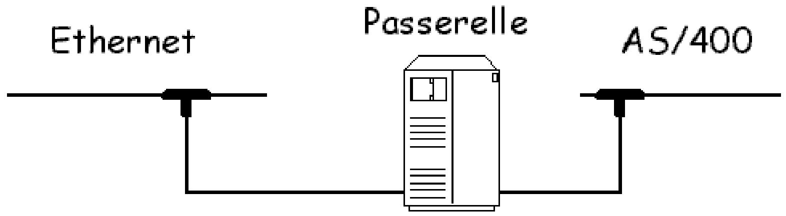

FIG. 2.8 – La passerelle

Le concentrateur (HUB)

C'est un boîtier qui a la fonction de répéteur mais, sa fonction principale est de pouvoir concentrer plusieurs lignes en une seule. Plusieurs stations peuvent y être connectées, dont le nombre dépend du type de HUB. Un HUB sera connecté sur un autre HUB ou sur un serveur avec une seule et unique ligne.

Le commutateur(ou switch)

Le commutateur est un système assurant l'interconnexion de stations ou de segments d'un LAN en leur attribuant l'intégralité de la bande passante, à l'inverse du concentrateur qui la partage. Les commutateurs ont donc été introduits pour augmenter la bande passante globale d'un réseau d'entreprise, et sont une évolution des concentrateurs Ethernet (ou hubs). Ils ne mettent en oeuvre aucune fonctionnalité de sécurité (certains commutateurs savent toutefois gérer l'adresse ethernet (@ MAC)), hormis l'amélioration de la disponibilité. Plusieurs communications simultanées peuvent avoir lieu à condition qu'elles concernent des ports différents du commutateur. La bande passante disponible n'est plus de 10 Mbit/s partagés entre tous les utilisateurs, mais n x 10 Mbit/s.

2.3 Les Clients/Serveurs

Pour expliquer le modèle client/serveur, il faut remonter quelques années en arrière pour analyser le seul modèle de répartition des données et des traitements qui existait alors, celui des systèmes centralisés. Les terminaux passifs, qui sont inutiles si le système sur lequel ils sont connectés est éteint, ne fait qu'afficher les données à l'écran et transmettre les ordres saisis au clavier sans les interpréter. Mais le fait que les données et les programmes soient sur le même ordinateur permet des performances importantes et une meilleure gestion de la sécurité. Dans le domaine de la micro-informatique, ce modèle est encore présent sur les serveurs qui utilisent le système d'exploitation Unix.

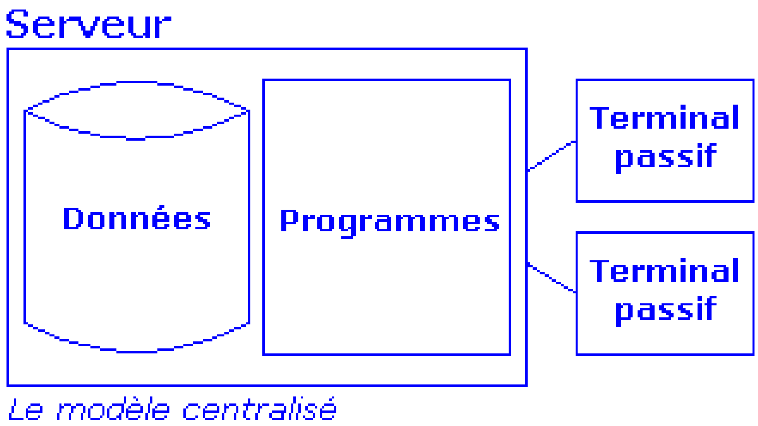

FIG. 2.9 – Le modèle centralisé

Dans le modèle serveur de fichiers, le serveur contient toutes les données (fichier client, écritures comptables...) et ne fait que servir les informations vers les micro-ordinateurs, un peu comme un disque dur. Dans ce modèle, c'est l'application sur le micro-ordinateur qui va parcourir toutes les fiches clients à la recherche des clients situés ailleurs, faisant transiter inutilement sur le réseau les 90% de fiches qui ne sont pas concernées par la recherche. Ce modèle est encore prédominant dans les petites et moyennes organisations. Le serveur de fichiers fait également office de serveur d'impression et de sauvegarde.

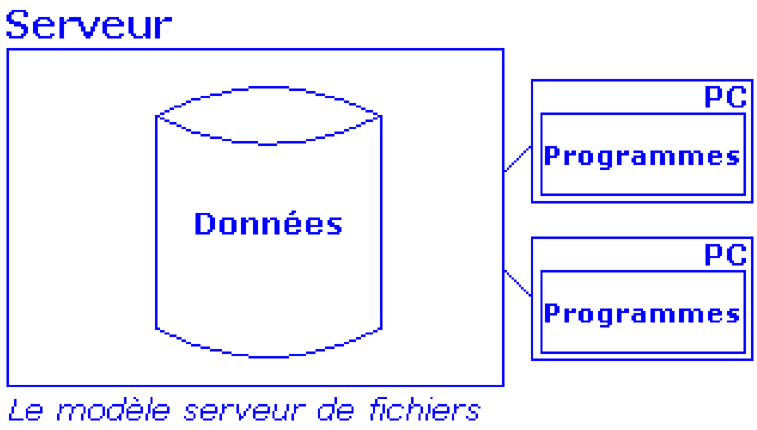

Fig. 2.10 – Le modèle serveur de fichiers

Le modèle client/serveur, qui reprend le meilleur des deux modèles précédents, permet une répartition optimisée des programmes et des données. Dans le modèle client/serveur, l'application installée sur le serveur va directement effectuer les recherches sur les données, et ne va retourner au micro-ordinateur que les fiches sélectionnées, pour un traitement final par ce dernier : affichage, impression, importation dans un tableur...

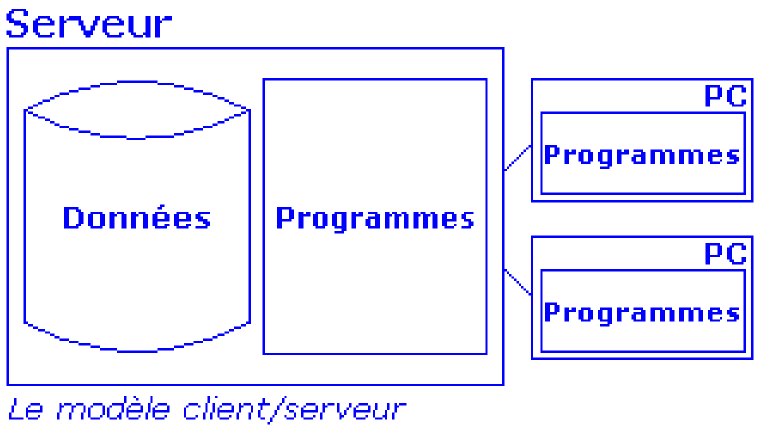

Fig. 2.11 – Le modèle Client/Serveur

L'application client/serveur la plus courante est le système de gestion de Bases de Données (SGBD), mais même dans des petites structures, d'autres applications peuvent bénéficier de ce modèle (serveur de messagerie).

2.4 Les sockets

Les sockets permettent des échanges de données entre plusieurs processus. Les programmes utilisant les sockets pour communiquer entre eux, peuvent être situés sur des machines distinctes ou sur la même machine.

La communication entre des programmes demande au moins un processus serveur, et des processus clients. Un processus serveur est un processus qui attend les demandes d'ouverture de connexions. À l'inverse, le processus client effectue des demandes d'ouverture.

Sur le serveur, il faut indiquer le protocole utilisé et le numéro du port local sur lequel les connexions devront être attendues. Le client doit lui aussi indiquer le protocole utilisé, l'adresse et le port de la machine ou fonctionne le processus serveur. Le numéro du port local du client est alloué dynamiquement. Une fois la connexion établie, un serveur fonctionne de la même façon qu'un client. Les deux peuvent envoyer ou recevoir des informations. Il existe 2 modes de connexion pour les sockets :

- Le mode connecté :
  - Socket : permet d'indiquer au système que l'application désire communiquer avec un autre programme.
  - Bind : permet à l'application de fournir au noyau les paramètres nécessaires au fonctionnement de la socket. Pour le serveur, il s'agit du port sur lequel il attendra les demandes de connexion.
  - Listen : permet de placer la socket et le protocole associé dans un état d'attente de requête de connexion.
  - Accept : place le programme en attente effective de requête de connexion.
  - Read, write : permettent l'échange des données .
  - Close : permet de fermer localement la socket.
  - Connect : permet d'ouvrir une connexion avec une application serveur

  A partir de l'ouverture de la connexion, le client et le serveur ont un comportement symétrique. Les instructions sont donc les mêmes.
- Le mode datagrame :

  Nous retrouvons le même mode de fonctionnement que le mode connectée, sauf que le client a un comportement symétrique par rapport au serveur et qu'il n'y a pas de procédure de connexion. Le client et le serveur envoient et recoivent des datagrames.

