TABLE DES MATIERES

ABREVIATION

ARP	Address Resolution Protocol
DNS	Domain Name Server
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
DoD	Department of Defense
DNS	Domain Name Server
EGP	Exterior Gateway Protocol
FTP	File Transfer Protocol
GNU	GNU is Not Unix
HTTP	Hyper Text Transfer Protocol
IBM	International Business Machines
ICMP	Internet Control Message Protocol
IGP	Interior Gateway Protocol
ISO	International Standards Organization
IP	Internet Protocol
LDAP	Lightweight Directory Access Protocol
MAC	Medium Access Control
NetBIOS	Network Basic Input Output System
NSS	Name Service Switch
OSI	Open System Interconnection
PAM	Pluggable Authentification Service

POP	Post Office Protocol
SMTP	Simple Mail Transfer Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
UDP	User Datagram Protocol

INTRODUCTION GENERALE

Un réseau en général est le résultat des connexions de plusieurs machines entre elles, afin que les utilisateurs et les applications qui fonctionnent sur ce dernier puissent échanger des informations.

Les réseaux sont nés d'un besoins d'échange des informations de manière simple et rapide entre des machines. Une grande machine centralise les informations nécessaires pendant un travail. Elle les duplique à tous les utilisateurs et les applications qui ont accès à ces informations.

Cependant, un réseau informatique doit toujours faire face à des problèmes, les attaques venant de l'extérieur. Actuellement linux est nécessaire pour éviter ces problèmes.

Ce mémoire de fin d'étude qui s'intitule «configuration d'un réseau local sous linux avec samba» va nous permettre de mieux approfondir l'installation et la maintenance d'un réseau en utilisant le système d'exploitation linux. Ce document comporte trois chapitres :

Premièrement, nous allons parler du réseau informatique en général. Ensuite, nous détaillerons les réseaux TCP/IP et ses protocoles. Et enfin, la réalisation d'un réseau sous linux.

CHAPITRE 1
GENERALITE SUR LES RESEAUX INFORMATIQUES

1.1 Introduction

Les informations sont assez souvent inutiles à moins de pouvoir être partagées avec d'autres personnes.

Le réseau informatique est le véhicule qui permet aux données d'être facilement mis en commun et l'Internet en est l'actuel aboutissement. Les réseaux informatiques qui permettent à leur origine de relier des terminaux passifs [3] à de gros ordinateurs centraux autorisent à l'heure actuelle l'interconnexion de tous types, d'ordinateurs que ce soit de gros serveurs, des stations de travail, des ordinateurs personnels ou de simples terminaux graphiques. Les services qu'ils offrent font partie de la vie courante des entreprises et administrations (banques, gestion, commerce, bases de données, recherche, etc.) et des particuliers (messagerie, loisirs, services d'informations par minitel et Internet…).

1.2 Les réseaux client /serveur

1.2.1 Serveur

Le serveur est une machine destinée à partager et à gérer les ressources pour les utilisateurs connectés. C'est lui qui détient donc les ressources particulières [11] [12]. Cette grande machine héberge des logiciels serveurs.

1.2.2 Client

Un ordinateur d'un utilisateur qui a besoins des informations centralisées dans un serveur en se branchant via un réseau.

1.2.3 Les topologies des réseaux

1.2.3.1 Topologies en bus

La plupart des réseaux locaux utilise la topologie en bus, dans cette topologie tous les ordinateurs sont reliés à une même ligne de transmission par l'intermédiaire de câble, généralement coaxial.

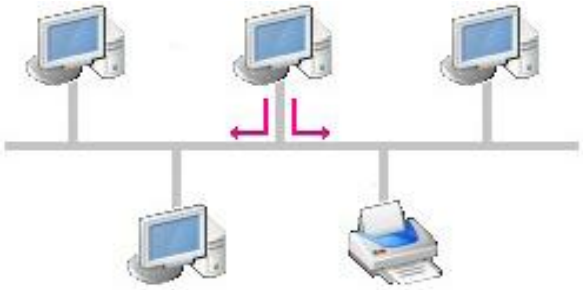

Figure 1.01 : *Topologie en BUS*

Le principe du "BUS" est extrêmement simple :

- Un conducteur unique représente le réseau.
- Chaque extrémité est bouclée sur un "bouchon" dont l'impédance électrique est égale à l'impédance caractéristique du conducteur, ceci afin d'éviter les réflexions des signaux en bout de câble.
- Chaque poste est "piqué" sur ce bus au moyen d'un "T" de raccordement.

Il existe deux types de réseau en bus, qui se distinguent par son protocole d'accès et la nature de son support :

- Ethernet : le plus utilisé.
- Apple Talk

1.2.3.2 Les topologies en étoile

Dans une topologie en étoile, les ordinateurs du réseau sont reliés à un système matériel appelé Hub ou concentrateur. Chaque machine est reliée par un câble pair torsadé à ce concentrateur. Ce dernier est composé d'un certain nombre des jonctions pour connecté les câbles.

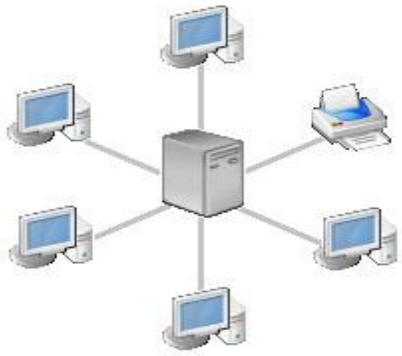

Figure 1.02 : *Topologie en Etoile*

La topologie en étoile est plus onéreuse par rapport à la topologie en bus parce qu'elle a un matériel supplémentaire.

1.2.3.3 Topologie en anneau

Les informations circulent en sens unique sur la boucle pour éliminer les collisions entre différentes messages. Pendant son passage, chaque nœud examine l'adresse de son destinataire : le message est accepté si c'est pour lui ; sinon, le message est régénéré par ce nœud et suit son parcours vers le nœud suivant.

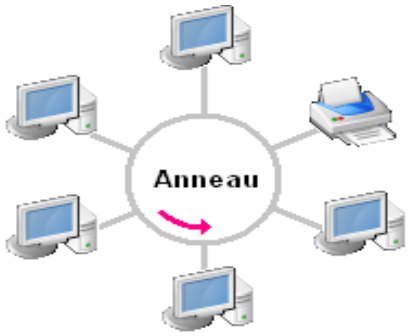

Figure 1.03 : *La topologie en Anneau*

En topologie logique, cette topologie est comme le Token Ring de la compagnie IBM (International Business Machines).

On doit avoir une station particulière du réseau, appelé « moniteur actif ». Ce dernier a le rôle de vérifier la présence des autres stations en lançant un jeton au démarrage. [11][4][6][1]

1.3 Architecture client /serveur

1.3.1 *Présentation de l'architecture d'un client/serveur*

Lors des applications fonctionnant entre les machines clientes connectés au réseau et le serveur, des ressources sont partagés aux clients par un serveur. Ce dernier est prêt aux services pour les clients.

Chaque machine a besoin d'un logiciel pour les applications, un logiciel serveur pour plusieurs logiciels clients. S'il s'agit des partages des fichiers, c'est le cas du client FTP (File Transfer Protocol) et lorsqu'on parle du courrier électronique, c'est le cas du client de messagerie.

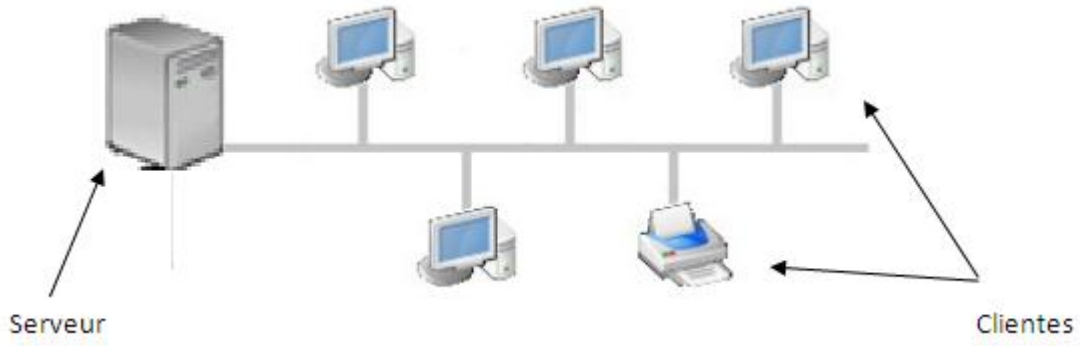

Figure 1.04 : *Architecture client/serveur*

1.3.2 Fonctionnement d'un système client/serveur

Voici le schéma simplifié pour le fonctionnement d'un système client/serveur:

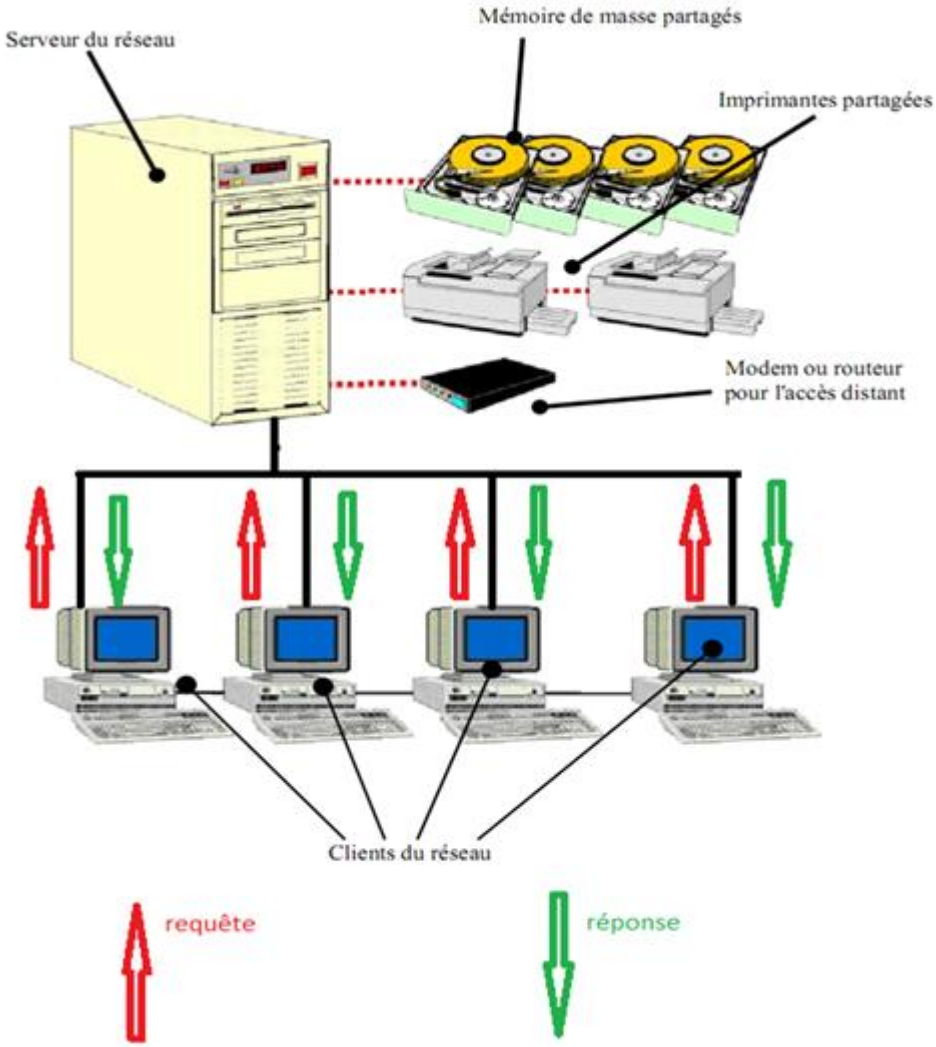

Figure 1.05 : *Communication Client/Serveur*

- Le client émet une requête vers le serveur grâce à son adresse et le port, qui désigne un service particulier du serveur
- Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine client et son port

1.3.3 Avantages de l'architecture client/serveur

Le modèle client/serveur est toujours très applicable quand on veut réaliser des réseaux fiables, ses principaux atouts sont [12]:

- **des ressources centralisées**: Les serveurs sont conçus pour le partage de ressources et ne servent pas de station de travail. Il suffit de les dimensionner en fonction de la taille du réseau et du nombre de clients susceptibles de s'y connecter.
- **une meilleure sécurité:** les systèmes d'exploitation de serveurs proposent des fonctions avancées de sécurité que l'on ne trouve pas sur les réseaux "peer to peer".
- **une administration au niveau serveur**: les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés. Un administrateur gère le fonctionnement du réseau et les utilisateurs n'ont pas à s'en préoccuper.
- **un réseau évolutif**: grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modifications majeure

1.3.4 Inconvénients du modèle client/serveur

L'architecture client/serveur a tout de même quelques lacunes parmi lesquelles:

- **un coût élevé.** Le coût est évidemment plus élevé puisqu'il faut la présence d'un ou de plusieurs serveurs.
- **un maillon faible**: le serveur est le seul maillon faible du réseau client/serveur, étant donné que tout le réseau est architecturé autour de lui! Si un serveur tombe en panne, ses ressources ne sont plus disponibles. Il faut donc prévoir des solutions plus ou moins complexes, plus ou moins onéreuses, pour assurer un fonctionnement au minimum en cas de panne. [12]

1.4 Méthodes d'accès

Toutes machines et autres équipements connectés au même réseau doivent suivre des ensembles de règle. Il existe deux méthodes d'accès qui sont la méthode d'accès aléatoire et la méthode d'accès déterministe ayant toutes les deux leurs caractéristiques distinctes. Voyons donc ces deux méthodes l'une après l'autre. [2]

1.4.1 Aléatoire

Pour la topologie en bus, on utilise la méthode d'accès CSMA/CD, dit aléatoire. Dans cette topologie, les trames circulent simultanément sur le BUS, sur lequel les trames peuvent s'interférer, ce qui entraine la collision. Le CSMA/CD permet la détection de cette collision et de solutionner l'existence de ces interférences en contrôlant les conflits entre les machines qui tentent d'émettre en même temps pour que le réseau fonctionne bien.

On va détailler cette méthode :

Si une station souhaite émettre une trame sur le BUS, doive d'abord voir s'il n'y en a aucune qui circule. On utilise un dispositif situé dans une carte réseau, qui mesure le niveau d'un signal analogique. Ce dispositif détecte si une trame circule à cet instant. A condition que cette station est connectée par le réseau. C'est le « Carrier Sens » ou « détection de porteuse » :

Si une trame est détectée la station attend et continue à écouter le support.

Si aucune trame n'est détectée

La carte réseau de la station attend que la période d'absence de trame soit supérieur ou égal au temps inter trame (960 ns pour un débit de 100 Mbips).

S'il n'y a pas toujours de trame qui passe, la station émet des informations en déclenchant l'écoute d'une collision possible, c'est le « Collision Detection ».

1.4.2 Déterministe

Cette méthode d'accès est essentiellement utilisée dans les réseaux organisés selon la topologie en anneau.

