Ministère des Enseignements Secondaire
Supérieur et de la Recherche Scientifique
(MESSRS)

Université Polytechnique de Bobo-Dioulasso(UPB)

Ecole Supérieure d'Informatique (ESI)

Cycle des Ingénieurs de Travaux Informatiques(CITI)

Option: Réseaux et Maintenance Informatiques (RéMI)

BURKINA FASO

Unité-Progrès-Justice

Agence universitaire de la Francophonie

Projet de fin de cycle

THEME:

GESTION CENTRALISÉE DES RESSOURCES INFORMATIQUES DE L'UPB AVEC LE PROTOCOLE LDAP

Présenté et soutenu par:

M[r] AGBEZOUTSI Kodjo Edem
et
M[r] DIARA Ahmed Alex Carlox

étudiants en fin de cycle RéMI

Année académique: 2006-2007

Superviseur:

Dr Borlli Michel Jonas SOME
Enseignant-Chercheur à l'ESI

Maître de stage:

M[r] Jean-Baptiste MILLOGO
Responsable du CAI de Bobo-Dsso

DÉDICACE

À

Nos parents respectifs qui nous ont soutenus tout au long de notre formation

REMERCIEMENTS

Nous tenons à exprimer notre profonde gratitude à toutes les personnes dont le soutien a rendu ce stage possible. Il s'agit notamment de:

Monsieur Jean-Baptiste MILLOGO, Responsable du Centre d'Accès à l'Information de Bobo-Dioulasso, notre maître de stage, pour les moyens qu'il a constamment mis à notre disposition, pour nous avoir guidés dans nos recherches et pour les corrections apportées au rapport final;

Dr Borlli Michel Jonas SOME Enseignant-Chercheur à l'ESI, notre superviseur, pour ses apports qui ont considérablement facilité notre étude et ses suggestions qui nous ont été d'une très grande utilité dans la rédaction du rapport final;

Monsieur Pasteur PODA, enseignant à l'ESI, pour ses précieuses corrections et suggestions ;

Monsieur Yacouba OUATTARA, étudiant en DEA à l'ESI pour ses précieuses suggestions.

Nous n'oublions pas de remercier l'**Ecole Supérieur d'Informatique** pour la formation que nous avons reçu,

Et toutes les personnes qui ont, de près ou de loin, contribué à faire de ce stage une réussite.

Résumé

Les lignes ci-dessous se veulent être une brève description de ce que nous allons développer dans cet écrit. Fruit des trois mois de travail que nous avons effectué au Centre d'Accès à l'Information (CAI) de Bobo-Dioulasso pour la mise en place d'un système informatique global à l'Université Polytechnique de Bobo-Dioulasso (UPB), nous l'avons subdivisé en trois grandes parties. La première présentera les structures cadres de notre formation à savoir, l'Ecole Supérieur d'Informatique et le CAI. Dans la deuxième partie, nous entrerons dans le vif du sujet par une étude critique du patrimoine technologique de l'UPB. Cela nous permettra d'en ressortir les limites afin de proposer une solution plus adaptée aux besoins de l'université. Quant à la troisième partie qui sera la plus consistante de notre travail, elle nous permettra de mener une étude approfondie des solutions envisageables pour la mise en place du système informatique escompté. Dans cette partie nous aurons trois grands axes: le premier consistera à définir des architectures physique puis logique du système futur. Dans le second, nous nous attellerons à étudier les services réseaux qui seront déployés en commençant par une définition de chaque service; puis nous passerons au choix dûment justifié du système d'exploitation et des applications serveurs les plus adaptées pour l'implémentation des services définis; ces applications et leur mise en oeuvre sécurisée feront également l'objet d'une étude détaillée, avant que nous ne passions aux estimations du temps et des moyens nécessaires à la réalisation effective de notre projet d'étude. Le troisième grand axe de cette partie sera un bilan des travaux effectués, qui en fera un récapitulatif pour finir par des perspectives permettant l'amélioration de l'étude.