2.5 Les protocoles

La couche protocole, située au dessus de la couche Internet, est la couche de transport machine-à-machine, appellée aussi couche transport. Nous y rencontrons deux protocoles des plus important TCP et UDP. TCP fournit un bon rapport de fiabilité en utilisant un système de détection et correction d'erreur tandis que UDP est moins coûteux en ressource. Il est ainsi utile, selon le besoin de l'application, d'utiliser l'un ou l'autre de ces protocoles.

2.5.1 IP

Il s'agit du protocole le plus important de la couche Internet. IP fournit les services de base afin de transmettre des datagrammes (format des trames IP) sur le réseau. Tous les protocoles situés au dessus et en dessous de IP utilisent ce protocole pour transmettre des données. Ses fonctions incluent :

- définition d'un datagramme de transmission
- définitions du système d'adressage Internet

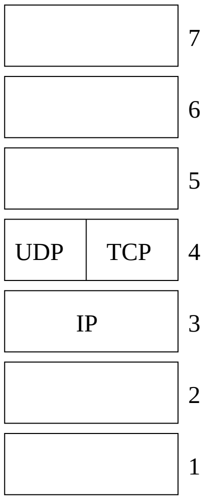

FIG. 2.12 – Couches des protocoles, Voir [Rose 95]

- assembler et fragmenter les datagrammes
- router les datagrammes vers les machines distantes
- transporter les données de la couche d'accès réseau à la couche de transport

IP est un protocole de type non connecté c'est à dire qu'il n'établit pas de connexion entre les deux machines pour réaliser la transmission des données. Il n'y a donc aucun système de détection des erreurs. Cela permet aux datagrammes d'être commuter vers la destination finale sans avoir forcement pris le même chemin. IP définit une adresse IP sur 32 bits qui permet de router au mieux les paquets distants, elle doit être unique afin d'identifier un réseau ou une machine.

2.5.2 TCP

C'est un protocole fiable de type connecté c'est à dire qu'il établit une connexion virtuelle entre les deux machines souhaitant dialoguer. La fiabilité vient du fait que chaque paquet émis reçoit un accusé de réception (les données sont arrivées et ne sont pas endommagées), si ce n'est pas le cas, le paquet est réémis jusqu'à ce qu'il y ait une bonne transmission. La connexion se fait en plusieurs étapes.

Prenons par exemple deux machines A et B : A veut savoir si la machine B distante est prête à communiquer, A envoie un segment à B, qui lui renvoie un accusé de réception disant qu'elle (B) est prête, puis A envoie un accusé de réception pour dire à B, qu'elle aussi est prête à communiquer. La connexion est ainsi établie entre les deux machines. La connexion est maintenant représentée par un flot continu d'octets. Le même mécanisme est utilisé pour fermer une connexion.

2.5.3 UDP

Ce protocole permet de transférer directement des datagrammes sur le réseau. Il est à l'opposé de TCP, de type non connecté et peu fiable, ce qui permet d'éviter une surcharge de messages. La notion de fiabilité signifie pour UDP, qu'il n'a aucun moyen de vérifier les erreurs, c'est à dire qu'il n'y a pas d'accusé de réception pour les messages émis. L'utilisation de UDP permet d'éviter une surcharge du réseau afin d'établir des connexions. UDP est efficace dans le cas où l'application est du type Questions/Réponses car l'application traite elle-même l'acquittement ou les erreurs de transmissions.

Chapitre 3

Administration de réseaux

3.1 Le protocole SNMP (Simple Network Management Protocol)

Il a été conçu pour être simple, d'où son nom, et prévu pour de petits sites où la puissance de traitement dans les systèmes à administrer n'est pas trop importante. Ce protocole a été enrichi dans sa version 2 afin de pouvoir administrer des sites plus importants et répartis géographiquement. Dans cette version, il fait concurrence au protocole CMIP (Common Management Information Protocol) de l'ISO.

Le rapport de 1998, concernant ce même projet, effectuait une comparaison qualitative de ces différents protocoles. Celle-ci avait conduit à choisir le protocole SNMPv1 pour mettre en œuvre l'outil. Lors de la reprise de ce projet, en 1999, le choix s'est porté sur le protocole SNMPv2 qui avait l'avantage de fournir des fonctionnalités supplémentaires. Nous avons donc poursuivit dans ce sens.

SNMP est le premier protocole d'administration réseau standardisé, il s'agit d'un protocole manager/agent. L'agent surveille le périphérique sur lequel il est installé et rend compte de son état au manager. Le manager interroge ses agents afin qu'ils répondent à ses requêtes et réunis ses informations afin de les montrer à l'administrateur. Le travail se situe au niveau du manager ce qui permet au programme agent d'être simple et facilement implantable. SNMP est donc un protocole requêtes/réponses. Il utilise UDP comme protocole de transport, ce qui permet d'éviter le surcoût qu'engendre TCP. Les requêtes du manager ne sont acquittées que lorsque l'agent a renvoyé la réponse, sinon le manager recommence son interrogation. Ce qui permet de réduire la charge du réseau au moment critique, et ainsi le temps d'intervention.

SNMP transmet ses requêtes et ses réponses dans des paquets appellés PDU, 5 sortes de PDU sont utilisées par SNMP afin de définir 3 types d'interactions entre le manager et les clients :

- Les interrogations du manager :
  Celui-ci envoie ses requêtes à l'agent qui interroge sa base de données afin de récupérer la ou les valeurs désirée(s).
- Les modifications du manager :
  Le manager décide de modifier une ou plusieurs valeurs dans la base de données de l'agent en lui envoyant le nom de la variable et sa valeur.
- Les alarmes :

Cette fois-ci, c'est l'agent qui envoie un message au manager indiquant qu'il se passe un événement extraordinaire. L'agent doit connaître l'adresse du manager.

Voici les 5 PDU disponibles avec le protocole SNMP :

PDU	Utilisation
GetRequest	permet au manager de consulter une série d'objets
GetNextRequest	permet de parcourir un tableau de la base de données
GetResponse	réponse de l'agent à une requête du manager
SetRequest	le manager modifie une série d'objets
Trap	l'agent alerte le manager

Le manager envoie de façon périodique des demandes de variables grâce à un GetRequest, l'agent lui renvoie la valeur dans un GetResponse. Pour cela l'agent et le manager utilisent le port UDP 161. Il suffit au manager d'interroger tous ses agents pour connaître l'état global du réseau à un moment donné. Cependant, cette méthode ne permet pas de faire face à un problème en temps réel, car il faut attendre que le manager interroge l'agent pour s'apercevoir qu'il y a un problème.

C'est à ce niveau qu'interviennent les Traps. L'agent informe le manager qu'un événement inhabituel est survenu. Pour ne pas rentrer en conflit avec les requêtes du manager, l'agent utilise le port UDP 162 pour envoyer ces Traps.
Les Traps sont générés par des événements prédéfinis :

Traps	Evénements
coldStart	l'équipement est mis sous tension, l'état de l'agent est modifié
warnStart	l'équipement vient de redémarer, l'état de l'agent n'est pas modifié
linkDown	une des liaisons de l'équipement a détecté une panne
linkUp	une des liaisons de l'équipement est devenue active
enterpriseSpecific	format d'événement spécifique à un constructeur
authenticationFailure	l'agent a reçu un message non identifié
egpNeighborLoss	l'agent est un routeur utilisant EGP, qui a perdu le contact avec un de ses voisins

L'administration réseau englobe de nombreux périphériques hétérogènes, des routeurs aux simples terminaux. Il nous est donc nécessaire d'avoir un langage indépendant du constructeur permettant de représenter les structures des objets.

3.2 Le protocole SNMPV2

SNMPv2 est la seconde version du protocole SNMP, il remédie aux limitations de la première version, au niveau de la sécurité, en assurant toujours des fonctionnalités et une implémentation simple. SNMPv2 définit 3 nouvelles opérations en plus de celles de SNMPv1 :

- Get_bulk_request : minimise les intéractions avec le réseau en permettant d'aller chercher plusieurs valeurs en une seule fois, et de renvoyer de grands paquets. Il faut ajouter à la structures les variables
  - non_repeaters : nombre de variables recherchées au plus une fois,
  - max_repetition : nombre maximun d'instances voulues des variables suivantes,
  - variable_bindings : les variables non répétées sont en tête et leurs valeurs ignorées.
- Inform_request : permet d'établir une communication entre deux entités gestionnaire et ainsi, de transmettre des informations permettant d'assurer une meilleure interaction avec le réseau.
- Snmpv2_trap : à la différence de SNMPv1, les notifications sont de structures identiques à celles des PDU utilisées dans les autres demandes.

3.3 Les RFCs (Request For Comments)

Les RFCs permettent de diffuser les informations sur la famille des protocoles Internet. Certains permettent de définir des standards, mais la majorité correspond à des notes de recherches. Les RFCs ne se limitent pas à la spécification formelle des protocoles TCP/IP mais contiennent bon nombre d'informations utiles et de mises à jour. Contrairement aux normes de l'ISO, ils ne sont pas payants. Les RFCs portent tous un numero d'identification correspondant à leur ordre de parution.