Un jeton circule en permanence, toujours dans le même sens, de poste en poste (ou nœud) de l'anneau. Pour qu'une machine puisse émettre une trame sur l'anneau, elle doit s'emparer du jeton (qui devient occupé) lorsqu'il passe à sa portée, ce qui peut nécessiter l'attente du temps nécessaire pour qu'il parcoure un tour complet. La trame émise traverse chaque machine qui contrôle si elle lui est destinée (en lisant le champ destination de la trame) si ce n'est pas le cas, elle la transmet à la machine suivante après l'avoir régénérée (simple remise en forme électrique du signal) ou la marque mauvaise si elle contient des erreurs.

Enfin, la machine à qui était destinée la trame la transmet aux couches supérieures qui vont décoder son sens et la traiter. La trame continue son parcours et revient à la machine qui l'a émise ; celle-ci va vérifier si elle a été correctement reçus par la machine destinataire, puis la détruit.

Dans les systèmes les plus basiques, c'est seulement à ce moment que le jeton est libéré, afin qu'il puisse être utilisé par une autre machine (il a fallu attendre un tour complet) ; dans les systèmes plus évolués, le jeton est libéré dès que la trame est parvenue à la machine destinataire.

1.5 Avantages et inconvénients des topologies

1.5.1 Topologies en bus

1.5.1.1 Avantage

Après avoir vu les divers constituants, on constate que ce procédé est peu coûteux, facile et rapide à mettre en œuvre, c'est ça ses avantages par rapport aux autres topologies.

1.5.1.2 Inconvénient

Un défaut qui n'est pas toujours visible : problème connectique et c'est tout le réseau qui dévient inopérant.

1.5.2 Topologies en étoile

1.5.2.1 Avantage

Si un lien vient se rompre, seul ce PC connecté est absent du réseau. Pour un fonctionnement beaucoup plus sûr que le BUS.

Facile d'ajouter des postes au réseau.

1.5.2.2 Inconvénient

La longueur totale de câble mise en œuvre est importante.

- Au voisinage du HUB, on obtient un faisceau de câbles imposant.
- Le coût est tout de même plus élevé que dans une architecture BUS.

1.5.3 Topologie en anneau

1.5.3.1 Avantage

La régénération du message permet à un réseau en anneau d'être plus important qu'un réseau en étoile ou en bus.

1.5.3.2 Inconvénient

Toutes les stations doivent fonctionner en permanence pour que les nœuds puissent régénérer les trames. Face à ce problème, pour la fiabilité du système : on doit utiliser deux anneaux qui ont un sens différant. [12] Ce qui entraine l'augmentation des couts.

1.6 Le modèle OSI de l'ISO

Dans un réseau, l'information peut circuler de deux stratégies :

L'information est envoyée de façon complète qui n'est pas très utile car le risque d'erreur est fort et complexe à résoudre.

L'information est fragmentée en petits morceaux, chaque paquet est envoyé séparément sur le réseau et elles sont ensuite rassemblées sur la machine destinataire. On parle ici de réseau à commutations de paquets.

Le fonctionnement du réseau à commutation de paquet est décrit par le modèle OSI (Open System Interconnexion) : c'est le modèle en sept couches.

Le modèle OSI est développé en 1978 par l'ISO (International Organization of Standards). Les réseaux basés du modèle OSI peuvent dialoguer par le même langage. Chaque couche, excepté la première et la dernière, prend en charge les services de la couche inférieure et propose des services à la couche supérieure. [13]

1.6.1 les sept couches du modèle OSI

1.6.1.1 Couche Physique

La couche physique a pour rôle de transmettre des bits sur un canal de transmission. Ces bits peuvent être des bases de données ou des fichiers à transférer. Cette couche doit normaliser les interfaces électriques (un bit 1 doit être représenté par une tension de 5 V, par exemple) et mécanique (forme des connecteurs, de la topologie...).

1.6.1.2 Couche Liaison de données

La couche liaison de données reçoit les bits de la couche physique et les rassemble pour avoir son propre unité logique appelé trame. Elle fait circuler cette trame d'un nœud vers un autre nœud du réseau. [2] Les bits de trame ont chacun ses significations, il existe 3 parties :

- Les bits pour indiquer les différents types de trame.
- Les bits qui représentent les données transmises par les trames.
- Les bits qui détectent et contrôlent les erreurs dans la trame de liaison de donnée.

C'est donc la première couche qui gère les erreurs de transmission.

1.6.1.3 Couche Réseau

Cette couche assure toutes les fonctionnalités de services entre les entités du réseau, c'est à dire : l'interconnexion des différents sous-réseaux entre eux, l'adressage, le routage, le contrôle de flux, la détection et les corrections d'erreurs non résolues par la couche liaison pour préparer le travail de la couche transport.

L'unité d'information de la couche réseau est le paquet.

1.6.1.4 Couche Transport

Elle complète alors les services de la couche réseau. Le rôle principal de la couche transport est de prendre les messages de la couche session, de les découper s'il le faut en unités plus petites et de les passer à la couche réseau, tout en s'assurant que les morceaux arrivent correctement de l'autre côté. Cette couche effectue donc aussi le réassemblage du message à la réception des morceaux.

L'unité d'information de la couche réseau est le message.

1.6.1.5 Couche Session

Cette couche fournit aux entités de la couche présentation les moyens d'organiser et de synchroniser les dialogues et les échanges de données. Il joue alors le rôle de transmettre les informations de programme en programme.

1.6.1.6 Couche Présentation

Cette couche gère la manière dont les données sont échangées entre les applications c'est-à-dire les formats des données.

1.6.1.7 Couche Application

Cette fois c'est dans cette couche que les applications communiquent ensemble (courrier électronique et transfert des fichiers.

1.7 Principe d'interconnexion des réseaux différents

1.7.1 Objectifs

Les réseaux informatiques ont pris une telle importance qu'il devient de plus en plus nécessaire de les interconnecter. [11] Un organisme comportant plusieurs secteurs d'activité pourrait disposer d'un réseau spécifique pour chaque activité, tous ces réseaux pouvant être interconnectés pour une meilleure distribution de l'information.

1.7.2 Les outils d'interconnexion

1.7.2.1 Les répéteurs

Ils sont utilisés pour prolonger le support de communication. Ils connectent alors deux segments du réseau. Il intervient au niveau de la couche physique.

1.7.2.2 Les ponts

Les ponts sont utilisés pour relier deux réseaux utilisant le même protocole. [11] Les ponts travaillent au niveau de la couche 2 du modèle OSI (liaison de données).

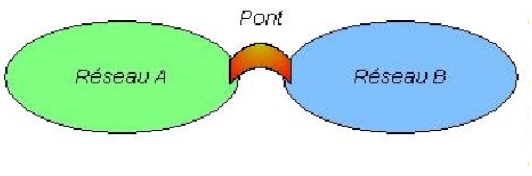

Figure 1.06 : *Liaison de deux réseaux via un pont*

1.7.2.3 Les routeurs

Les routeurs permettent de relier des réseaux de différents protocoles. Ils travaillent au niveau de la couche 3 du modèle OSI (couche réseau). [1]

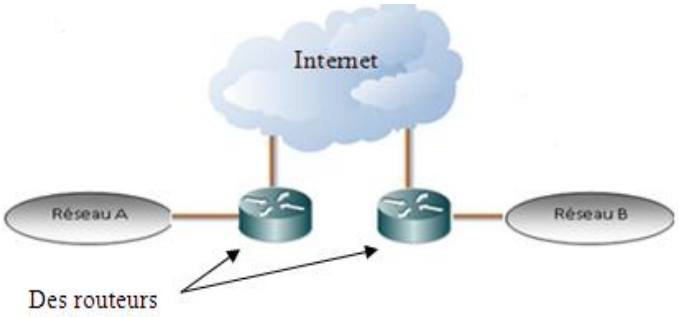

Figure 1.07 : *Le rôle du Routeur dans le réseau*

1.7.2.4 Les passerelles

Les passerelles sont des systèmes matériels et logiciels permettant de faire la liaison entre deux réseaux.

Au sens strict du terme, une passerelle est un outil permettant de faire communiquer entre eux deux réseaux n'utilisant pas le même protocole. La passerelle doit alors dépouiller la trame des informations spécifiques au protocole émetteur et les remplacer par leurs équivalentes dans le protocole récepteur.

1.8 Conclusion

On a vu le fonctionnement du réseau client/serveur, la méthode d'accès dépend donc du choix de la topologie à utiliser. Pour interconnecter les réseaux, on a besoins des quelques matériels. Le réseau à commutation de paquet décrit par le modèle OSI de l'ISO est moins risqué que l'envoi de façon complète alors on va voir maintenant d'autre modèle qui utilise ce principe, c'est le modèle TCP/IP et ses protocoles.

CHAPITRE 2
LES RESEAUX TCP/IP ET SES PROTOCOLES

2.1 Les protocoles

2.1.1 *Définition du protocole*

C'est un dialogue entre des machines connectés. Il n'y a pas de communication possible sans avoir recours à un protocole. Le protocole doit être alors adapté au type de communication que l'on veut mettre en œuvre.

2.1.2 *Description*

Le modèle OSI définit sept couches. TCP/IP (Transmission Control Protocol/Internet Protocol) est basé sur le modèle DoD (Department of Defense), qui ne comporte que quatre couches, mais en cohérence avec le modèle OSI.

<table>
<tr><th>Modèle OSI</th><th>Modèle TCP/IP</th></tr>
<tr><td>Application</td><td rowspan="3">Application</td></tr>
<tr><td>Presentation</td></tr>
<tr><td>Session</td></tr>
<tr><td>Transport</td><td>Transport</td></tr>
<tr><td>Réseau</td><td>Internet</td></tr>
<tr><td>Liaisons de données</td><td rowspan="2">Accès au Réseau</td></tr>
<tr><td>Physique</td></tr>
</table>

Figure 2.01 : *Les modèles en couche*

2.1.3 *Les protocoles de la couche application jusqu'à la couche physique*

Par organisation hiérarchique, les protocoles de haut niveau sont les protocoles applicatifs. Ils sont destinés à permettre le dialogue entre applications serveurs et clientes. [14] Les plus utilisés par

les internautes sont : HTTP (Hyper Text Transfer Protocol), FTP, POP (Post Office Protocol) et SMTP (Simple Mail Transfer Protocol). Il y a aussi TELNET (TELecommunication Network Protocol) qui permet de piloter une machine à distance.

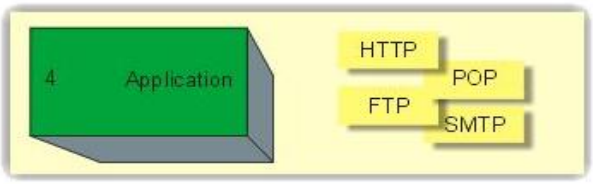

Figure 2.02 : *Les protocoles mise en œuvre dans la couche Application*

Les protocoles pour le transport de données sont d'une part UDP (User Datagram Protocol) dit « mode non connecté », d'autre part TCP dit « mode connecté ». Ces protocoles permettent à ceux de la couche 4 de transporter leurs données de façon fiable.

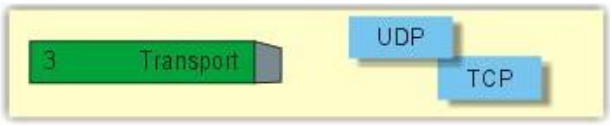

Figure 2.03 : *Les protocoles mise en œuvre dans la couche Transport*

Dans la couche réseau, il y a aussi de protocole de haut niveau, c'est le protocole IP qui permet le routage des informations entre réseau. Ensuite, le protocole ICMP (Internet Control Message Protocol): un protocole de « contrôle », c'est-à-dire des outils de détection d'erreur et de signalisation.

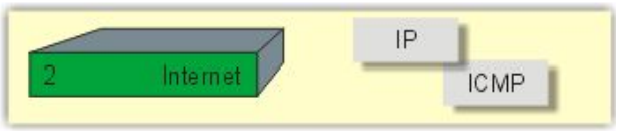

Figure 2.04 : *Les protocoles mise en œuvre dans la couche Internet*

Le protocole de plus bas niveau dans le réseau, c'est le CSMA /CD.il permet la détection de cette collision et de solutionner l'existence de ces interférence en contrôlant les conflits entre les machines qui tentent d'émettre en même temps pour que le réseau fonctionne bien.

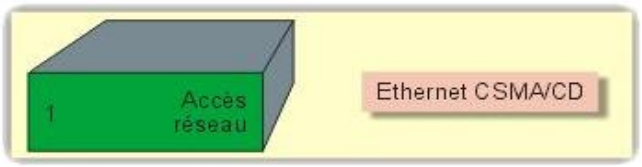

Figure 2.05 : *Les protocoles mise en œuvre dans la couche Accès Réseau*

2.1.4 Les deux modes de transfert

2.1.4.1 Le mode connecté (TCP)

TCP est l'acronyme de « Transport Control Protocol », c'est un protocole de la couche transport. Il est dit de bout en bout (END to END). Lors d'une application, on a besoins de lui pour le transport de donnés. Il y a une accusé réception qui arrive sur l'émetteur lorsque les donnés à transmettre sont bien reçus par le récepteur, si non, on les retransmet. Il ouvre une session et contrôle lui-même le contrôle d'erreur, il est alors appelé « mode connecté ».Il permet alors l'établissement d'un « circuit virtuel » entre les deux points qui échange de l'information. Autres services fournit par TCP : le contrôle de flux et il est en mode full duplex. [15]

TCP fragmente le message à transmettre avant de repasser à la couche internet. Et en réception TCP replace en ordre des segments.

L'établissement d'une connexion TCP s'effectue en trois temps, comme le schéma de la figure 2.05 l'explicite.

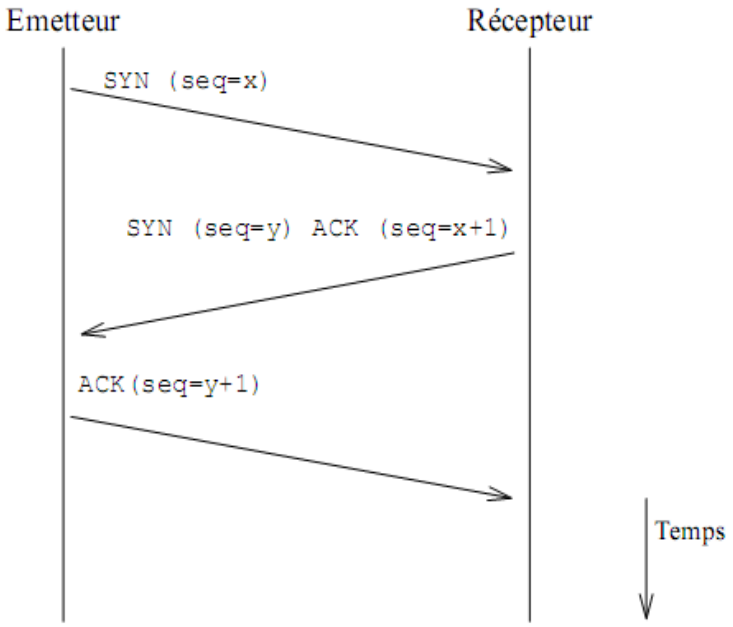

Figure 2.06 : *Etablissement d'une connexion*

On suppose que l'émetteur du premier paquet avec le bit SYN a connaissance du couple (adresse IP du récepteur, numéro de port du service souhaité).

L'émetteur du premier paquet est à l'origine de l'établissement du circuit virtuel, c'est une attitude généralement qualifiée de "cliente". On dit aussi que le client effectue une "ouverture active" (active open). Le récepteur du premier paquet accepte l'établissement de la connexion, ce qui suppose qu'il était prêt à le faire avant que la partie cliente en prenne l'initiative. C'est une attitude de "serveur". On dit aussi que le serveur effectue une "ouverture passive" (passive open).

1. Le client envoie un segment comportant le drapeau SYN, avec sa séquence initiale (ISN = x).

2. Le serveur répond avec sa propre séquence (ISN = y), mais il doit également acquitter le paquet précédent, ce qu'il fait avec ACK (seq =x + 1).

3. Le client doit acquitter le deuxièmes segment avec ACK (seq = y + 1).

2.1.4.2 Le mode non connecté (UDP)

UDP est un protocole de transport qui est très proche d'IP (Internet Protocol). Il permet d'échanger des informations entre des applications. UDP prend le datagramme de l'utilisateur et le

transmet à la couche IP. [14] Cette dernière l'achemine sur la machine destinataire pour le remettre au protocole UDP.

UDP est un protocole plus simple. Son utilisation suppose qu'on n'a pas besoins ni de contrôle de flux ni de conservation de l'ordre de remise de papier.

On utilise UDP lorsque le temps de remise de papier est très dominant, par exemple, on l'utilise sur la voix IP.

2.1.4.3 Les protocoles de résolution d'adresse (ARP)

Le protocole ARP [2], signifiant Address Resolution Protocol, fonctionne en couche Internet du modèle TCP/IP correspondant à la couche 3 du modèle OSI. L'objectif d'ARP est de permettre de résoudre une adresse physique par l'intermédiaire de l'adresse IP correspondante d'un host distant.

Le protocole ARP apporte un mécanisme de « translation » pour résoudre ce besoin. Il permet d'obtenir l'adresse physique (MAC (Medium Access Control), niveau 2) d'une machine connaissant son adresse IP (logique, niveau 3).

2.2 Adressage IP

Lors de la mise en place d'un réseau TCP/IP, on a besoins d'affecter des adresses internet aux nœuds du réseau. L'ensemble doit être cohérent et tous les périphériques doivent avoir une adresse unique. [2] [14]

Dans sa version 4, IP définit une adresse sur quatre octets. Une partie définit l'adresse du réseau (Net-ID ou Subnet-ID suivant le cas), l'autre partie définit l'adresse de l'hôte dans le réseau (Host-ID). La taille relative de chaque partie varie suivant la classe choisie.

2.2.1 Les classes d'adresses

Tous les hôtes dans un même réseau possèdent le même Net-ID et un Host-ID différent. [2] On a 32 bits utilisés par l'adresse IP, on utilisera le premier, les deux premiers ou les trois premiers octets comme Net-ID. à partir de cette différence, on peut créer les classes d'adresses :

- Si le premiers est utilisé comme Net-ID, les trois restants servent pour le Host-ID, on a une adresse de classe A.

- Si les deux premiers octets sont utilisés comme Net-ID et les deux autres comme Host-ID, c'est une adresse de classe B.
- Si les trois premiers octets sont utilisés comme Net-ID et le dernier comme Host-ID, c'est une adresse de classe C.

En plus des trois classes A, B et C, il existe encore deux autres classes, ces utilisations sont très rares. La classe D est réservé à la multidiffusion (multicast), technique utilisé par des protocoles spéciaux pour transmettre simultanément des messages à un groupe donné de nœuds différant. La classe E est réservée à un usage ultérieur.

2.2.2 Etendue de chaque classe

Comment fait-on pour savoir à quelle classe appartient une adresse ? Il y a deux méthodes pour le savoir :

- La triviale, qui consiste à apprendre par cœur le tableau [4]
- La subtile, qui consiste à retenir la règle, qui est logique.

Enumérons la règle :

La classe est définie par les bits les plus lourds (les plus à gauche).

Le bit le moins signifiant pour la classe est toujours un 0.

Les autres sont tous à 1.

La classe A est signalée par un seul bit, donc obligatoirement un 0.

La classe B par deux bits, donc 10.

La classe C par trois bits, donc 110.

La classe D par 4 bits donc 1110.

Les adresses de classe A :

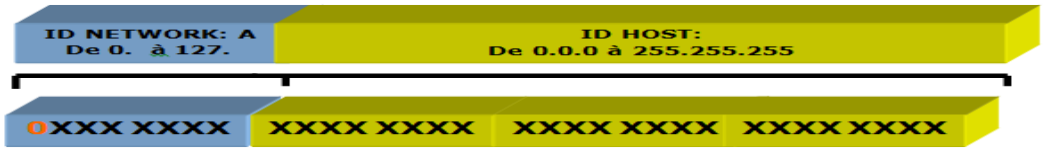

Figure 2.07 : *Schéma simplifié des adresses de classe A*

Les adresses de classe B :

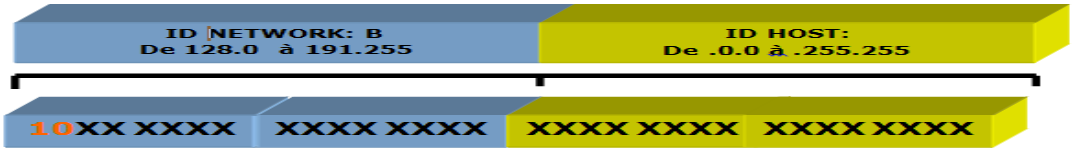

Figure 2.08 : *Schéma simplifié des adresses de classe*

Les adresses de classe C :

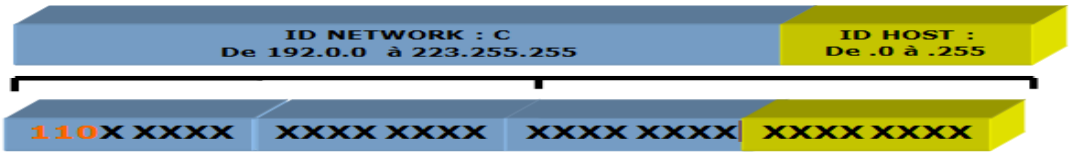

Figure 2.09 : *Schéma simplifié des adresses de classe*

Les adresses de classe D :

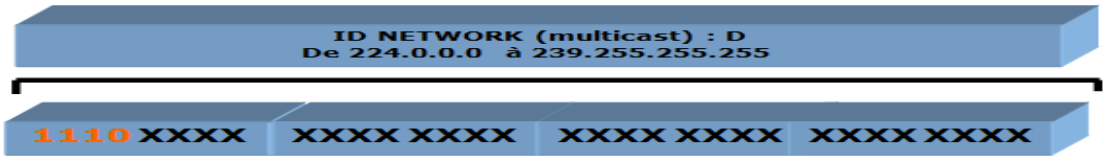

Figure 2.10 : *Schéma simplifié des adresses de classe*

Les adresses de classe E :

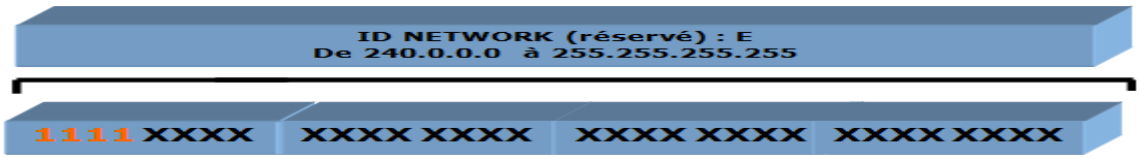

Figure 2.11 : *Schéma simplifié des adresses de classe*

: Adresse du réseau

: Adresse de l'hôte

Il existe même une classe E, dont les bits les plus lourds sont 11110, qui est réservée à un usage ultérieur.

Enumérons dans un tableau l'étendu de chaque classe en tableau :

Classe	Première adresse	Dernier adresse
A	0.0.0.1	127.255.255.254
B	128.0.0.1	191.255.255.254
C	192.0.0.1	223.255.255.254
D	224.0.0.1	239.255.255.254

Tableau 2.01: *Etendu de chaque classe*

2.2.3 Les réseaux privés

Pour la raison de sécurité du réseau privé des entreprises, il est nécessaire de leur réserver dans chaque classe A, B et C des adresses de réseau qui ne sont jamais attribuées sur l'Internet. [14]

Tout paquet de données contenant une adresse appartient à ces réseaux doit être éliminé par le premier routeur établissant une connexion avec l'Internet. On dit que ces adresses ne sont pas routables. On va voir dans un tableau ces réseaux privés :

Classe	Réseau privé	Identification
A	10.0.0.0	Pour les réseaux privés
	127.0.0.0	Pour l'interface de boucle locale
B	172.16.0.0 à 172.31.0.0	Pour les réseaux privés
C	192.168.0.0 à 192.168.255.0	Pour les réseaux privés

Tableau 2.02: *Réseaux privés*

2.2.4 Le masque de sous réseau

Le masque de sous réseau a une importance que peu d'utilisateur connaissent, elle est pourtant fondamentale. C'est un ensemble de 4 octets destiné à isoler :

-Soit l'adresse de réseau (NetID ou SubnetID) en effectuant un ET logique bit à bit entre l'adresse IP et le masque. [1] [2]

-Soit l'adresse de l'hôte (HostID) en effectuant un ET logique bit à bit entre l'adresse IP et le complément du masque.

Les masques de sous réseau par défaut sont, suivant les classes d'adresses :

Classe	Masque par défaut	Nombre d'octet pour l'hôte
A	255.0.0.0	3
B	255.255.0.0	2
C	255.255.255.0	1

Tableau 2.03: *Les masques de sous réseau par défaut*

Pour l'administration de réseau, on peut faciliter sa gestion, de diagnostiquer des pannes, on fractionne les réseaux en sous réseau. Un masque de sous réseau doit être utilisé pour fabriquer des sous réseau à partir d'un réseau d'adresse IP donné.

Pour fabriquer un sous réseau, on enlève une partie de bit permettant d'identifier les machines. On dit qu'on a emprunté des bits.

En empruntant n bits pour fabriquer un sous réseau, on peut disposer de 2^n sous réseau.

Voici alors les nombres de bits à emprunter et les nombres de sous réseau qu'on puisse fabriquer dans chaque classe :

Classe C : emprunt 3 ⟶ 8 sous réseaux

Classe B : emprunt 10 ⟶ 1024 sous réseaux

Classe A : emprunt 16 ⟶ 65536 sous réseaux

Exemple de sous réseau :

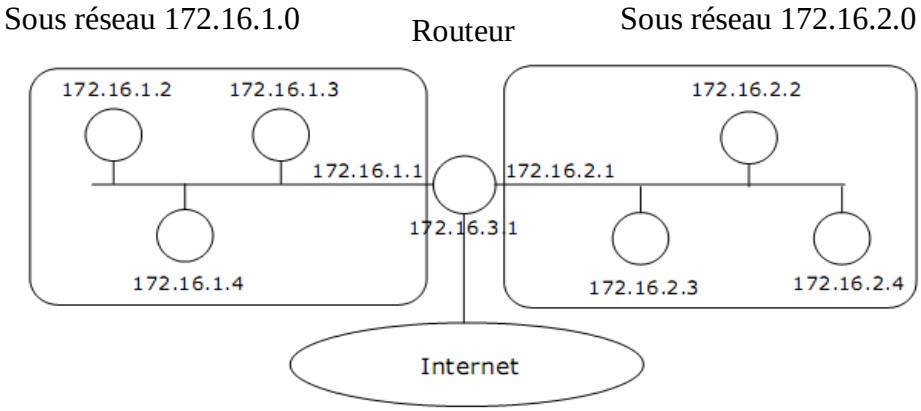

Figure 2.12 : *Schéma détaillé des adresses de sous réseau*

Net ID	Sous réseau	Host ID
172.16	1	1
172.16	1	2
172.16	1	3
172.16	1	4
172.16	2	1
172.16	2	2
172.16	2	3
172.16	2	4

Tableau 2.04: *Les adresses de sous réseau*

Tableau détaillé des adresses de sous réseau

2.3 Les routages de datagramme IP

2.3.1 *Le routage*

Le routage est la tâche consistant à trouver un chemin d'un émetteur à une destination souhaitée. Il se réduit essentiellement à trouver des routeurs entre des réseaux. Aussi longtemps qu'un message reste sur un réseau ou sous-réseau unique, tout problème de routage est résolu par une technologie qui est spécifique au réseau. Par exemple, Ethernet définit un moyen par lequel tout émetteur peut parler à toute destination spécifiée à l'intérieur de ce propre réseau. Le routage IP entre en jeu essentiellement quand les messages doivent aller d'un émetteur sur un tel réseau vers une destination située sur un autre réseau. Dans ce cas, le message doit traverser des routeurs connectant les réseaux. Si les réseaux ne sont pas adjacents, le message peut traverser plusieurs réseaux intermédiaires, et les routeurs les connectant. Une fois que le message arrive sur un routeur situé sur le même réseau que la destination, la technologie propre de ce réseau est utilisée pour atteindre la destination.

2.3.2 *Les routeurs*

Les routeurs sont les dispositifs permettant de "choisir" le chemin que les datagrammes vont emprunter pour arriver à destination.

Il s'agit de machines ayant plusieurs cartes réseau dont chacune est reliée à un réseau différent. Ainsi, dans la configuration la plus simple, le routeur n'a qu'à "regarder" sur quel réseau se trouve un ordinateur pour lui faire parvenir les datagrammes en provenance de l'expéditeur.

Toutefois, sur Internet le schéma est beaucoup plus compliqué pour les raisons suivantes:

- Le nombre de réseau auquel un routeur est connecté est généralement important.

- Les réseaux auquel le routeur est relié peuvent être reliés à d'autres réseaux que le routeur ne connaît pas directement.

Ainsi, les routeurs fonctionnent grâce à des tables de routage et des protocoles de routage, selon le modèle suivant:

- Le routeur reçoit des datagrammes provenant d'une machine connectée à un des réseaux auquel il est rattaché.

- Les datagrammes sont transmis à la couche Internet.

- Le routeur regarde l'en-tête du datagramme.

- Si l'adresse IP fait partie du réseau duquel le datagramme provient, le routeur n'a aucune action à accomplir car la machine visée aura reçu ce même datagramme.

- Si l'adresse IP fait partie d'un réseau différent, le routeur consulte sa table de routage, une table qui définit le chemin à emprunter pour une adresse donnée.

- Le routeur envoie le datagramme grâce à la carte réseau reliée au réseau sur lequel le routeur décide d'envoyer le paquet.