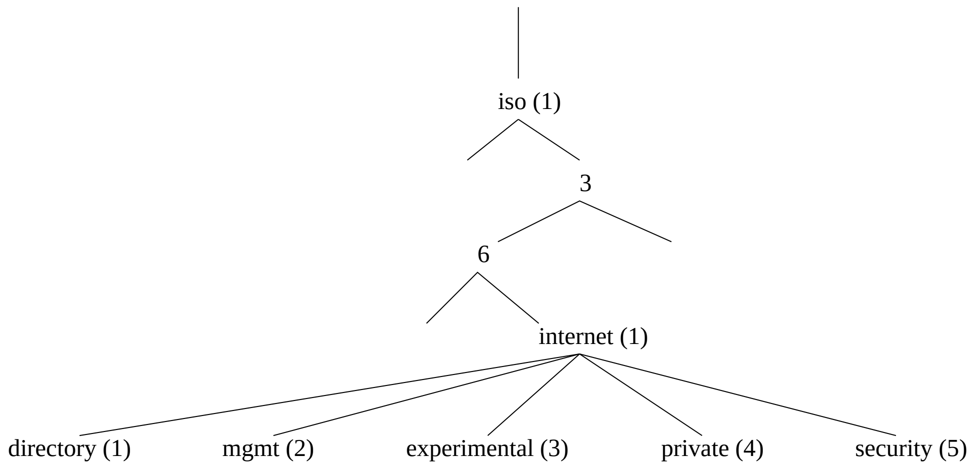

Fig. 3.1 – Arbre des OID

3.4 La syntaxe ASN.1

Chaque objet géré reçoit un nom unique appelé identificateur d'objet. Cet identificateur fait partie de l'espace de nommage géré par ISO. La figure 3.1 représente l'arbre des OID. L'OID permet d'attribuer de manière unique une référence à une structure de données, voir [Rose 95].

3.5 Le modèle ISO

Il faut ensuite donner une définition stricte de ces objets. Pour cela, une syntaxe spécifique est nécessaire : ASN.1. Il s'agit d'un ensemble formel de régles de langage pour définir des données de manière universelle. ASN1 est dérivé de BNF. Grâce aux macros définies dans le RFC1212, il est plus facile de définir un objet, celui-ci comprend les champs suivants :

- SYNTAX donne le type de l'objet par exemple universel ;
- ACCESS donne les droits d'accès du manager :
  - read-only : lecture seulement,
  - read-write : lecture écriture,
  - write-only : écriture seulement,
  - non-accessible : le manager n'a pas accès à l'objet (l'objet n'a qu'un intérêt syntaxique) ;
- STATUS donne la place de l'objet :
  - mandatory : obligatoire,
  - optional : optionnel,
  - deprecated : ne doit plus être utilisé,
  - obsolete : le système administré ne doit plus le mettre en œuvre ;
- INDEX facultatif, permet d'identifier de manière unique l'entrée d'un tableau,
- DESCRIPTION donne en anglais une description de l'objet.

Voici un exemple :

```
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 }
```

L'objet déclaré sysName est de type DisplayString et de longueur maximale de 255 caractères. Son accès est autorisé en lecture/écriture. Il doit être présent dans tous les systèmes administrables par SNMP. Son identificateur est 1.3.6.1.2.1.1.5.. Une fois la structure définie, il faut la coder pour son transfert sur le réseau. Pour cela nous utilisons le codage BER. Il permet, indépendamment de la machine utilisée, d'être décodé.

3.6 La MIB

Lorsque tous les objets ont été définis, il suffit de les réunir pour former la base de données : la MIB. Chaque agent a une MIB renfermant toutes les informations utilisables par SNMP. ASN.1 laisse une grande liberté afin de créer des MIB, c'est pour cela que le RFC 1155 a été défini. Il permet de spécifier la conception des MIB appellée SMI. Chaque MIB est définie de façon formelle par un RFC. Cela permet d'avoir une MIB standart : la MIB I, permettant de contrôler TCP/IP. La MIB II n'est qu'un sur-ensemble de MIB I définie dans le RFC 1213. Chaque agent gère MIB I ou MIB II. La MIB privée, définie par un constructeur pour un périphérique spécifique tels que les routeurs, les ponts, permet de rajouter des informations. Cependant, la MIB privée n'est utile que si le manager la gère.

3.7 La MIB RMON

La MIB RMON est un module de surveillance défini par le RFC 1757 et permet à l'utilisateur d'obtenir des statistiques sur le réseau local. Elle met en œuvre des sondes, directement branchées sur le réseau, qui traitent les informations jusqu'au niveau Transport. Elle definit neuf groupes :

- statistiques sur l'Ethernet : permet d'obtenir des informations sur les paquets d'une source quelconque,
- contrôle historique : contient les informations sur le contrôle,
- alarmes : objet de génération d'alarmes,
- évenements : génération et notification d'événements,
- filtres : permet de filtrer les paquets,
- captures de trames : capture de paquets definis par les filtres précédents,
- matrices : permet de trier les statistiques entre deux réseaux,
- host : contient les statistiques et informations relatives à chaque machine découverte,
- classement : classe les stations selon un critère voulu.

Nous disposons, maintenant, des connaissances suffisantes dans le domaine réseau pour débuter la réalisation de l'outil d'administration d'interconnexion.

Chapitre 4

L'outil d'administration

Le développement des interconnexions permet d'accroître la " puissance " des réseaux mais elle entraîne aussi une complexification de la maintenance de ces structures. L'administration de réseaux comprend donc la détection et la résolution des problèmes qui peuvent survenir sur un réseau.

4.1 L'administration d'interconnexion réseau

L'administration de réseaux recouvre l'ensemble des fonctions qui sont nécessaires pour l'exploitation, la sécurité, le suivi et l'entretien du réseau. L'ISO définit cinq domaines : fautes, performances, sécurité, comptabilité et configuration. L'administrateur réalise trois principaux types d'actions :

- des actions en temps réel pour connaître l'état de fonctionnement de son réseau (surveillance et diagnostic des incidents, mesure de la charge réelle, maintenance, contrôle, information aux utilisateurs, ...), agir sur celui-ci (réparation, ajout et retrait de nouveaux abonnés, ...) et assurer la sécurité (contrôler les accès, créer et retirer des droits d'accès, ...).
- des actions différées pour planifier, optimiser, quantifier et gérer les évolutions du réseau (statistiques, comptabilité, facturation, prévention, évaluation des charges, ...).
- des actions prévisionnelles qui lui permettent d'avoir une vision à moyen et à long terme, d'évaluer des solutions alternatives, d'envisager les configurations, de décider du plan d'extension, de vérifier la pertinence de la solution réseau pour un problème donné.

L'administration d'interconnexion nous amène à définir quelques notions supplémentaires :

- Un agent est un programme exécuté sur l'équipement à administrer, qui a directement accès aux parties matérielles et logicielles. Il est interrogé à distance et fournit les informations ou exécute les instructions demandées.
- A l'autre extrémité se trouve(nt) la ou les plate(s)-forme(s) d'administration ou manager. Ce logiciel, souvent graphique, interroge, via un protocole d'administration propriétaire ou normalisé, les agents et présente les résultats à l'administrateur après des traitements de complexités variables.

Afin d'améliorer notre perception des diffèrents problèmes (interconnexion de réseaux, topologies, ...) énoncés précédemment, nous avons mis en œuvre une application en vue de faciliter l'administration de réseaux et ainsi éviter le mode commande plus difficiles à déchiffrer. Pour permettre une vue d'ensemble de la topologie du réseau, nous avons travaillé sur une carte représentant les différents agents à administrer (machines, switch, serveur, imprimantes...). Lorsque la carte est construite, il nous est possible de passer en mode surveillance, et d'interroger, en lançant des requêtes, les agents susceptibles d'avoir une MIB.