Ainsi, il y a deux scénarios, soit l'émetteur et le destinataire appartiennent au même réseau auquel cas on parle de remise directe, soit il y a au moins un routeur entre l'expéditeur et le destinataire, auquel cas on parle de remise indirecte.

Dans le cas de la remise indirecte, le rôle du routeur et surtout de la table de routage est très important. Ainsi le fonctionnement d'un routeur est déterminé par la façon selon laquelle cette table de routage est créée.

- Si la table routage est entrée manuellement par l'administrateur, on parle de routage statique (viable pour de petits réseaux).

- Si le routeur construit lui-même la table de routage en fonctions des informations qu'il reçoit (par l'intermédiaire de protocoles de routage), on parle de routage dynamique.

2.3.3 La table de routage

La table de routage est une table de correspondance entre l'adresse de la machine visée et le nœud suivant auquel le routeur doit délivrer le message. En réalité il suffit que le message soit délivré sur le réseau qui contient la machine, il n'est donc pas nécessaire de stocker l'adresse IP complète de la machine: seul l'identificateur du réseau de l'adresse IP (c'est-à-dire le Net id a besoin d'être stocké.

La table de routage est donc un tableau contenant des paires d'adresses :

Adresse de destination Adresse du prochain routeur directement accessible Interface

Ainsi grâce à cette table, le routeur, connaissant l'adresse du destinataire encapsulé dans le message, va être capable de savoir sur quelle interface envoyer le message (cela revient à savoir quelle carte réseau utiliser), et à quel routeur, directement accessible sur le réseau auquel cette carte est connectée, remettre le datagramme.

Ce mécanisme consistant à ne connaître que l'adresse du prochain maillon menant à la destination est appelé routage par sauts successifs (en anglais next-hop routing).

Cependant, il se peut que le destinataire appartienne à un réseau non référencé dans la table de routage. Dans ce cas, le routeur utilise un routeur par défaut (appelé aussi passerelle par défaut).

Exemple de table de routage :

Destination	Passerelle	Genmask	Indic	Metric	Ref	Use	Iface
255.255.255.255	*	255.255.255.255	UH	0	0		eth0
193.253.160.3	*	255.255.255.255	UH	0	0		ppp0
192.168.0.0	*	255.255.255.0	U	0	0		eth0
10.0.0.0	*	255.0.0.0	U	0	0		eth1
127.0.0.0	*	255.0.0.0	U	0	0		l0
default	193.253.160.3	0.0.0.0	UG	0	0		PPP0

Tableau 2.05: *Table de routage*

Le message est ainsi remis de routeur en routeur par sauts successifs, jusqu'à ce que le destinataire appartienne à un réseau directement connecté à un routeur. Celui-ci remet alors directement le message à la machine visée.

Dans le cas du routage statique, c'est l'administrateur qui met à jour la table de routage.

Dans le cas du routage dynamique, par contre, un protocole appelé protocole de routage permet la mise à jour automatique de la table afin qu'elle contienne à tout moment la route optimale.

2.3.4 Les protocoles de routage

Pour interconnecter un ensemble de réseaux, on a besoin des différents routeurs qui ne font pas le même travail selon le type de réseau sur lequel ils se trouvent.

En effet, il y a différents niveaux de routeurs, ceux-ci fonctionnent donc avec des protocoles différentes :

- Les routeurs noyaux : ils sont les routeurs principaux car ce sont eux qui relient les différents réseaux.
- Les routeurs externes : ils permettent une liaison des réseaux autonomes entre eux. Ils fonctionnent avec un protocole appelé EGP (Exterior Gateway Protocol) qui évolue petit à petit en gardant la même appellation.
- Les routeurs internes permettent le routage des informations dans un réseau local. Il s'échange des informations grâce à des protocoles appelés IGP (Interior Gateway Protocol), tel que RIP (Routing Information Protocol) et OSPF (Open Shortest Path First).

RIP signifie protocole d'information de routage. Il s'agit d'un protocole de type Vecteur Distance, c'est-à-dire que chaque routeur communique aux autres routeurs la distance qui les sépare (le nombre de saut qui les sépare). Ainsi, lorsqu'un routeur reçoit un de ces messages il incrémente cette distance de 1 et communique le message aux routeurs directement accessibles. Les routeurs peuvent donc conserver de cette façon la route optimale d'un message en stockant l'adresse du routeur suivant dans la table de routage de telle façon que le nombre de saut pour atteindre un réseau soit minimal. Toutefois ce protocole ne prend en compte que la distance entre deux machines en termes de saut, mais il ne considère pas l'état de la liaison afin de choisir la meilleure bande passante possible.

OSPF est plus performant que RIP et commence donc à le remplacer petit à petit. Il s'agit d'un protocole de type protocole route-link (que l'on pourrait traduire par Protocole d'état des liens), cela signifie que, contrairement à RIP, ce protocole n'envoie pas aux routeurs adjacents le nombre de sauts qui les sépare, mais l'état de la liaison qui les sépare. De cette façon, chaque routeur est capable de dresser une carte de l'état du réseau et peut par conséquent choisir à tout moment la route la plus appropriée pour un message donné.

De plus, ce routeur évite aux routeurs intermédiaires d'avoir à incrémenter le nombre de sauts, ce qui se traduit par une information beaucoup moins abondante, ce qui permet d'avoir une meilleure bande passante utile qu'avec RIP.

2.4 Les applications TCP/IP

2.4.1 Adresses et ports

Un port TCP/IP est un numéro de service. Quel que soit l'OS sur une machine, gérer des numéros n'est pas difficile à faire. Un port doit être vu comme un lien de rendez-vous. Le programme serveur, va demander au système d'exploitation de lui donner toutes les informations qui arrivent sur un ou plusieurs ports donnés. [14] [7] Le programme client qui veut dialoguer avec le serveur, doit donc émettre ses données vers un port spécifié sur une machine donnée.

Un client n'a donc aucune idée de l'appellation de l'applicatif (programme, job, processus,..).

Le problème pour un client est donc de connaitre le numéro de port qui lui permettra de joindre le bon serveur.

Sous TCP/IP, les ports de 0 à 1023 sont réservés pour les applications standards : les ports « bien connus » (Well Know Ports).

D'autres numéros de port sont disponibles pour les applications développées par les utilisateurs (1024 à 65535).

2.4.2 Le serveur et le client

Les serveurs ont une fonction particulière : ils doivent envoyer des informations pertinentes aux clients qui en réclament. Comme un serveur ne convient pas d'un rendez-vous avec le client, il doit rester attentif en permanence pour ne pas risquer de rater une question. Pour ce faire, on y installe des « daemons », petits programme qui tournent en tache de fond et qui écoutent continuellement sur un numéro de port donné. Il y a de convention pour attribuer ces ports sur des services connus, pat exemple le port 80 pour HTTP, le port 110 pour POP3, le port 21 pour FTP. Il faut qu'il ait des conventions de ce genre pour que les clients puissent atteindre ces services. Lorsque l'on écrit http://62.161.120.45, on ne spécifie pas de port ; sous-entendu, il s'agit du port 80 parce que l'on invoque un service http. Il serait possible d'écrire : http://62.161.120.45:80 ici, on spécifie le port. Certaines protections triviales consistent justement à forcer un service à ne pas

employer à ne pas employer le port standard. Un administrateur pourrait décider de mettre son serveur http à l'écoute du port 88. [11] Dans ce cas, si l'utilisateur n'est pas au courant de cette particulier, il ne pourra pas accéder à ce serveur (sauf s'il dispose d'un scanneur de port et qu'il découvre la supercherie)

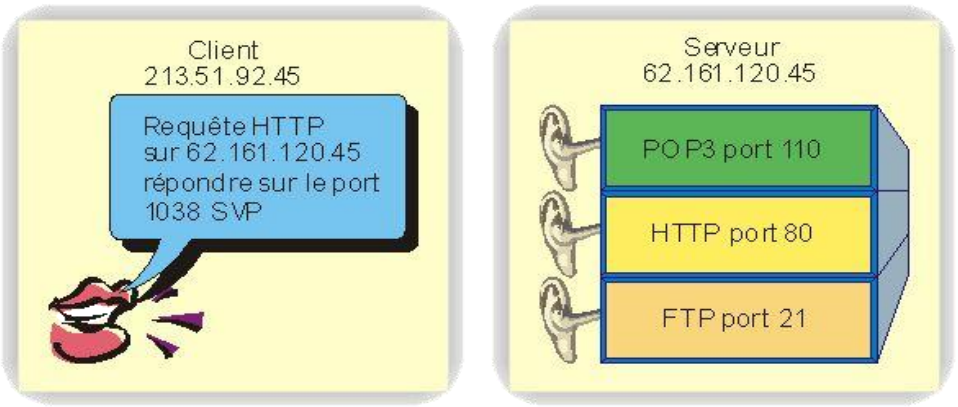

Figure 2.13 : *Demande de communication du client à son serveur*

En revanche, le client qui émet la requête ne dispose pas de port d'écoute attiré. Ce n'est pas un serveur, c'est un client ; il n'a donc rien à écouter d'autre que les réponses à ses questions ? Il faut donc, lorsqu'il envoie sa requête, qu'il spécifie sur quel port il va écouter la réponse, de manière à ce que le serveur puisse construire un socket efficace pour latitude réponse.

Vous êtes-vous demandé par quel miracle, si vous ouvrez deux fois votre navigateur pour afficher deux pages différents sur le même serveur, les informations ne se mélangent pas.

C'est parce que les deux sessions du navigateur indiquent des ports de réponse différents. C'est parce que les deux sessions du navigateur indiquent des ports de réponse en fonction de ceux qui sont disponibles sur la machine.

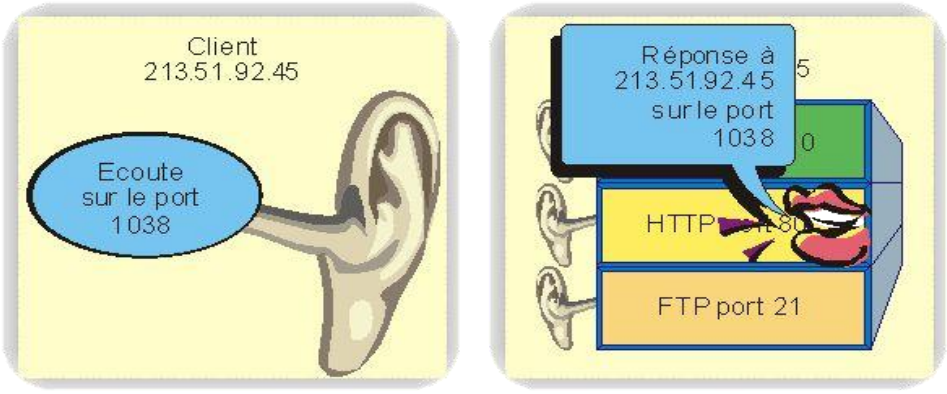

Figure 2.14 : *Réponse du serveur*

2.5 Conclusion

En bref, les machines connectés se dialoguent entre elles, les protocoles sont donc indispensables du réseau, on a vu divers modèle, il suffit alors d'attribuer des adresses pour les machines et de définir les ports pour les connexions. Les chemins des datagrammes IP d'un émetteur vers un récepteur sont réalisés par le routage.

CHAPITRE 3
LA REALISATION D'UN RESEAU SOUS LINUX

3.1 Présentation du système d'exploitation linux

3.1.1 Historique

En 1991, **Linus Torvalds**, un étudiant de l'Université de Helsinski en Finlande, entreprend de créer son propre système d'exploitation sur son temps libre. Ce système a pris le nom de Linux, en référence au nom de son créateur (Linux est la contraction de Linus et Unix). Il est complémentaire avec le projet GNU (GNU is Not Unix (de Richard Stallman qui créait les programmes de base (programme de copie de fichier, suppression de fichier, éditeur de texte), Linus s'était lancé dans la création du "cœur" d'un système d'exploitation (le noyau).

Le projet GNU (programmes libres) et Linux (noyau d'OS) ont fusionné pour créer **GNU/Linux**. Théoriquement, on doit donc parler de GNU/Linux. C'est toutefois un peu difficile à écrire et prononcer, et par abus de langage on dit souvent juste "Linux".

Aujourd'hui, le succès de Linux s'explique par la qualité technique du noyau, par la présence de nombreuses distributions Linux qui facilitent l'installation du système et des programmes, mais s'explique surtout par son appartenance au monde du Libre qui lui apporte une grande rapidité et qualité de développement. Le nombre d'utilisateurs de Linux est aujourd'hui estimé à plusieurs millions. [16]

3.1.2 Les différentes distributions existantes

Il existe un grand nombre de distributions Linux différentes, voici au moins les principales :

- **Slackware** : une des plus anciennes distributions de Linux. Elle existe toujours aujourd'hui.
- **Mandriva** : éditée par une entreprise française, elle se veut simple d'utilisation.
- **RedHat** : éditée par une entreprise américaine "RedHat", cette distribution est célèbre et très répandue, notamment sur les serveurs.
- **SuSE** : éditée par l'entreprise Novell.
- **Debian** : la seule distribution qui soit gérée par des développeurs indépendants au lieu d'une entreprise. C'est une des distributions les plus populaires.

Les différences entre eux sont les fonds d'écran et les logiciels préinstallés.

On a fait le choix d'utiliser un système d'exploitation des distributions Debian.

Un autre gros avantage de Debian, c'est le gestionnaire de paquets apt-get. C'est un programme qui gère tous les logiciels installés. Vous pouvez les désinstaller en un rien de temps. D'autre part, tous les logiciels sont centralisés en un même endroit (paquets), ce qui fait que vous n'avez pas à parcourir tout le Web pour retrouver un programme.

Voici alors les distributions Debian :

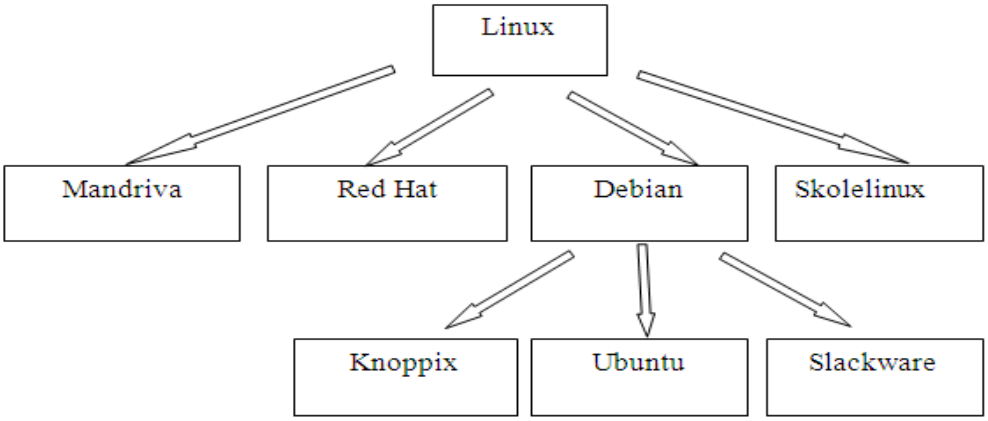

Figure 3.01 : *Les distributions Debian*

3.1.3 Les causes de l'utilisation de linux comme serveur

Linux est un système qui connait actuellement le plus grand développement sur l'internet.

Principalement pour la raison suivante [17] :

- Le logiciel Samba qui lui permet d'être serveur de fichier et d'impression avec des clients windows ;
- La stabilité et la sécurité que lui donne le développement de son architecture et de ses modules au sein de la communauté Open Source (communauté de logiciel libre).
- Le large choix d'applications dans de très nombreux domaines. Par exemple, la dernière distribution Debian donne accès à plus de 2000 logiciels différents.
- Un serveur sous linux ne doit être redémarré que lors d'une modification matérielle comme l'ajout d'un disque ou d'une carte.

- ❖ Logiciel libre. Linux est gratuit et librement repliable. Cela signifie que l'on peut télécharger une version de Linux ou l'emprunter et l'installer sur n'importe quel nombre d'ordinateur.
- ❖ Accès aux sources des logiciels. Tous les utilisateurs peuvent modifier le fonctionnement des programmes ou engager un programmeur pour le faire.
- ❖ Linux est plus efficace et consomme moins de ressources CPU et mémoire que Windows. On peut par exemple faire un serveur d'impression avec un vieux 486.

3.2 Les gestions d'accès au réseau sous linux et à tous ces ressources

3.2.1 Buts du projet

- Mettre en place un contrôleur de domaine Samba-LDAP, on choisit samba pour partager des ressources avec des clientes windows et LDAP est pour la gestion d'authentification.
- Configurer toutes les machines clientes pour joindre le domaine
- Tous les utilisateurs doivent être membres du domaine
- Mettre en place un système d'authentification lors d'une demande de connexion avec NSS *(Name Service Switch)* et PAM (Plugable Authentification Module).

3.2.2 Le serveur LDAP : OpenLDAP

3.2.2.1 Quelques définitions utiles

Bases de données :

Une base de données (son abréviation est BD, en anglais DB, *data base*) est une entité dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance possible. Ces données doivent pouvoir être utilisées par des programmes, par des utilisateurs différents. [12] Ainsi, la notion de base de données est généralement couplée à celle de réseau, afin de pouvoir mettre en commun ces informations, d'où le nom de **base**. On parle généralement de système d'information pour désigner toute la structure regroupant les moyens mis en place pour pouvoir partager des données.

Annuaire :

Un annuaire est une bibliothèque (imprimée ou électronique) mise à jour régulièrement qui regroupe des informations (nom, adresse, coordonnées, etc.) sur les membres d'une association, d'une entreprise ou d'un organisme professionnel, ou sur les abonnés à un service.[5]

Bibliothèque : [12]

Une bibliothèque (du grec biblio, « livre » ; thêkê, « place ») est une collection organisée de livres.

Annuaire électronique :

Un annuaire électronique [12] est une base de données spécialisée, dont la fonction première est de retourner un ou plusieurs attributs d'un objet grâce à des fonctions de recherche multicritères.

Contrairement à un SGBD (Système de Gestion de Base de Données), un annuaire est très performant en lecture mais l'est beaucoup moins en écriture. Sa fonction peut être de servir d'entrepôt pour centraliser des informations et les rendre disponibles, via le réseau à des applications, des systèmes d'exploitation ou des utilisateurs.

OpenLDAP :

L'implémentation LDAP (*Lightweight Directory Access Protocol)* la plus répandue actuellement dans le monde libre est OpenLDAP. [5] Elle est développée par The OpenLDAP Project. OpenLDAP est disponible dans le paquet Debian et portant le nom « slapd ». [8]

3.2.2.2 Les principes d'un annuaire LDAP

- Pour définir ce qu'est le service LDAP, [15] on peut retenir les caractéristiques suivantes:
- Un service de publication [3] d'annuaire
- Un protocole d'accès aux annuaires de type X.500
- Un dépôt de données basées sur des attributs ou un «genre» de BD
- Un logiciel optimisé pour les recherches avancées et les lectures
- Une implémentation client/serveur
- Un mécanisme extensible de schémas de description de classes d'objets.

Les entrées d'un annuaire LDAP sont distribuées suivant une arborescence hiérarchisée.

La figure ci-dessous montre l'arborescence LDAP élémentaire:

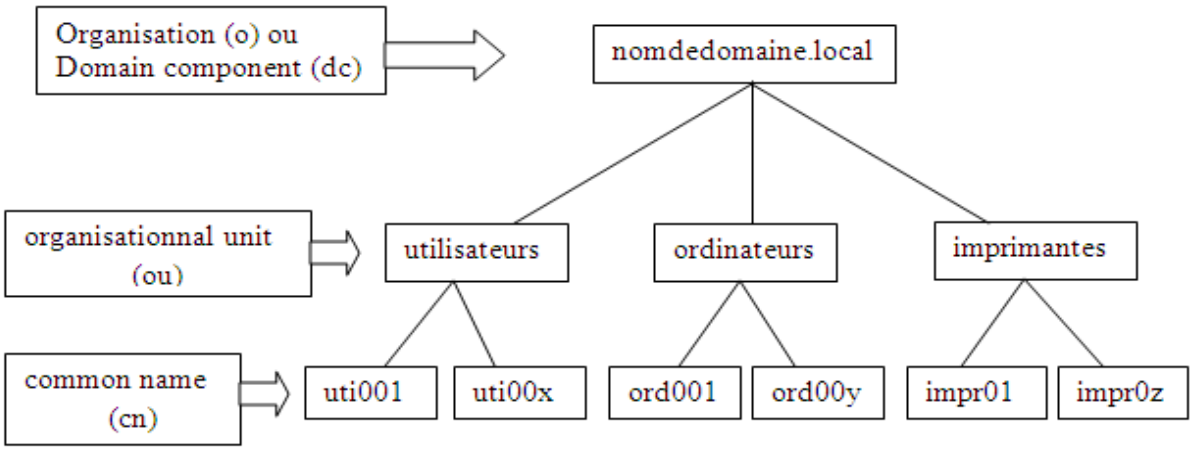

Figure 3.02 : *Arborescence LDAP élémentaire*

Détaillons l'arborescence LDAP élémentaire pour une vue complet :
L'adresse d'une entrée de l'annuaire LDAP est appelée : distinguished name (dn).

Généralement, les champs « dc (domain controller) » correspondent aux parties du nom de domaine DNS auquel le serveur LDAP appartient. Dans notre exemple, on peut faire correspondre le domaine DNS (Domain Name Server) « nomdedomaine.local » avec la déclaration LDAP : dc=nomdedomaine,dc=local

En reprenant l'exemple d'arborescence ci-dessus, les adresses des différentes entrées sont notées comme suit :

dn: dc=nomdedomaine,dc=local
dn: ou=utilisateurs,dc=nomdedomaine,dc=local
dn: ou=ordinateurs,dc=nomdedomaine,dc=local
dn: ou=imprimantes,dc=nomdedomaine,dc=local
dn: cn=uti001, ou=utilisateur,dc=nomdedomaine,dc=local
dn: cn=uti00x,ou=utilisateur,dc=nomdedomaine,dc=local
dn:cn=ordi001, ou=ordinateur,dc=nomdedomaine,dc=local
dn:cn=ordi00y,ou=ordinateur,dc=nomdedomaine,dc=local
dn: cn=impr01, ou=imprimante,dc=nomdedomaine,dc=local
dn: cn=impr0z,ou=imprimante,dc=nomdedomaine,dc=local

- Parlons un peu de classe d'objet et schéma :

L'adresse de chaque entrée appartient à une classe d'objet spécifiée dans un schéma.

En reprenant les mêmes exemples d'entrées, on peut associer les classes d'objets correspondantes :

entry	**Object class**
o : nomdedomaine.local dc : nomdedomaine dc : local	organisation dc object dc object
ou : utilisateurs	Organisational unit
cn : etu sn : etu	person

Tableau 3.01: *Classe d'objet*

Un schéma peut être vu comme un ensemble de règles qui décrivent la nature des données stockées. C'est un outil qui aide à maintenir la cohérence, la qualité et qui évite la duplication des données dans l'annuaire.

Les attributs des classes d'objets déterminent les règles qui doivent être appliquées à une entrée.

Un schéma contient les éléments suivants [3] :

- Les attributs requis
- Les attributs autorisés
- Les règles de comparaison des attributs
- Les valeurs limites qu'un attribut peut recevoir
- Les restrictions sur les informations qui peuvent être enregistrées
- Gestionnaire de BD [12]:

Par définition, un annuaire LDAP est une base de données optimisée en lecture. Du point de vue implémentation, les entrées sont stockées sous forme «binaire» et indexées à l'aide d'un gestionnaire de base de données. [19] Le gestionnaire d'arrière-plan proposé par défaut est hdb. Il s'agit d'une variante du gestionnaire Berkeley DB transactional backend qui offre un fonctionnement hiérarchisé.

3.2.2.3 Free radius

- ❖ Définition :

C'est le mode d'authentification à distance, il se charge aussi de l'autorisation et des informations de session.

- ❖ Exemples d'utilisation :

On l'utilise dans un réseau VPN, dans l'application d'un annuaire LDAP,…

- ❖ Authentification :
- – Par le Système

Le nom et le mot de passe de l'utilisateur sont ceux du système (UNIX), Free radius utilise /etc/passwd et/ou NIS.

C'est la configuration par défaut de Free Radius.

- – En Interne

Le nom et le mot de passe de l'utilisateur sont stockés dans les fichiers de configuration de radius

- – Par Base de données (MySQL, Oracle, Postgre SQL, LDAP) :

Les informations de l'utilisateur sont stockés dans une base de donnée, attention la communication Radius<->SQL n'est pas encryptée.

- – PAM (Pluggable Authentication Service) :

C'est selon la configuration de PAM que va se faire l'authentification.

- ❖ Installation et configuration de Free Radius

Exécuter la commande suivante pour installer Free Radius ainsi que d'autres logiciels nécessaires à son fonctionnement:

```
# apt-get install openssl libssl libssl-dev libldap2-dev g++ build-essential debian-builder
```

libstdc++6.4-1-dev libmysqlclient15-dev

Une fois l'installation finalisée, nous pouvons tester le serveur Free Radius sans modification nécessaire. Lancer la commande suivante: radiusd -X

3.2.2.4 Installation

Commençons par l'installation du serveur LDAP et sa configuration. Le serveur LDAP permettra de fournir à Samba les informations concernant les comptes utilisateurs du domaine.

Nous allons avoir besoin de plusieurs composants :

- les librairies LDAP compilées,

- les sources de ces librairies (pour des compilations futures),

- les utilitaires d'interrogation d'annuaire,

- le serveur LDAP (slapd)

L'installation de tout ceci se fait de manière très simple sous Debian, via apt :

```
# apt-get install slapd ldap-utils ldapvi
```

Lors de cette installation un certain nombre de questions seront posées :

Faut-il omettre la configuration d'OpenLDAP ? **Non**

Nom de domaine : **trav.local**

Nom de votre organisation : **trav**

Mot de passe administrateur : ********

Faut-il autoriser le protocole LDAPv2 : **Non**

Le serveur est désormais installé, nous allons voir comment configurer LDAP pour qu'il nous fournisse un accès à des comptes Samba.

L'installation du schéma samba est essentielle car il contient les attributs nécessaires au LDAP pour le bon dialogue avec samba en PDC.

apt-get install samba-doc

Puis décompresser les répertoires qui contiennent le schéma.

gunzip -c /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz > /etc/ldap/schema/samba.schema

Une fois cette étape finie dans l'ancienne version il suffisait d'ajouter le schéma dans slapd.conf. Mais avec la nouvelle version de samba quand on fait un

ls -l /etc/ldap/slapd.d/cn=config/cn=schema

on obtient :

-rw------- 1 openldap openldap 15474 24 sept. 13:39 cn={0}core.ldif
-rw------- 1 openldap openldap 11308 24 sept. 13:39 cn={1}cosine.ldif
-rw------- 1 openldap openldap 6438 24 sept. 13:39 cn={2}nis.ldif
-rw------- 1 openldap openldap 2802 24 sept. 13:39 cn={3}inetorgperson.ldif

On constate que notre schéma n'est pas présent. La solution (qui n'est peut être pas la bonne, car j'avoue avoir cherché longtemps sans en trouvé de meilleur et la doc d'openldap est très très chiante à comprendre) est de créer un fichier /etc/ldap/slapd.conf manuellement et d'y inclure les schémas nécessaires. Attention il faut remettre les anciens schémas sous peine d'avoir un message d'erreur.

vim /etc/ldap/slapd.conf

On ajoute les schemas dans ce fichier :

include /etc/ldap/schema/core.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/inetorgperson.schema
include /etc/ldap/schema/samba.schema

Et on sauvegarde en sortant. Une fois ce fichier créé on lance la commande suivante qui convertira au nouveau format :

slaptest -f /etc/ldap/slapd.conf -F /etc/ldap/slapd.d/

La commande doit vous retourner config testing succeeded.

Si on refait un petit ls -l /etc/ldap/slapd.d/cn=config/cn=schema on constate que le schéma est bien ajouté, mais que le propriétaire est root et non openldap comme ci-dessous.

-rw------- 1 openldap openldap 15474 24 sept. 13:39 cn={0}core.ldif
-rw------- 1 openldap openldap 11308 24 sept. 13:39 cn={1}cosine.ldif
-rw------- 1 openldap openldap 6438 24 sept. 13:39 cn={2}nis.ldif
-rw------- 1 openldap openldap 2802 24 sept. 13:39 cn={3}inetorgperson.ldif
-rw------- 1 root   root   12492 24 sept. 14:13 cn={4}samba.ldif

On va donc lancer les commandes suivantes pour lui donner les droits
chown openldap:openldap /etc/ldap/schema/ -R
chown openldap:openldap /etc/ldap/slapd.d/ -R

Un fois ces commandes exécuté on peut enfin faire un redémarrage du serveur ldap.

/etc/init.d/slapd restart

Il sera prêt à être interrogé par Samba, une fois que nous l'aurons peuplé.

3.2.3 identification et authentification

L'identification permet de connaître l'identité d'une entité alors que l'authentification permet de vérifier cette identité.

Sous Unix/Linux, NSS se charge de l'identification et PAM de l'authentification.

- NSS [4]:

NSS (commutateur de services de nommages) fournit un service de nommage qui autorise le serveur LDAP d'être utilisé comme service de nom. Ceci signifie qu'il fournit les données informatives concernant le compte utilisateur, identificateurs, informations machines, alias, groupes réseau et d'une manière générale, tout ce qui est dans les fichiers du répertoire /etc ou NIS. Autrement dit, NSS permet de fournir à Unix non des services d'authentification, mais des services de correspondances entre noms, de toutes sortes (noms de machines et noms d'utilisateurs

intelligibles par l'homme, par exemple), et les identifiants de ces mêmes objets pour la machine (adresses IP et uid/gid par exemple). [18]

NSS autorise le remplacement des traditionnels fichiers de configuration (/etc/passwd, /etc/group, /etc/hosts) par une ou plusieurs BD centralisées.