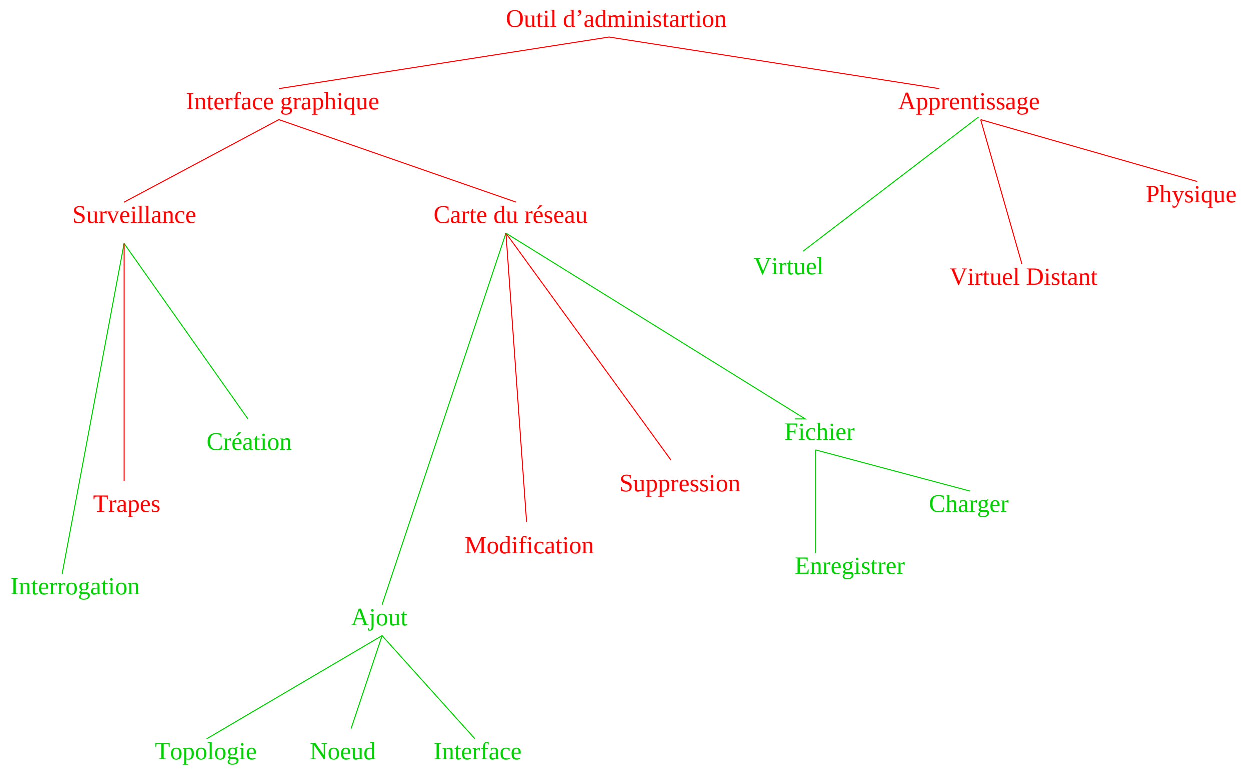

Fig. 4.1 – Concepts

4.2 Modes d'apprentissage

Afin d'assurer la notion d'apprentissage, nous définissons trois modes de travail accessibles. Chacun des modes est accessible selon le niveau d'administration désiré. Dans la suite, nous considérons des étudiants en T.P. comme utilisateurs. Ainsi, il sera possible de commencer par un travail simple et facile à comprendre, puis lorsque un bon niveau est atteint, de passer au suivant qui s'accentue en difficulté et qui demande plus de précisions. Cependant, ces trois fonctionnalités ne changent pas le concept de l'interface graphique, mais seulement quelques fonctionnalités dans chaque mode.

4.2.1 Le mode Virtuel local

Le mode LOGIQUE permet de construire et d'administrer un réseau virtuel. Toujours dans le concept Agent/Manager, mais les agents et le manager sont réunis sur la même machine. Cela permet de centraliser le travail sur un poste, ce qui permet à chaque groupe d'étudiants de travailler séparement. C'est dans cette optique que nous avons réalisé l'interface graphique. Les étudiants pourront tester diffèrentes configurations possibles, administer, et commettre des erreurs. C'est ce mode que nous avons développé dans notre application.

4.2.2 Le mode Virtuel distant

Les requêtes et les traps sont générés par l'administrateur lui-même sur le périphérique désiré, afin de simuler les conditions de déclenchement voulues (nous travaillons alors sur les vues/copie MIB des agents). Il y a un agent par machine distante. Cette situation permet d'analyser le concept des sockets et des clients serveurs. Il faut effectuer une copie des MIBs sur tous les agents. Les seules différences sont :

- lorsque nous installons un nouveau noeud dans le réseau, il faut tester si il peut y avoir une connexion possible entre les deux machines ; Par exemple, en effectuant un test avec la commande ping. Il est encore possible que les données des machines distantes soient différentes des données réelles. Cependant, il doit y avoir correspondance entre la connaissance du manager (nombre d'interfaces sur un switch) et l'agent (nécessité du même nombre d'interfaces).
- le fait que les vues MIBs et les traps soient enregistrés sur la machine distante. C'est à dire qu'il est possible de faire une application qui crée la vue MIB distante. Grâce à un telnet, nous lançons l'application et la paramètrons comme désirée, puis nous pouvons rapatrier le fichier par un ftp.

4.2.3 Le mode Physique

Nous travaillons directement sur les MIBs des périphériques et tout changement est effectué directement. Ce mode ne doit-être accessible qu'en mode super-administrateur, car il est alors possible de changer certaines variables des MIBs, pouvant causer des dégâts sur les périphériques. Par exemple, si nous effectuons un SetRequest sur une variable pouvant causer des dégâts, le réseau peut rentrer dans un état critique. Cette fois-ci, l'agent est le démon snmpd qui tourne sur la machine distante.

4.3 Interface graphique

Le but d'une interface graphique est de permettre une accessibilité plus facile afin de créer et d'interagir avec l'environement de travail. Nous avons donc utilisé les librairies de Tkinter et Pmw la version 8.3

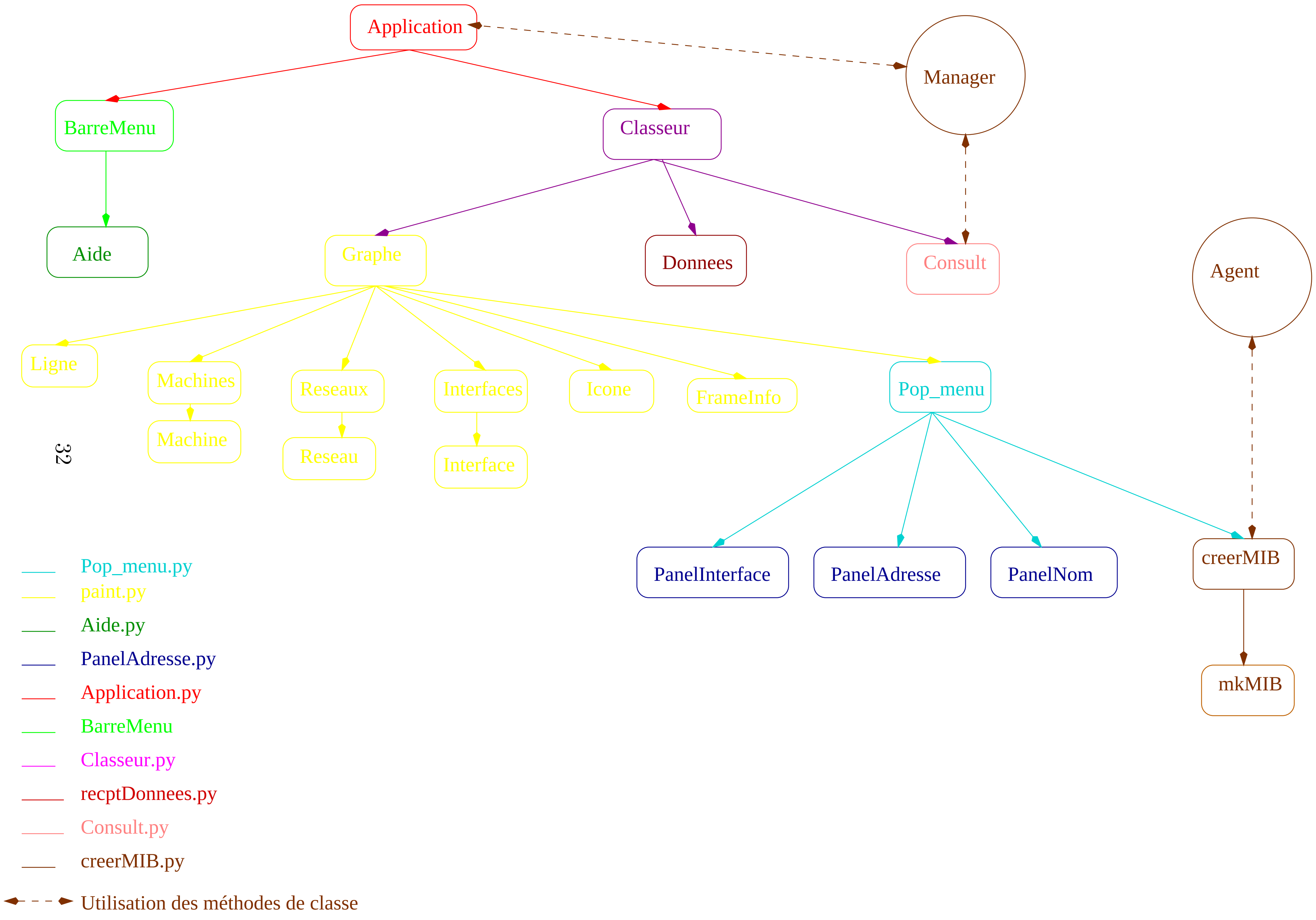

FIG. 4.2 – Hiérarchie des classes

4.3.1 Analyse

L'interface doit être capable de modéliser les différents types de réseaux. Par exemple, avec l'interface, nous devons pouvoir créer un réseau en étoile qui est interconnecté à un réseau en bus. Le premier problème qui se pose est la manière de représenter ce réseau.
Quelle Représentation choisir ?

– La première solution est de représenter le réseau sur le même plan. L'intérêt de cette structure est sa simplicité.

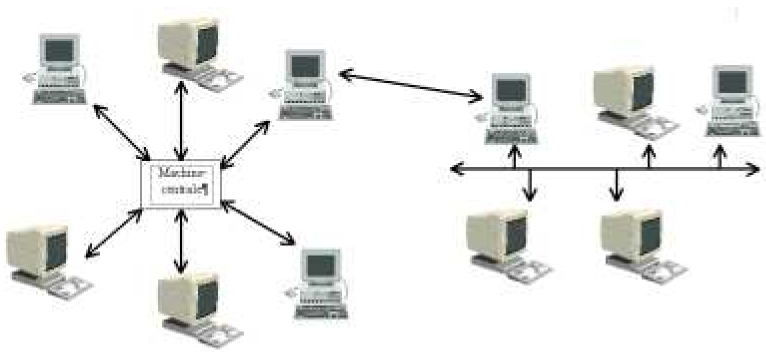

FIG. 4.3 – Première possibilité

– La deuxième solution consiste à représenter un sous réseau en un icône. Lorsque l'utilisateur veut avoir plus de renseignements sur un réseau, il suffit de développer l'icône. L'inconvénient d'une telle représentation, est qu'il faut parcourir les différents sous-réseaux

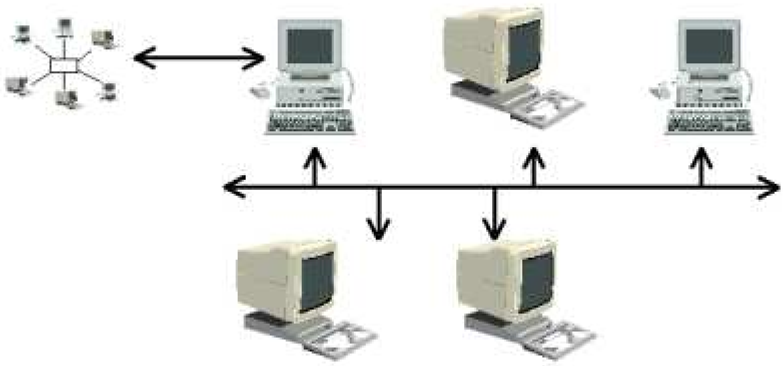

FIG. 4.4 – Seconde possibilité

pour voir la structure d'ensemble. Néanmoins, elle a l'avantage de permettre de bien distinguer les différents sous réseau et de rendre lisible la lecture de réseaux importants.

– Représentation choisie : la première interprétation est la plus facile à mettre en oeuvre, mais devient vite illisible dès que le réseau devient important. De plus, la notion de domaine n'apparaît pas. La deuxième représentation n'est pas pratique sur des petits réseaux, mais l'utilisation d'un tel logiciel est-elle utile sur de petits réseaux ? C'est pourquoi nous avons choisi la deuxième représentation.

4.3.2 L'ajout et la modification d'éléments (Réseaux, Machines)

L'ajout ou la modification sera possible grâce au bouton droit de la souris. Lorsque ce dernier sera appuyé, un popmenu permettra de choisir l'action à effectuer. Par exemple, si au moment du clic droit, la souris pointait un icône, les différents choix seront des modifications sur cet icône. Si la souris ne pointait aucun icône, alors les actions seront des ajouts.

4.3.3 L'interconnexion de réseaux

Les problèmes de l'interconnexion de réseaux sont essentiellement graphiques. En effet, lorsqu'on ajoute une interconnexion, il faut qu'elle soit visible (dans l'exemple ci-dessous) sur le réseau en étoile ou sur le réseau en bus. De plus, il faut sélectionner la machine et l'interface du

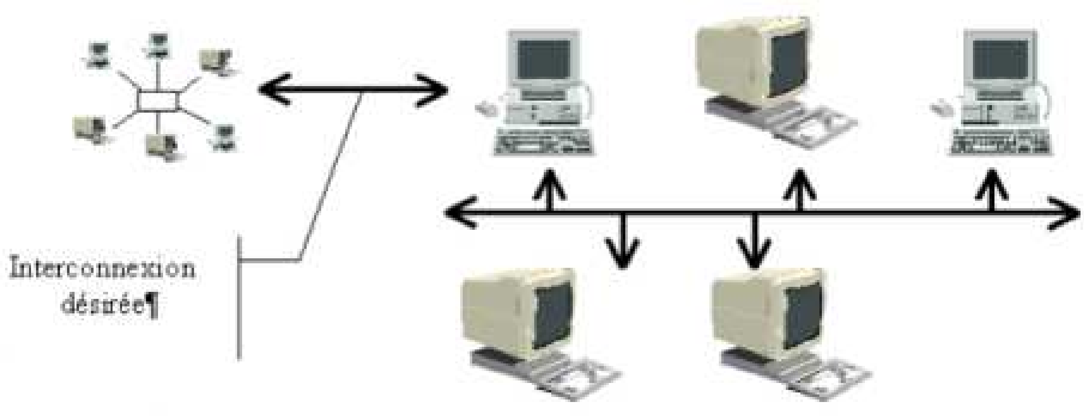

FIG. 4.5 – Interconnexion de réseaux

réseau extérieur afin d'avoir les informations nécessaires pour ajouter la passerelle sur l'autre réseau.

Voici un exemple de fenêtre qui demande les informations nécessaires pour réaliser l'interconnexion de réseaux :