NSS est habituellement configuré à l'aide du fichier /etc/nsswitch.conf. Celui-ci liste les « BD » (par exemple group, passwd, …) et d'autres façons d'obtenir l'information (par ex. : files pour les fichiers locaux, nis pour le Network Information Service, nisplus pour NIS+, ldap pour le protocole LDAP, ou encore mysql pour une base de données MySQL).

- PAM [4] :

PAM (modules d'authentification enfichables) permet l'authentification des utilisateurs dans les programmes. Autrement dit, il permet de personnaliser les procédures et sources d'authentification, mais aussi d'offrir des services supplémentaires aux programmes qui savent les utiliser. [18] C'est un premier pas pour la mise en place d'un SSO (Single Sign On ou authentification unique) sur un système Unix et/ou Linux, ses applications, et même au-delà. S'il est utilisé avec NSS, on peut remplacer la structure de configuration par fichiers /etc ou NIS avec LDAP.

- Les modules PAM:

Les modules PAM sont des bibliothèques dynamiques (par ex. pam_unix.so) fournissant les six primitives d'authentification définies dans la norme, regroupées dans quatre mécanismes :

Le mécanisme account fournit une seule primitive : il vérifie si le compte demandé est disponible (si le compte n'est pas arrivé à expiration, si l'utilisateur est autorisé à se connecter à cette heure de la journée, etc.).

Le mécanisme auth fournit deux primitives ; il assure l'authentification réelle, éventuellement en demandant et en vérifiant un mot de passe, et il définit des « certificats d'identité » tels que l'appartenance à un groupe ou des « tickets » kerberos.

Le mécanisme password fournit une seule primitive : il permet de mettre à jour le jeton d'authentification (en général un mot de passe), soit parce qu'il a expiré, soit parce que l'utilisateur souhaite le modifier.

Le mécanisme session fournit deux primitives : mise en place et fermeture de la session. Il est activé une fois qu'un utilisateur a été autorisé afin de lui permettre d'utiliser son compte. Il lui fournit certaines ressources et certains services, par exemple en montant son répertoire personnel, en rendant sa boîte aux lettres disponible, en lançant un agent ssh, etc.

3.2.4 Configuration de l'authentification LDAP

Nous allons maintenant installer et configurer la librairie qui permet d'utiliser l'annuaire (libnss-ldap) et la librairie qui permet de s'authentifier sous unix (libpam-ldap).

apt-get intall libnss-ldap libpam-ldap

Lors de l'installation un certain nombre de question seront posées :

Identifiant uniforme de ressource (« URI ») du serveur LDAP : **ldap : //127.0.0.1**

Nom distinctif (DN) de la base de recherche : **dc=trav,dc=local**

Version de LDAP utilisé : **3**

La base LDAP demande-t-elle une identification : **non**

Privilèges LDAP spécifiques pour le super utilisateur : **oui**

Rendre le fichier de configuration lisible et modifiable uniquement par son propriétaire : **non**

ompte LDAP pour le superutilisateur (« root ») : **cn=admin,dc=trav,dc=local**

Mot de passe du compte du super utilisateur LDAP : ********

Donner les privilèges de super utilisateur local au compte administrateur LDAP ? oui

La base LDAP demande-t-elle une identification : **non**

Compte LDAP pour le super utilisateur (« root ») : **cn=admin,dc=trav,dc=local**

Mot de passe du compte du super utilisateur LDAP : ********

Algorithme de chiffrement à utiliser localement pour les mots de passe : **chiffré**

Maintenant que les librairies sont configurées, on doit activer la recherche LDAP en modifiant le fichier de configuration **/etc/nsswitch.conf**. Pour cela il faut simplement ajouter ldap à passwd,group et shadow :

passwd: compatldap

group: compatldap

shadow: compatldap

3.2.5 Installation de samba

- Définition de samba :

Samba est un serveur de fichiers pour Linux compatible avec les réseaux Microsoft Windows. C'est-à-dire qu'il permet de partager les fichiers et les imprimantes d'un serveur linux avec les ordinateurs d'un réseau Microsoft Windows.

- Principe de fonctionnement :

Le protocole de communication permettant cette communication entre Windows et Linux s'appelle SMB (*Server Message Block*), on a ajouté les deux voyelles « a » d'où le nom Samba. Samba ne permet d'accéder à la station Windows qu'à travers le protocole TCP/IP.

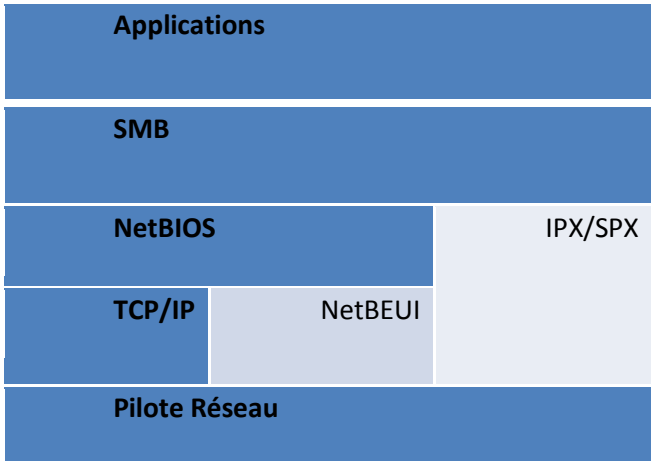

Figure 3.03 : *Les couches utilisées par samba*

❖ Installation des packages

Il suffit d'installer le package samba, qui va installer par dépendance le package samba-common qui sert à la fois au serveur et au client : [12]

apt-get install samba

- Créons les partages:

#mkdir -p /home/netlogon

Ce répertoire va contenir les scripts de connexion des utilisateurs. Ces éventuels scripts seront rédigés en .bat

mkdir -p /home/profile

Ce répertoire contiendra les profils Windows de chaque utilisateur (c:\Documents and settings\utilisateurs)

mkdir -p /home/partage

Partage publique accessible à tous les utilisateurs

mkdir -p /home/nom_utilisateur

Répertoire personnel de l'utilisateur sur le serveur

On ajoute les droits d'écriture sur les répertoires:

#chmod a+w /home/netlogon
#chmod a+w /home/profile
#chmod a+w /home/partage

❖ *Configuration du fichier smb.conf*

Avant toute chose faire une copie de sauvegarde du fichier d'origine

```
# cp /etc/samba/smb.conf / etc/samba/smb.conf.save
```

Une fois la copie effectuée, éditez le fichier smb.conf avec votre éditeur préférer et changer les paramètres à l'intérieur. [17] Voici ci-dessous mon fichier smb.conf pour plus de simplicité.

```
[global]
workgroup = TRAV
server string = Controleur de domaine
netbios name = developpement
domain master = yes
local master = yes
domain logons = yes
client lanmanauth = yes
client ntlmv2 auth = yes
lanmanauth = yes
ntlmauth = yes
security = user
os level = 40
ldapssl = off
ldappasswd sync = yes
passdb backend = ldapsam:ldap://127.0.0.1
ldap admin dn = cn=admin,dc=trav,dc=local
ldap suffix = dc=trav,dc=local
ldap group suffix = ou=Groups
ldap user suffix = ou=Users
ldap machine suffix = ou=Machines
add user script = /usr/sbin/smbldap-useradd -m "%u"
ldap delete dn = yes
delete user script = /usr/sbin/smbldap-userdel "%u"
add machine script = /usr/sbin/smbldap-useradd -w "%u"
add group script = /usr/sbin/smbldap-groupadd -p "%g"
#delete group script = /usr/sbin/smbldap-groupdel "%g"
```

```
add user to group script = /usr/sbin/smbldap-groupmod -m "%u" "%g"
delete user from group script = /usr/sbin/smbldap-groupmod -x "%u" "%g"
set primary group script = /usr/sbin/smbldap-usermod -g "%g" "%u"
logon path = \\%L\profiles\%U
logon drive = P:
logon home = \\%L\%U
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
case sensitive = No
default case = lower
preserve case = yes
short preserve case = Yes
#character set = iso8859-1
#domain admin group = @admin
dns proxy = No
wins support = Yes
winbind use default domain = Yes
ntacl support = Yes
msdfs root = Yes
hide files = /desktop.ini/ntuser.ini/NTUSER.*/
# autre possibilité "veto files = "

# Reglage de l'encodage des caracteres :
unix charset = iso-8859-15
display charset = iso-8859-15
dos charset = 850

[netlogon]
path = /home/dev/netlogon
writable = No
browseable = No
write list = Administrateur
```

[profiles]
path = /home/dev/profiles
browseable = No
writeable = Yes
profile acls = yes
create mask = 0700
directory mask = 0700

[homes]
comment = Repertoire Personnel
browseable = No
writeable = Yes

[partage]
comment = Repertoire commun
browseable = Yes
writeable = Yes
public = No
path = /home/partage

[printers]
comment = All Printers
path = /var/spool/samba
create mask = 0700
printable = Yes
browseable = No

[print$]
comment = Printer Drivers
path = /var/lib/samba/printers

Une fois que le fichier est bien personnalisé, vérifiez s'il est bien valide avec le programme

testparm

Si le fichier de configuration est bien valide, vous pouvez alors demander à Samba de relire son fichier de configuration :

/etc/init.d/samba reload

Reloading /etc/samba/smb.conf (smbd only).

Note : Il n'y a pas de processus samba qui tourne en tâche de fond pour le serveur, mais deux processus : smbd qui s'occupe des partages et de l'authentification, et nmbd qui s'occupe de la résolution des noms NetBIOS (Network Basic Input Output System). [19]

- *Mot passe LDAP vers SAMBA*

Afin d'administrer correctement le serveur LDAP samba aura besoin du mot de passe administrateur de celui-ci. Pour cela tapez la commande suivante et donné le mot de passe administrateur du ldap.

    # smbpasswd -w *****

Samba doit vous répondre :

Setting stored password for "cn=admin,dc=trav,dc=local" in secrets.tdb

- *configuration des outils smbldap-tools*

smbldap-tools est un ensemble de script permettant de créer les utilisateurs POSIX de manière automatique. La configuration de celui-ci se fait par l'intermédiaire de deux fichiers qui ne sont pas encore présent sur notre système :
/etc/smbldap-tools/smbldap_bind.conf
/etc/smbldap-tools/smbldap.conf

Il faudra donc les créer manuellement.

Voici le détail du fichier smbldap_bind ci-dessous :

masterDN="cn=admin,dc=trav,dc=local"

masterPw="toto"
slaveDN="cn=admin,dc=trav,dc=local"
slavePw="toto"

Le mot de passe administrateur du LDAP est en clair, il est donc important de lui changer les droits d'accès.

chmod 600 /etc/smbldap-tools/smbldap_bind.conf

Le fichier smbldap.conf, contient un peu plus de paramètre et a aussi besoin du SID du domaine. Pour cela faire un :

netgetlocalsid

On doit obtenir :

SID for domain TRAV is: S-1-5-21-738282185-3837675623-175508106

Cette valeur est pour mon cas.

S'il y a des erreurs, cela veut dire qu'on a oublié un paramètre dans votre fichier smb.conf. [10]Une fois le SID en poche il ne reste plus qu'a faire le fichier de configuration comme ci-dessous :

SID="S-1-5-21-738282185-3837675623-175508106"
masterLDAP="127.0.0.1"
masterPort="389"
slaveLDAP="127.0.0.1"
slavePort="389"
ldapTLS="0"
verify="require"
suffix="dc=trav,dc=local"
usersdn="ou=Users,${suffix}"
computersdn="ou=Machines,${suffix}"
groupsdn="ou=Groups,${suffix}"
idmapdn="ou=Idmap,${suffix}"

```
# La ligne ci-dessous est commentée pour éviter une erreur lors de
# l'execution de la commande smbldap-populate.
# sambaUnixIdPooldn="cn=NextFreeUnixId,${suffix}"
#sambaUnixIdPooldn="sambaDomainName=TRAV,${suffix}"
scope="sub"
hash_encrypt="SSHA"
crypt_salt_format="%s"
userLoginShell="/bin/bash"
userHome="/home/%U"
userHomeDirectoryMode="700"
#Nom d'affichage - utiliser smbldap-useradd -c
userGecos="User"
defaultUserGid="513"
defaultComputerGid="515"
skeletonDir="/etc/skel"
#Les mots de passe expirent dans 10ans
defaultMaxPasswordAge="3650"
with_smbpasswd="0"
smbpasswd="/usr/bin/smbpasswd"
with_slappasswd="0"
slappasswd="/usr/sbin/slappasswd"
# mk_ntpasswd="/usr/local/sbin/mkntpwd"
```

Le paquet smbldap-tools contient plusieurs scripts pour gérer les utilisateurs, groupes et machines dans l'annuaire LDAP.

smbldap-groupadd smbldap-groupshow smbldap-useradd smbldap-usermod

smbldap-groupdel smbldap-passwd smbldap-userdel smbldap-usershow

smbldap-groupmod smbldap-populate smbldap-userinfo

- *smbldap-populate*

La commande smbldap-populate va créer notre arbre de base au sein du LDAP. Pour cela il suffit de faire :

smbldap-populate

Le résultat doit être:

Populating LDAP directory for domain trav (S-1-5-21-738282185-3837675623-175508106)
(using builtin directory structure)

entry dc=trav,dc=local already exist.
adding new entry: ou=Users, dc=trav,dc=local
adding new entry: ou=Groups, dc=trav,dc=local
adding new entry: ou=Machines, dc=trav,dc=local
adding new entry: ou=Idmap, dc=trav,dc=local
adding new entry: uid=root,ou=Users, dc=trav,dc=local
adding new entry: uid=nobody,ou=Users, dc=trav,dc=local
adding new entry: cn=Domain Admins,ou=Groups, dc=trav,dc=local
adding new entry: cn=Domain Users,ou=Groups, dc=trav,dc=local
adding new entry: cn=Domain Guests,ou=Groups, dc=trav,dc=local
adding new entry: cn=Domain Computers,ou=Groups, dc=trav,dc=local
adding new entry: cn=Administrators,ou=Groups, dc=trav,dc=local
adding new entry: cn=Account Operators,ou=Groups, dc=trav,dc=local
adding new entry: cn=Print Operators,ou=Groups, dc=trav,dc=local
adding new entry: cn=Backup Operators,ou=Groups, dc=trav,dc=local
adding new entry: cn=Replicators,ou=Groups, dc=trav,dc=local
entry sambaDomainName=trav,dc=trav,dc=local already exist. Updating it...

Please provide a password for the domain root:
Changing UNIX and samba passwords for root
New password:
Retype new password:

Si on détecte des messages d'erreurs le schéma samba n'est pas présent dans les schémas du serveur openldap.