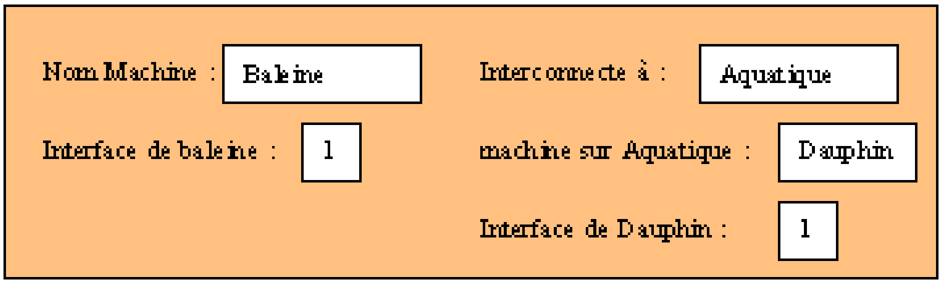

FIG. 4.6 – Un exemple d'interconnexion

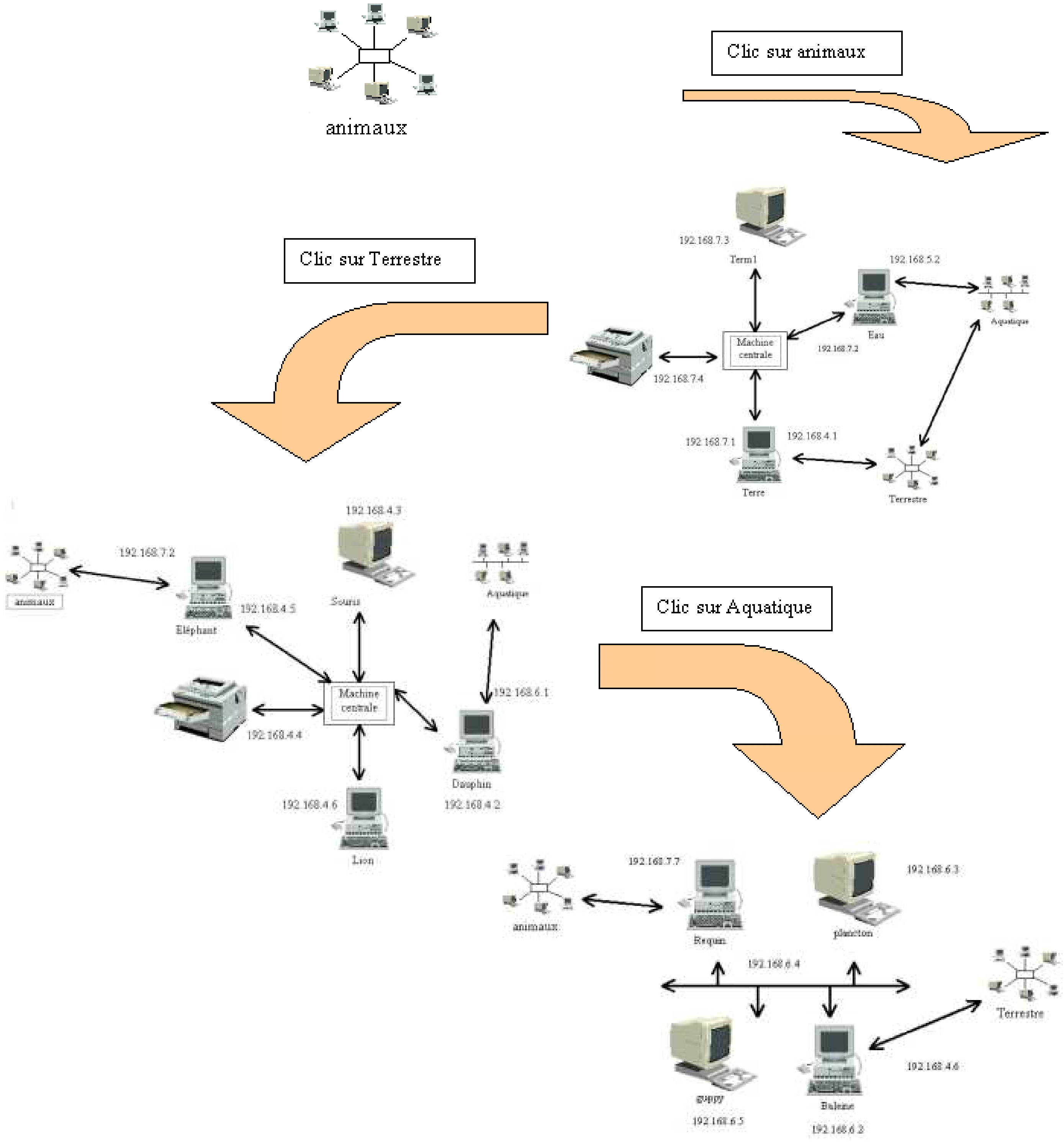

FIG. 4.7 – Un exemple d'utilisation

4.3.4 Mise en place de la structure de données

Afin de mieux comprendre le problème, nous allons créer un réseau fictif (cf FIG. 4.7) et essayer de voir les différents problèmes et les différentes solutions. Nous pouvons remarquer qu'une machine a toujours au moins une interface. Lorsque cette machine est connectée à un autre réseau, elle devient une passerelle et possède une adresse IP supplémentaire. Chaque adresse IP appartient à un réseau et possède des caractéristiques (BNC, RJ45, etc.). Nous appellerons cette structure " interface ".

Un réseau est constitué de machines connectées à lui-même, et peut contenir des machines qui appartiennent à d'autres réseaux. Nous ne pouvons donc pas modéliser un réseau par une liste de machines, car quand un paramètre de la machine changera, il faudra faire une recherche dans tous les réseaux afin de faire le changement partout. C'est pourquoi, toutes les machines seront dans une liste, et un réseau possédera une liste d'identifiants de la machine. Si un changement doit être fait sur une machine, il suffira juste de le changer dans la liste des machines.

4.3.5 Spécification

Type abstrait de données

Une interface est représentée par un identifiant, un type, une adresse, un masque de sous réseau et un Identifiant de réseau. Le type permet de savoir si une interface et de Type BNC ou RJ45 (ou autre chose). Cette variable est importante, car elle permet de vérifier que les interfaces sont bien branchées entre elles. En effet, il n'est pas possible de brancher une interface de type BNC sur une interface de type RJ45. Les interfaces seront stockées dans une structure qui regroupera toutes les interfaces. L'identifiant permettra de récupérer tous les informations de cette interface. L'identifiant réseau permet de savoir vers où l'interface est reliée.

```
Sorte : Interfaces

Constructeurs :

InitInterface : Id, Type, IdReseau, Adresse, Masque -> Interface

Opérations :

GetId : Interface -> Id     //Permet de récupérer l'identifiant
GetType : Interface -> Type     //Permet de récupérer le type (RJ45, BNC)
GetAddress : Interface -> Adresse  //Permet de récupérer l'adresse IP
GetMask : Interface -> Masque     //Permet de récupérer le masque de sous-réseau
GetIdReseau : Interface-> IdReseau //Permet de récupérer l'identifiant du réseau

Axiomes :

Id : Id, IdRes : IdReseau, T : Type
M : Masque, T : Type
```

```
GetId(InitInterface(Id, T, IdRes, Adr, M)) = Id
GetType(InitInterface(Id, T, IdRes, Adr, M)) = T
GetAddress(InitInterface(Id, T, IdRes, Adr, M)) = Adr
GetMask(InitInterface(Id, T, IdRes, Adr, M)) = M
GetIdReseau(InitInterface(Id, T, IdRes, Adr, M)) = IdRes
```

Une machine contient un identifiant, un nom, un type et une liste d'entiers. Le nom permet à l'administrateur de nommer une machine. Le type permet de savoir le rôle de la machine. Par exemple, on pourra il sera possible de différencier une imprimante d'un serveur ou d'un terminal. La liste d'entiers est en faite, une liste qui contient les identifiants des interfaces de la machine. Chaque interface de la machine qui est connectée à un réseau, permet de déterminer à quel réseau appartient la machine.

```
Sorte : Machine

Constructeurs :

InitMachine : Id, Chaine, Type, ListeInterfaces, Entier ->   Machine
AddInterface : Machine, Interfaces -> Machine


Opérations :

GetId : Machine -> Id
GetNom : Machine -> Chaine
GetType : Machine -> Type
GetInterfaces : Machine -> ListeInterfaces
GetNbInterfaces : Machine -> Entier

Axiomes :

M : Machine
Id : id, x : Interfaces, T : Type
Nom : Chaine, L : ListeInterfaces, nb : Entier

GetId(InitMachine(Id, Nom, T, L, nb)) = Id
GetId(AddInterface(M,x)) = GetId(M)

GetNom(InitMachine(Id, Nom, T, L, nb)) = Nom
GetNom(AddInterface(M,x)) = GetNom(M)

GetType(InitMachine(Id, Nom, T, L, nb)) = Type
GetType(AddInterface(M,x)) = GetType(M)
```

```
GetInterfaces(InitMachine(Id, Nom, T, L, nb)) = nil
GetInterfaces(AddInterface(M,x)) = cons(x, GetInterfaces(M))

GetNbInterfaces(InitMachine(Id, Nom, T, L, nb)) = 0
GetNbInterfaces(AddInterface(M,x)) = 1 + GetNbInterfaces(M)
```

Un réseau possède un Identifiant, un nom et une liste d'identifiants de machines. L'identifiant permet de retrouver le réseau parmis d'autres réseaux. En effet, les réseaux seront stockés dans une liste, l'identifiant permettra de retrouver un réseau facilement.

```
Sorte : Réseau

Constructeurs :

InitReseau : Id, Chaine, Type, ListeMachines -> Réseau
AddMachine :Réseau, Machine -> Réseau


Opérations :

GetId : Réseau -> Id
GetNom : Réseau -> Chaine
GetType : Réseau -> Type
GetMachines : Réseau -> ListeMachines

Axiomes :

R : Réseau
Id : Id, x : Machine, T : Type
Nom : Chaine, L : ListeMachines

GetId(InitReseau(Id, Nom, T, L)) = Id
GetId(AddMachine(R,x)) = GetId(R)

GetNom(InitReseau(Id, Nom, T, L)) = Nom
GetNom(AddMachine(R,x)) = GetNom(R)

GetType(InitReseau(Id, Nom, T, L)) = T
GetType(AddMachine(R,x)) = GetType(R)

GetMachines(InitReseau(Id, Nom, T, L) = nil
GetMachines(AddMachine(R,x)) = cons (x, GetMachines(R))
```

4.3.6 Conception

Le diagramme ci-dessous est une représentation UML de l'interface. Un graphe se compose

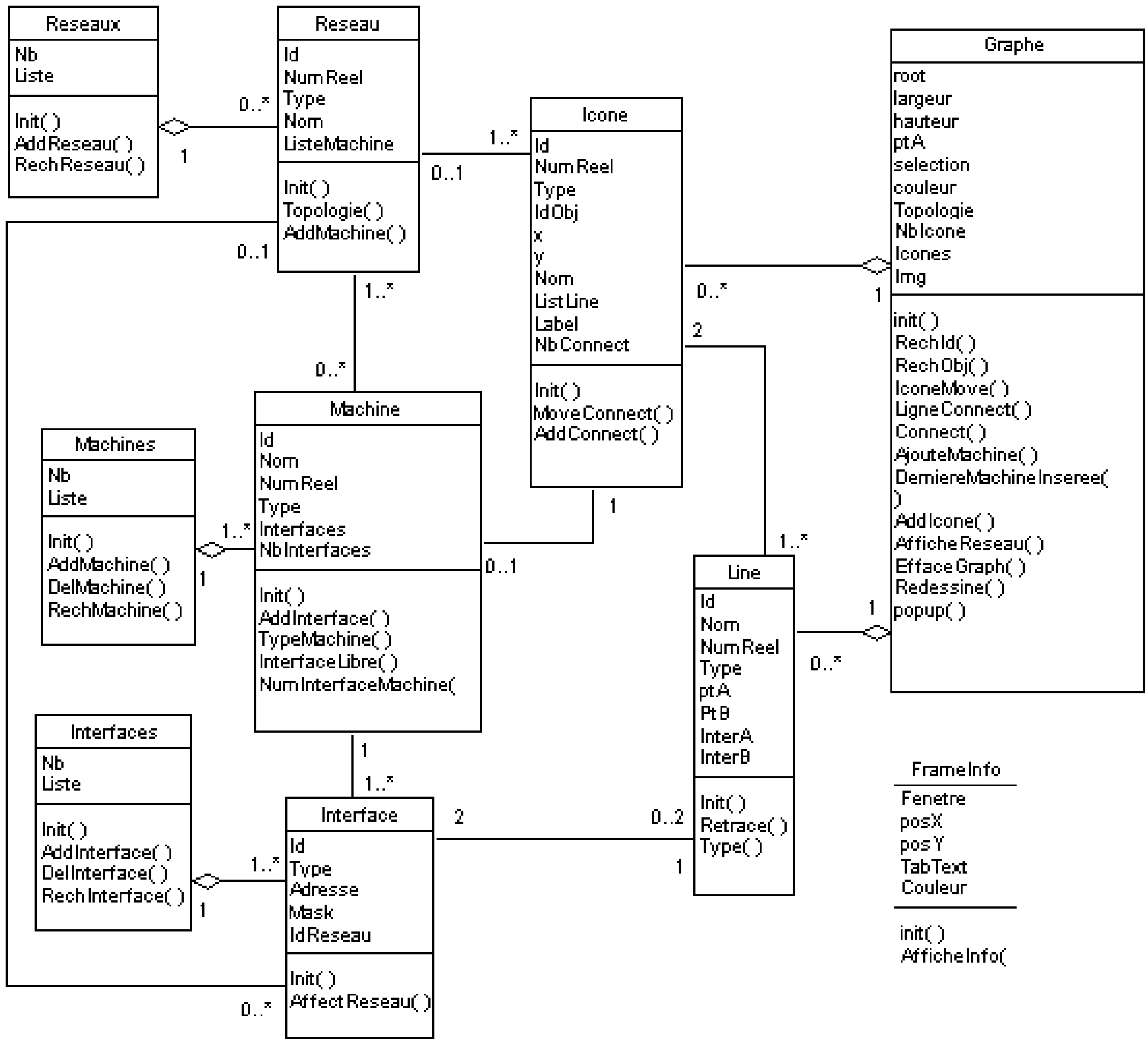

FIG. 4.8 – Diagramme UML

d'une liste d'icônes et d'une liste de lignes. La liste de lignes est en réalité la liste des connexions entre des éléments. Ces éléments sont des objets de type icône qui peuvent être des machines ou des réseaux. Une machine, en elle-même, n'appartient pas à un réseau. En effet, ce sont leurs interfaces qui déterminent le(s) réseau(x) auquel(s) appartient la machine. L'objet " FrameInfo " permet l'affichage d'informations sur un élément du réseau. Tous les objets ont accès à cet objet.

4.3.7 Gestion des listes

Pour accéder à un élément d'une liste, Python utilise une notation qui peut induire en l'erreur. En effet, il utilise une notation qui ressemble à celle des vecteurs.

```
Tab=['a','b','c','d']
Print Tab[2]
```

Dans cet exemple le résultat va être c, mais il faut faire attention car si jamais un des éléments Tab[1] ou Tab[2] est supprimé, le résultat sera 'd'. Ceci nous empêche de faire un index sur l'identifiant. Par exemple :

```
Reseaux=[['net1',[1,4]],['net2',[2,3]]]
Machines=['toto','titi','tata','palin']
```

Cela veut dire que les machines 'titi' et 'tata' appartiennent au réseau 'net2' et que la machine 'toto' appartient au réseau 'net1'. Si aucune fonction de suppression n'est faite, alors cette structure fonctionne correctement. Mais, si on supprime la machine 'toto', nous serons alors obligé de recalculer la variable Reseaux :

```
Reseaux=[['net1',[3]],['net2',[1,2]]]
Machines=['titi','tata','palin']
```

Pour éviter cela, nous ajoutons au nom du réseau son identifiant :

```
Reseaux=[[1,'net1',[1,4]],[2,'net2',[2,3]]]
Machines=[[1,'toto'],[2,'titi'],[3,'tata'],[4,'palin']]
```

Si nous supprimons 'toto', cela donne :

```
Reseaux=[[1,'net1',[4]],[2,'net2',[2,3]]]
Machines=[[2,'titi'],[3,'tata'],[4,'palin']]
```

4.3.8 Gestion des événements

Trois types d'événements sont gérés par l'interface : les clics gauche, milieu et droit. Le clic gauche permet de déplacer les icônes. Pour ce faire, lorsque Python détecte un clic droit, nous lance une fonction qui recherche tous les icônes qui sont sous le curseur souris. Lorsque la souris se déplace, il prend le dernier icône de la liste et le déplace. Si la liste est vide, il ne se passe rien. Le clic du milieu permet d'entrer dans un réseau. A chaque clic du milieu, le programme recherche si un icône de type réseau est sous le pointeur souris. Le clic droit fait apparaître un menu contextuel. Le Pop Menu diffère selon la position du pointeur souris et selon le niveau de profondeur dans le réseau. Si le pointeur souris pointe sur une machine, alors le menu contextuel concernera essentiellement la modification des machines. S'il n'y a pas d'icône, alors le pop menu permettra des opérations d'ajout.

4.3.9 Chargement/Sauvegarde

Le chargement et la sauvegarde sont des éléments indispensables à l'interface. Ils permettent de sauvegarder le travail qui a été fait et de le recharger plus tard. Pour faciliter les procédures de lecture et d'écriture, le fichier généré est un fichier Python. Pour le lire, il suffit de faire : import « nom de fichier ». La génération d'un fichier consiste à stocker toutes les variables nécessaires sous forme de listes. Pour faire la sauvegarde, nous utilisons 5 listes :
- net
- machine
- interface
- icône
- root

La liste 'net' permet de sauvegarder toutes les informations sur les réseaux (adresse, topologie, nom, liste machines). La liste 'machine' contient toutes les informations de toutes les machines (Nom, liste interfaces, type machine). La liste 'interface' permet de sauvegarder toutes les interfaces de toutes les machines (type d'interface, adresse, masque, numéro réseau). La liste icône contient les listes d'icônes de tous les réseaux. Elle permet de reconstituer exactement la position de chaque icône comme l'utilisateur l'avait définie. La liste 'root' est la liste icône de la fenêtre principale.

Pour éviter que la procédure de chargement ne boucle indéfiniment, un booléen a été ajouté sur la structure réseau. En effet, lors d'une interconnexion, les réseaux s'appelle mutuellement, le booléen permet d'éviter ce piège.

4.4 L'interrogation

Après avoir réalisé une interface graphique permettant de représenter l'interconnexion de réseaux, nous avons poursuivi le projet dans le sens de la simulation. En effet, une fois la visualisation du réseau réalisée, nous devions pouvoir gérer l'architecture ainsi créée. Pour cela, il était nécessaire de travailler sur les MIBs des équipements du réseau.

Nous avons donc intégré et remanié les fonctions implémentées par le projet de 1999. L'utilisation de requêtes sur une MIB virtuelle nécessite deux étapes : la création et l'association d'une MIB à un équipement, puis le lancement d'un agent, utilisant cette MIB, permettant de traiter les différentes requêtes disponibles.

4.4.1 Surveillance

Pour faire face à la multiplicité des différents types d'interconnexion, et donc à la diversité des problèmes pouvant intervenir, il nous faut un language simple de Questions/Response compréhensible par tout le monde. SNMP est un protocole d'administration adéquat, il permet, grâce à TCP/IP, d'interroger les différentes MIB des agents, et ainsi, de se faire une idée de l'état du réseau. L'analyse des données renvoyées par les agents est effectuée par le manager, qui lui, décide des suites à mettre en œuvre. Il nous est donc possible d'interroger un agent ayant une MIB et d'analyser les réponses renvoyées.

Lorsque nous choisissons la page du classeur *Requetes*, des listes déroulantes nous permettent de choisir la requête, ainsi que la variable, et une fenêtre nous permet de saisir le numéro de l'interface. La requête ainsi construite, il est maintenant possible de la renvoyer au serveur agent, qui lui, nous renverra la réponse dans l'onglet *Données*.

4.4.2 Création et affectation d'une MIB

Le but de cette opération est de pouvoir utiliser des agents virtuels. Lors d'une utilisation réelle (mode physique), le démon snmp (snmpd) permet de consulter et de modifier une MIB. Dans le cas d'une simulation, il fallait créer un "pseudo-démon" qui réalise les mêmes opérations, mais sur une vue de MIB. C'est pourquoi, nous avons créé, à l'aide de la commande snmpwalk, un fichier contenant les variables de base d'une MIB. Nous avons ensuite réalisé une interface permettant d'afficher ces variables, associées à des valeurs par défaut que nous pouvons modifier à notre convenance. Une fois ces modifications réalisées, l'utilisateur n'a plus qu'à confirmer pour générer la vue MIB correspondante.