3.2.6 Installation complète d'un serveur web sous Debian

On peut verifier le fonctionnement de l'annuaire en installant le paquet phpldapadmin :

aptitude install phpldapadmin

Lancer phpldapadmin [8] sur un navigateur web et taper l'adresse suivante dans la barre de liens: http://<adresse_IP_du_serveur_ldap>/phpldapadmin

Figure 3.04 : *Page d'accueil de phpldapadmin*

Figure 3.05 : *Authentification au serveur LDAP*

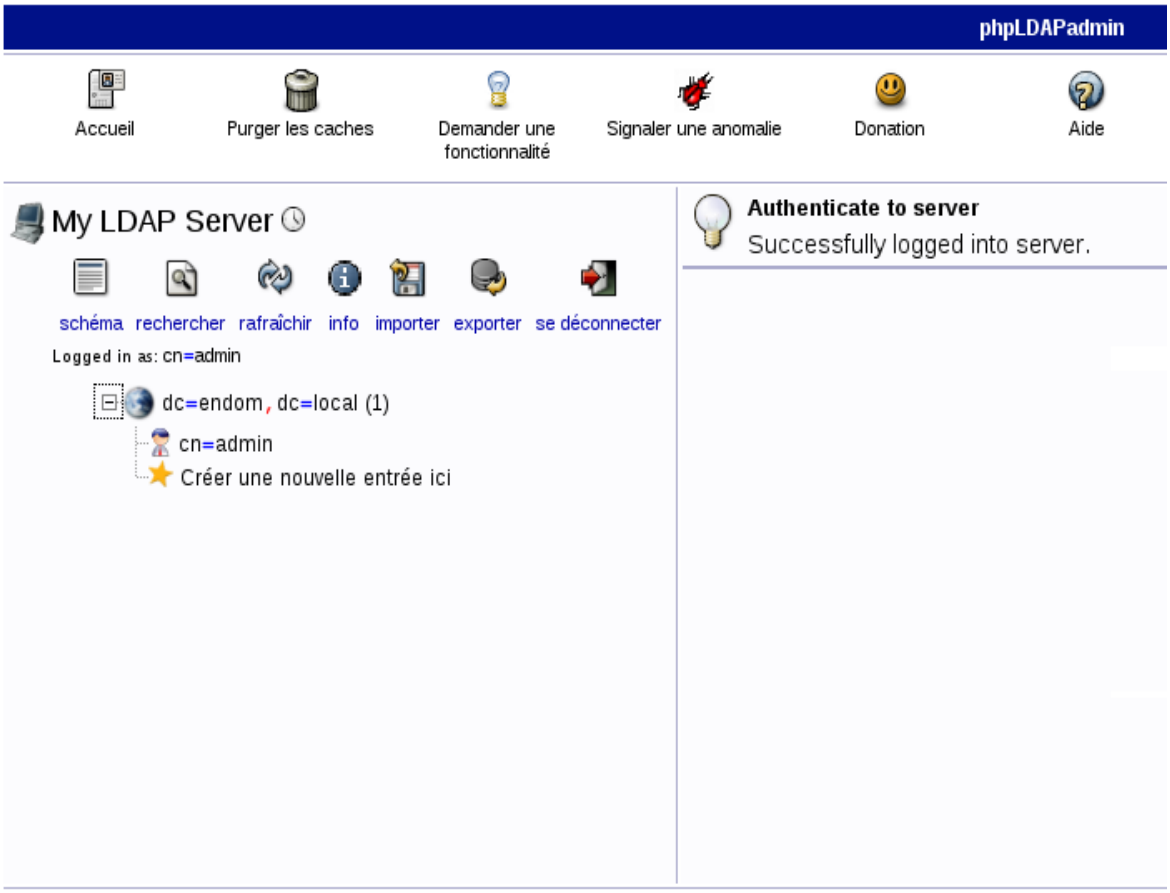

Figure 3.06 : *Premier aperçu de l'annuaire LDAP*

3.3 Simulation

3.3.1 But

Le but de la simulation est de réaliser un serveur linux et des machines clientes sous windows et de faire la gestion d'accès aux fichiers des machines clientes.

3.3.2 Outils de simulation

Pour éviter l'utilisation de plusieurs matériaux réseaux, on utilise un logiciel pour créer des machines virtuelles et pour les faire en réseau. Donc, on a choisi le logiciel Oracle VM Virtualbox License version 8, April 19, 2010.

Après avoir installé ce logiciel, on doit avoir une interface graphique comme ci-dessous :

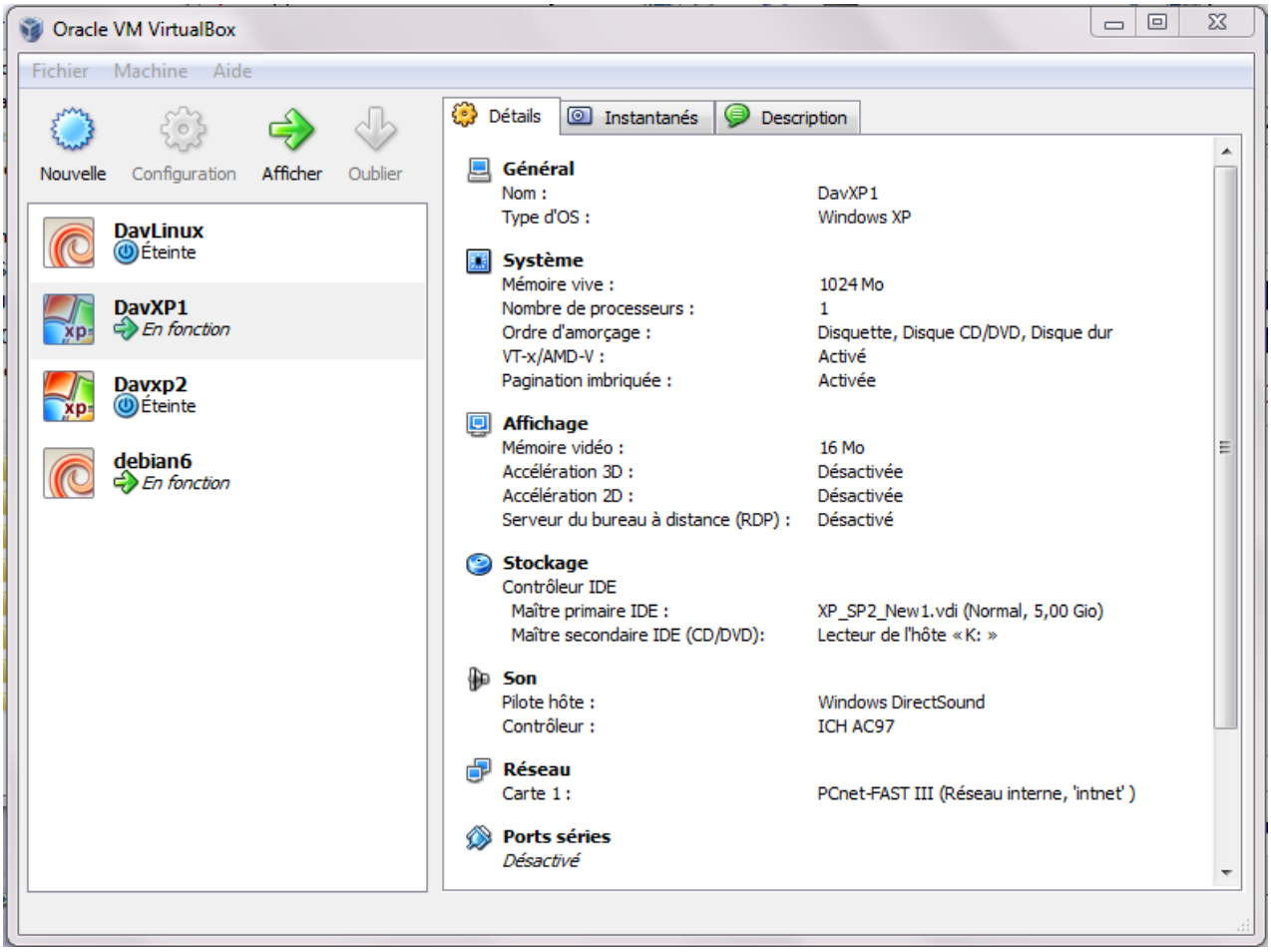

Figure 3.07 : *Oracle VM VirtualBox*

Alors, il suffit de cliquer sur le bouton « nouvelle » pour créer une nouvelle machine virtuelle. La figure suivant montre le démarrage d'une machine virtuelle :

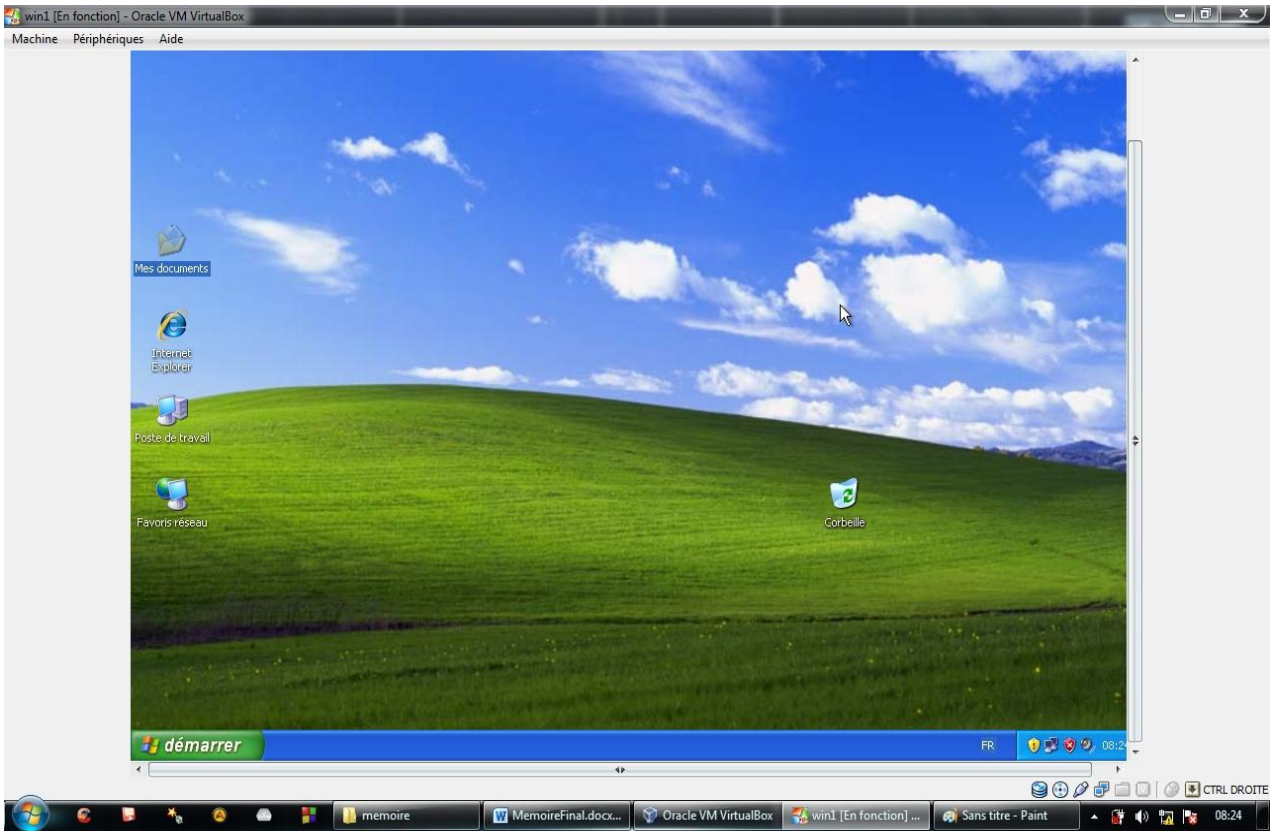

Figure 3.08 : *Lancement d'une machine virtuelle*

On doit maintenant installer les autres machines virtuelles et configurer Oracle VM VirtualBox pour que les machines soient en réseau.

3.3.3 Les étapes de la simulation

3.3.3.1 But

Le travail à suivre est de faire toutes configurations utiles pour gérer le droit d'accès aux fichiers du réseau local pour les groupes et les utilisateurs.

3.3.3.2 La gestion des groupes et des utilisateurs

Premièrement, on doit intégrer les machines windows au domaine. Une fois que vous avez terminé toute l'installation et la configuration de votre serveur Linux, pensé à relancer les deux services:

 # /etc/init.d/samba stop

 # /etc/init.d/slapd restart

 # /etc/init.d/samba start

Enfin, vous pouvez tester sous Windows.

- Click droit sur *Poste de Travail, Propriétés.*
- Onglet *Nom de l'ordinateur*
- Click sur le bouton *Modifier…*
- Membre de: Choisissez *Domaine* : Saisissez le nom de la directive workgroup que vous avez renseigné dans smb.conf
- OK
- On vous demande un login: *root* et votre mot de passe : celui de root
- Si tout va bien le message » Bienvenue dans le domaine …. » s'affiche.
- Redémarrez la machine
- Choisissez le domaine que vous avez créé et connectez-vous avec *root.*

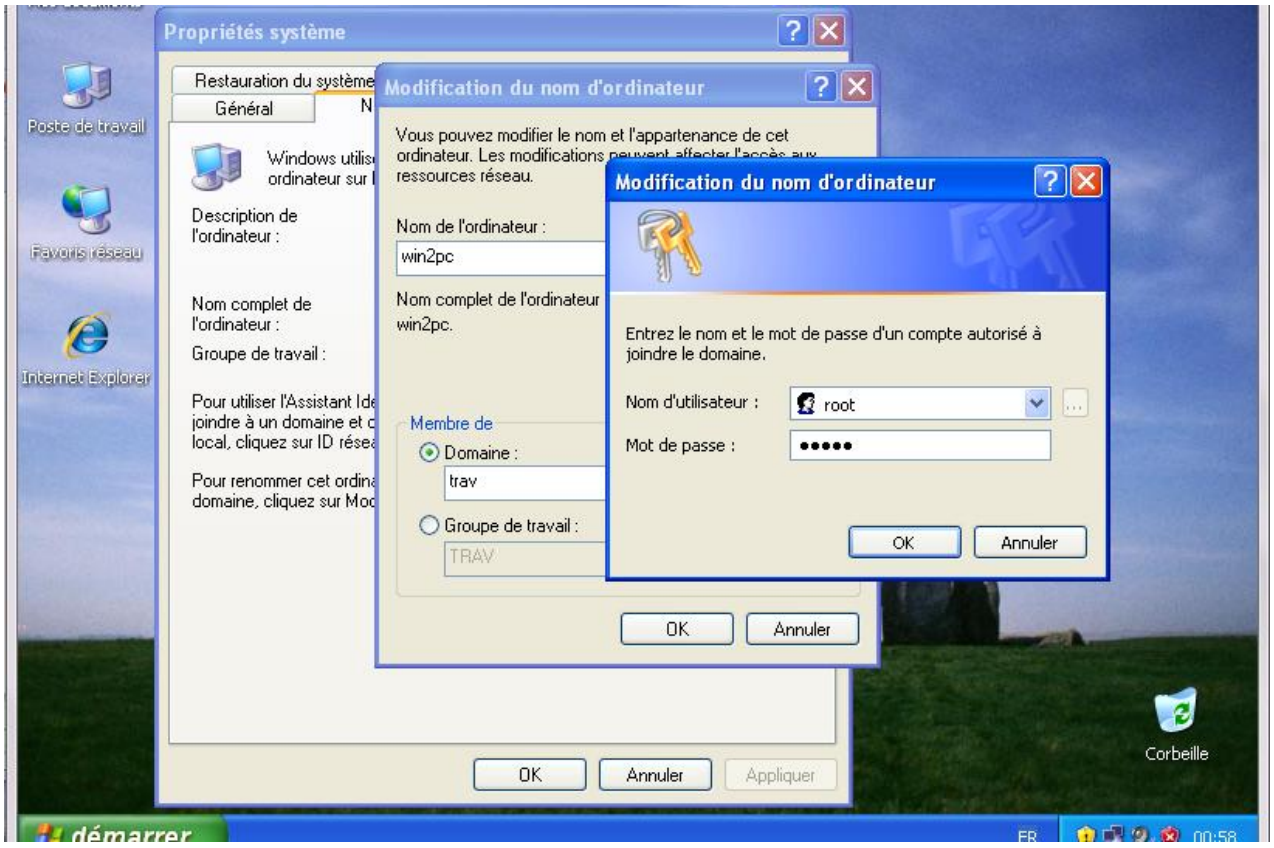

Figure 3.09 : Intégration des machines windows

Dans Organizational Unit (OU=Machines) [9] que doit contenir les machines windows.