Le travail qui s'effectue ensuite en tâche de fond, consiste à récupérer ces différentes valeurs et à les mettre en relation avec les descriptions apportées par le fichier RFC, cf RFC 4.4.3, correspondant à l'équipement traité. Nous obtenons alors une MIB qui peut être consultée par l'agent, l'utilisateur peut alors effectuer les opérations qu'il désire sans pour autant risquer d'endommager le réseau réel.

4.4.3 Diversité des types d'agents

Il existe un nombre important de RFCs, chacun définissant un aspect spécifique des réseaux. Nous répertorions, dans le tableau ci-dessous, les RFCs de base, correspondant aux composantes d'un réseau, qui nous intéressent. Nous avons repris les RFCs les plus généraux ; une liste détaillant les RFCs se trouve à l'URL suivante : www.cis.ohio-state.edu/htbin/rfc/INDEX_rfc.html

RFCs	Descriptions
1213	Base d'information de gestion pour la gestion d'un réseau basé sur le protocole Internet TCP/IP : MIB-II.
1230	MIB pour un réseau en Bus IEEE 802.4.
1231	MIB pour un réseau en anneau IEEE 802.5.
1239	Réassignement des MIBs expérimentales en MIBs standards.
1493	Définitions des objets gérés pour les ponts.
1516	Définitions des objets gérés pour les répéteurs IEEE 802.3.
1548	Le protocole Point à Point (PPP).
1643	Définitions des objets gérés pour les interfaces de type Ethernet.
1660	Définitions des objets gérés pour les imprimantes parrallèles.
1812	Exigences pour les Routeurs IP Version 4.
1907	Base d'informations de gestion pour la Version 2 du protocole de gestion de réseau simple : SNMPv2.

Chapitre 5

Evolutions

5.1 Algorithme de découverte

La topologie du réseau variant selon les besoins des utilisateurs et de l'administrateur, il nous faut trouver une méthode nous permettant de représenter, le plus fidèlement, la structure du réseau. Il est possible pour cela d'utiliser un algorithme de découverte basé sur le protocole ICMP. L'algorithme se divise en neuf étapes :

1. envoyer un message ICMP echo request sur toutes les routes connues,
2. construire toutes les routes grâce aux adresses IP découvertes,
3. determiner le masque de réseau pour chaque adresse IP,
4. pour chaque adresse IP, envoyer un paquet UDP et récupérer l'adresse de retour afin de connaître les machines à plusieurs interfaces,
5. identifier les réseaux et les sous-réseaux,
6. identifier les machines à plusieurs interfaces,
7. connecter les adresses IP aux réseaux,
8. fusionner les adresses des machines à plusieurs interfaces,
9. ajouter les nouveaux noeuds découverts.

Il n'est donc plus nécessaire de garder en dur la structure du réseau. Il suffit de relancer périodiquement l'algorithme afin de s'apercevoir, au plus vite, des changements. Cela permet de rajouter de la portabilité et de diminuer la taille du programme manager, car il peut se trouver sur n'importe quelle machine du réseau, et ainsi connaîtr la structure entière. cf [Algo]

5.2 Utilisation des données SNMP

Nous pouvons imaginer qu'en mode physique, lorsque le nom et l'adresse sont connus du manager, le reste des informations désirées afin de construire une image du réseau correcte se trouvent dans les MIBs des agents. Il suffit, par exemple, pour connaître le nombre d'interfaces d'un agent, de lui envoyer une requête spécifique et d'analyser la réponse. Si les adresses IP des interfaces sont différentes, alors nous sommes dans le cas d'une interconnexion. Avec tout un jeu de questions/réponses, il sera ainsi facile de déterminer les sous-réseaux, les passerelles et les machines spécifiques.

5.3 Agents intelligents

Les agents intelligents sont une famille d'applications logicielles qui permettent d'automatiser un certain nombre de tâches. Le dénominateur commun entre les différentes familles d'agents est qu'ils ont un certain degré d'autonomie, qu'ils mènent des opérations au nom de l'utilisateur et qu'ils ont une connaissance, au moins partielle, des objectifs et du profil de l'utilisateur.Un réseau pourra être administré par un de ces agents intelligents, qui navigerait entre les stations. Sur chaque machine, l'agent vérifiera le bon fonctionnement du système, et selon les résultats de ces analyses, il pourra décider des suites à fournir aux problèmes (envoie de mails aux utilisateurs ou désactivation d'un port surchargé). Après quoi, il stockera toutes les informations dans une Base de Données, avant de les analyser ultérieurement, ou pour les fournir au super-administrateur s'il en fait la demande, puis, il s'auto-téléchargera sur une autre machine distante et ainsi de suite cf [AgentIntel].

Conclusion

La comparaison entre le Cahier des charges (cf 1) et l'état du projet réalisé nous donne le tableau suivant :

Réf.	Etat de réalisation
$[CO1]$	réalisé
$[CO2]$	réalisé
$[CO3]$	réalisé
$[MV]$	réalisé
$[MVD]$	partiellement réalisé
$[MP]$	partiellement réalisé
$[CVD]$	réalisé
$[CVS]$	réalisé
$[CVC]$	réalisé
$[CCA]$	réalisé
$[CCS]$	pas réalisé
$[CCM]$	pas réalisé
$[CAI]$	réalisé
$[CAM]$	réalisé
$[CAS]$	réalisé
$[AG]$	partiellement réalisé
$[SNT]$	pas réalisé
$[SMT]$	pas réalisé
$[CDM]$	réalisé

La réalisation de ce projet nous a permis d'accentuer les connaissances que nous avions sur les réseaux et leurs administrations de façon concrète. Il nous a aussi fallu faire l'apprentissage du langage Python et de l'editeur de texte Latex. Nous avons dû nous immerger dans les projets antérieurs et en retirer le plus d'informations possibles sur l'application déjà réalisée.

Afin de faciliter la reprise du travail, il nous a semblé essentiel de clarifier, au maximun, les taches effectuées par l'application. Nous avons pour cela réalisé un Manuel de l'utilisateur, ainsi qu'un Manuel du programmeur ce qui permettra, aux prochain(e)s étudiant(e)s, une plus rapide mise au point sur le travail efféctué.

Ce projet nous a permis d'apprécier les besoins du travail en équipe, tout en démontrant sa nécessité. De plus, nous avons pu nous rendre compte de l'utilité d'une bonne organisation du temps en fonction de chacun.

Les connaissances acquises durant ce projet, nous permettrons de participer au développement d'applications réseau tant au niveau de l'administration qu'au niveau de la maintenance.

Glossaire

ATM	:	Transfert en Mode Asynchrone
ASN.1	:	Abstract Syntax Notation One
BER	:	Basic Encoding Rules
BNF	:	Backus Normal Form
CSMA/CD	:	Carrier Sense Multiple Access with Colision Detect
CMIP	:	Common Management Information Protocol
HDLC	:	High Level Data Link Control
ICMP	:	Internet Control Message Protocol
IP	:	Internet Protocol
ISO	:	International Standarts Organization
MIB	:	Management Information Base
LAN	:	Local Area Network
LAP-B	:	Link Access Procedure- Balanced
OID	:	Object IDentifier
OSI	:	Open Systems Interconnexion
PDU	:	Protocol Data Units
RFC	:	Request For Comments
SMDS	:	Service de Données Multimégabits Synchrone
SMI	:	Structure of Management Information
SNMP	:	Simple Network Management Protocol
TCP/IP	:	Transport Control Protocol/Internet Protocol
UDP	:	User Datagram Protocol
WAN	:	Wide Area Network

Bibliographie

[Toutain 99] Laurent Toutain, *Réseaux locaux et Internet : des protocoles à l'interconnexion* deuxième édition française, Hermes, France,1999.

[Hunt 97] Craig Hunt, *TCP/IP : Administration de réseau* deuxième édition française, O'REILLY, Etats-Unis, 1997

[Lutz 00] Mark Lutz & David Ascher, *Introduction à Python* édition française, O'REILLY, Etats-Unis, 2000

[Lutz 96] Mark Lutz, *Programming Python* O'REILLY, Etats-Unis, 1996

[Rose 95] Marshall Rose, *Gestion des réseaux ouverts, SNMPv2* InterEditions, Paris, 1995

[TER A] M. CHATTON, D. MAURANGE, X. PAQUET *Conception d'un outil pédagogique de gestion de réseaux* Projet 1998/1999

[TER B] C. BESSEAU, D. TURPAULT *Conception d'un outil pédagogique de gestion de réseaux* Projet 1998/1999

[Algo] J. Schönwälder & H. Langendörfer *How to Keep Track of Your Network Configuration* TU Braunschweig, Germany

[Pmw] Pmw 0.8.3 Python Megawidgets *http ://www.dscpl.com.au/pmw/*

[Tut] Tutoriel Python *http ://perso.club-internet.fr/olberger/python/doc/tut/tut.html*

[AgentIntel] pointeur sur un site *http ://ns1.tele.ucl.ac.be/ELEC2920/1997/agentsP/complet.html*