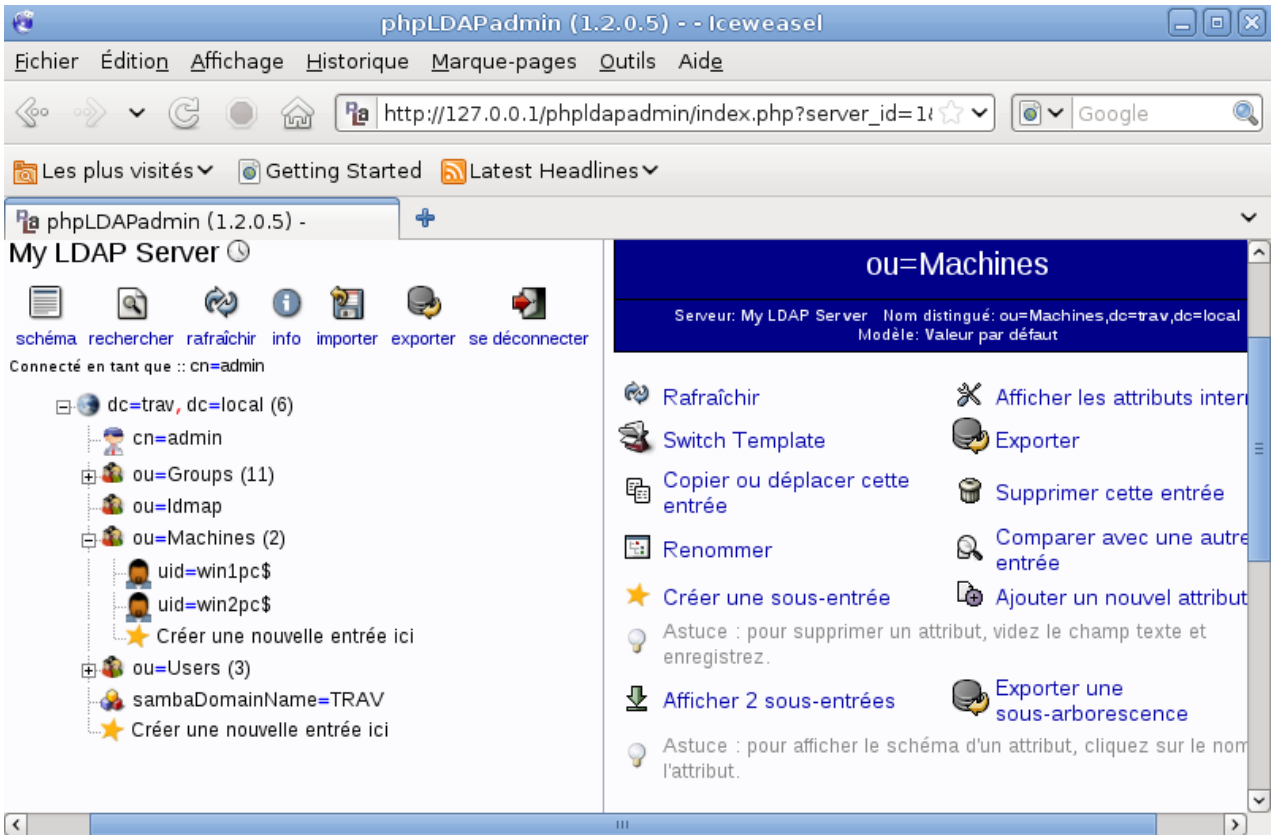

Figure 3.10 : Affichage des machines windows dans phpldapadmin

On va maintenant définir quels utilisateurs pourront venir se connecter sous Samba avec un profil errant. Ajoutons un utilisateur *user*. Notez qu'user doit être un utilisateur UNIX.

smbldap-useradd -g smb user

smbpasswd -a user

On créé un script appelé user.bat dans /var/samba/netlogon
Il s'agit d'un script BATCH DOS qui s'exécutera à l'ouverture de la session.
Par exemple :

- REM réglage heure windows sur serveur samba
- NET TIME \\NOM_MACHINE /SET /YES
- REM pour monter le home de user sur H:

- NET USE H: \\NOM_MACHINE\user
- REM exécuter des clés de registre
- regedit /s \\NOM_MACHINE\netlogon\aqs.reg

C’est le moment pour connecter un machine windows avec un utilisateur Unix/samba :
Loguer vous en tant que user au domaine TRAV avec votre mot de passe Samba.
Soyez attentif aux messages d'erreurs éventuels, vous aurez la première fois un message : *%REMOVE_NODIR_ERROR%* (constaté sous Win XP pro non SP1) ce qui est normal car il n'avait pas de profile jusqu'ici. Il crée ainsi un répertoire user dans /var/samba/profiles contenant le profil errant d'user.
Bien vérifier ensuite que tout se passe bien à la connexion/déconnexion des utilisateurs Windows. Beaucoup de problèmes sont des problèmes de droits UNIX mal réglés.
Notez qu'ici root peut très bien se loguer. Cela peut sembler problématique même si il n'accède qu’aux ressources partagées par Samba. Mais bon, vous utilisez Windows, vous êtes déjà à un niveau de sécurité critique.

3.3.4 Conclusion

Toutes les serveurs sont maintenant complets et prêts à être utilisés, nous pouvons maintenant organiser nos groupes et utilisateurs pour faire les droits d’accès aux fichiers en utilisant les commandes pour les créer et les commandes pour changer les droits aux fichiers.

CONCLUSION GENERALE

Ce travail a pour objet de gérer les données d'un réseau local. Tout utilisateur dans le réseau local doit pouvoir accéder à toute information utile à sa tâche à condition qu'il ait besoins de s'identifier et de s'authentifier, on a donc choisi d'utiliser l'annuaire openldap. Les utilisateurs peuvent lire, écrire et/ou exécuter un fichier. Toute tache peut être enregistrée grâce à samba et on peut informer la hiérarchie du réseau dans un domaine grâce à un annuaire ldap. L'accès à l'information doit avoir lieu par une interface aussi simple que possible. Nous avons vu les démarches à suivre pour installer, configurer et gérer les serveurs ldap et Samba sur le système d'exploitation Linux, plus précisément sur Debian. Il ne reste plus qu'à paramétrer ces différents serveurs pour faciliter les travailles sur le serveur web.

ANNEXE 1 :

PRINCIPAUX RÉPERTOIRES SYSTÈMES

A1.1Répertoires standards :

/ Répertoire racine (ou root) contenant tous les répertoires.

/home Répertoire contenant les répertoires personnels de tous les utilisateurs autres que root.

/root Répertoire personnel de l'administrateur système root.

A1.2 Répertoires système :

/bin Répertoire contenant les commandes et utilitaires employés par tous les utilisateurs (ls, rm, cp, etc...)

/boot Répertoire contenant des informations permettant le chargement de Linux.

/dev Répertoire contenant tous les fichiers périphériques permettant d'accéder aux composants matériels.

/etc Répertoire contenant les commandes et fichiers de paramétrages nécessaires à l'administration système.

/lib Répertoire contenant les bibliothèques communes à tous les utilisateurs

/proc Répertoire spécial utilisé par le système et contenant la liste des processus encours d'exécution.

/sbin Répertoire contenant les commandes et utilitaires utilisées seulement par l'administrateur système.

/tmp Répertoire contenant les fichiers temporaires.

/usr Répertoire composé d'un certain nombre de sous répertoires utilisés par l'ensemble des utilisateurs.

/var Répertoire spécial utilisé par le système pour stocker des données souvent modifiées.

ANNEXE 2 :

LES COMMANDES DE BASE SOUS LINUX

COMMANDE	DESCRIPTION	EQUIVALENT DOS
ls	Liste le contenu d'un répertoire	dir
cd	Change de répertoire	cd
cd..	Répertoire parent	cd..
mkdir	Crée un nouveau répertoire	md
rmdir	Supprime un répertoire	deltree
cp	Copie de fichier	copy,xcopy
mv	Déplacement de fichier	move
rm	Supprime le fichier	del
passwd	Change le mot de passe de l'utilisateur	
cat	Affiche le contenu du fichier	type
more	Affiche le contenu du fichier avec des pauses	type/more
man apropos	Aide sur la commande demandée	help
lpr	Imprime le fichier demandé	print
chmod	Change l'attribut d'un fichier chmod XXX fichier	

	XXX=Utilisateur|Groupe|Autres où X représente un entier Lecture=4, Ecriture=2, Exécution=1 X=Lecture+Ecriture+Exécution 0 signifie aucun droit 1 signifie droit d'exécution 2 signifie droit en écriture 3 signifie droit en écriture et en exécution 4 signifie droit en lecture 5 signifie droit en lecture et en exécution 6 signifie droit en lecture et droit en écriture 7 signifie tous les droits	
chfn	Change les informations personnelles vues avec finger	
chsh	Change le shell: *chsh user emplacement_du_shell*	
finger	Liste des utilisateurs en ligne	
traceroute	Trace le chemin entre la machine locale et la machine visée	

Tableau annexe 2 : *Commande de base sous linux*

ANNEXE 3:

LES SERVICES LES PLUS UTILES ET LEURS PORTS

Services	Ports utilisés	Protocoles utilisés
Echo	5	Tcp/udp
ftp	21	Tcp/udp
Ssh	22	Tcp/udp
telnet	23	Tcp/udp
Smtp	25	Tcp/udp
domain	53	Tcp/udp
http	80	Tcp/udp
Pop3	110	Tcp/udp
irc	194	Tcp/udp
https	443	Tcp/udp
shell	514	tcp

Tableau annexe 3 : *les services les plus utiles et leurs ports*

BIBLIOGRAPHIE

[1] A. Ratsimbazafy A., *Réseau Informatique*, Cours 2ème année, Dep. Tel.-ESPA, A.U. :2008- 2009

[2] L.E.Randriarijaona, *Réseau TCP/IP et Sécurité*, Cours 3ème année, Dep. Tel.-E.S.PA., A.U. :2009-2010

[3] P. Nicolas, « *cours de réseaux et Maitrise d'informatiques* », http : //www.info.univ-angers.fr/pub/pn, 1999-2000

[4] A. Amine and G. Abdeluahab, « *voix-ip* », promotion IGE 25 juin 2005

[5] A. Ranaivoarivony, « *mise en place d'un serveur web et FTP sécurisé sous linux* », Mémoire de fin d'étude, Dép.EN.-ESPA, AU : 2004/2005

[6] Rakotomavonjatovo, « *sécurité des réseaux sous linux* », Mémoire de fin d'étude, Dép.EN.-ESPA, AU : 2003/2004

[7] T. Adelstein and B. Lubanovic, « *Linux System Administration* », Copyright 2007 O'Reilly Media, Inc., 1005Gravenstein Highway North Sebastopol, CA 95472

[8] M. Vogehueith, « *Debian GNU/linux Annuaire LDAP-MDS* »,17 aout 2009

[9] H. Abdellaoui, « *Guide de migration de l'assistant de migration* », Copyright © Mandrakesoft, 2005

[10] R. TAUPIN, « Installation de Samba avec authentification LDAP sous Linux Debian Lenny », Gandi Hébergement le mercredi 11 mars 2009

[11] C. Caleca, « *les réseaux informatiques* », http://Christian.Caleca.free.fr/réseaux/, Laurant Baysse

[12] Comment ça marche? [Informatique]-CCM-L'encyclopedie informatique libre

[13] B. Péan, « *Support de cours Réseaux EISTI* », http://www.eisti.fr/, Avenu du Parc 95011 CERGY

[14] C. CALECA, « *TCP/IP* », http://christian.caleca.free.fr/tcpip/, Laurent Baysse 6 mars 2005

[15] F. Laissus, « *Cours d'introduction à TCP/IP* », http://fr.laissus@laissus.fr/TCP-IP(268)/, Version du 05 avril 2003

[16] Oya, « *Système d'exploitation Linux* », http://www.siteduzero.fr/Linux/Simple IT SARL crée: le 20/07/2005 modifié le 27/12/2008

[17] A. de Lattre, « *formation Debian GNU/linux*», http://people.via.ecp.fr/~alexis/formation-linux/, 23 décembre 2010

[18] G. Laplanche (Genesta) and D. Lacoste (EDF R&D), « *Mise en place d'un domaine Samba 2.2-LDAP* », http://www.martymac.com/samba 2.2Ldap/Avenue du Général De Gaulle F.92 141 Clamart Cedex, janvier 2004

[19] E. Berthomier, « *Formation Annuaire OpenLDAP* », http://eric.berthomier@free.fr/openldap/, 17 mars 2007

[20] L. Besson, « *Créer un Contrôleur Principal Domaine (Windows) avec SAMBA et un annuaire LDAP* », http://www.e-glop.net/main/GDM-LDAP-Ubuntu, Creative Commons License, 4 Novembre 2008

Résumé

Vu le développement et la décentralisation de toutes les grandes sociétés d'aujourd'hui, il est presque incontournable de faire relier toutes les sites ou agences d'une société par un réseau fiable et sécurisé. L'administration de ce réseau requiert donc un département et un personnel compétent capable de défendre le réseau contre toute attaque malveillante. Grâce à sa souplesse, sa fiabilité et sa stabilité, le système linux nous permet de renforcer cette sécurité.

Abstract

Considering the development and the decentralization of all the large companies of today, it is almost impossible to circumvent to make connect all the sites or agencies of a company by a reliable and protected network. The administration of this network requires a department and qualified personnel able to defend the network against any malevolent attack. Thanks to its flexibility, its reliability and its stability, the Linux system enables us to reinforce this safety