Abstract

This is a summary of what will be developed in this writing. It is about a work which took us three months in Centre d'Accès à l'Information (CAI) of Bobo-Dioulasso to study about setting up a Global Computing Network in the Polytechnic University of Bobo-Dioulasso (UPB). This writing will be divided in three main parts. The first one will make a presentation, respectively, of the High School of Computer Science (Ecole Supérieure d'Informatique -ESI- for french) where we have been studying for three years, and CAI, where we conducted this work. The second part will first deal with the report of some investigations that we made in the university, and which showed that the available resources of Information Technology (IT) are insufficient and inefficient for a suitable communication in the whole university. Then in this part we will give some solutions which are more adapted to the needs of the Polytechnic University of Bobo-Dioulasso in terms of IT. In the third part of this report, we will make a detailed study of the solutions given in the previous part. Being the most substantial part, it will be divided in three paragraphs. The first one will define the physical, then the logical architecture of what our computing network will be in the future. In the next paragraph we will make a choice of the most suitable Operating System and sofwares which can be used to implement the network services, after defining what these services are. The end of this paragraph will be consecrated to some estimations in terms of time and ressources which will be needed for the achievement of this project of study. The last paragraph will make an assessment of the whole study. It will first show the accomplished part of the work, then it will end by some perspectives in order to fulfill the study and make its achievement easy.

Table des matières

INTRODUCTION GENERALE

Pour compléter leur formation et en vue d'obtenir le diplôme d'Ingénieurs des Travaux Informatiques(ITI) option Réseaux et Maintenance Informatiques(RéMI), les étudiants en fin de cycle à l'Ecole Supérieure d'Informatique(ESI) doivent effectuer un stage pratique de trois(03) mois en entreprise. C'est dans ce cadre que nous avons travaillé au Centre d'Accès à l'Information (CAI) de l'Agence Universitaire de la Francophonie(AUF) sur le thème :« **Gestion centralisée des ressources informatiques de l'Université Polytechnique de Bobo-Dioulasso(UPB) avec le protocole LDAP».**

L'Université Polytechnique de Bobo-Dioulasso (UPB), deuxième université du Burkina Faso après celle de Ouagadougou, n'est pas encore dotée d'un Système Informatique à même de répondre convenablement aux besoins de communication de ses acteurs et de faciliter les enseignements qui y sont donnés. C'est en vue de contribuer à la mise en place d'une telle solution qu'il nous a été confié de mener une étude sur l'amélioration du Système Informatique de l'UPB et la gestion centralisée de ses ressources informatiques.

I. PRÉSENTATION DE L'ESI

1) Présentation Générale

L'Ecole Supérieure d'Informatique (ESI), créée en 1991, a d'abord été implantée à Ouagadougou, ensuite elle a été transférée au sein de l'Université Polytechnique de Bobo-Dioulasso (UPB) en septembre 1995.

2) La mission de l'ESI

L'ESI a pour mission:

- ✓ la formation fondamentale, appliquée et/ou professionnelle dans les domaines de l'informatique;
- ✓ la formation continue;
- ✓ la recherche scientifique et technologique ainsi que la valorisation des résultats de la recherche;
- ✓ la diffusion de la culture et de l'information dans les domaines relevant de sa compétence;
- ✓ la collaboration avec d'autres structures de formation et/ou de recherche pour la préparation des diplômés;
- ✓ la participation à des programmes internationaux de formations et de recherche.

3) Formations

L'ESI offre trois types de formations qui sont sanctionnées par les diplômes suivants :

- ✓ **Un diplôme d'ingénieur de travaux informatiques,**
- ✓ **Un diplôme d'ingénieur de conception en informatique,**
- ✓ **Un diplôme d'études approfondies en informatique.**

3.1 Le Cycle des Ingénieurs de Travaux Informatiques (CITI)

La durée de formation pour ce cycle est de trois (3) ans. Les étudiants doivent effectuer pendant leur formation des stages obligatoires: suivre un stage pratique et

réaliser un projet de fin de cycle. Les stage pratique vise à garantir une intégration rapide des futurs diplômés en milieu professionnel et doivent s'effectuer en entreprise au cours des huit (8) dernières semaines de la deuxième année d'étude pour les étudiants admissibles (étudiants ayant obtenu une moyenne de classe supérieure ou égale à 12/20). Les étudiants déclarés admissibles en troisième année d'étude réaliseront un projet de fin de cycle sur l'utilisation des techniques informatiques dans un secteur d'activité d'une entreprise publique, privée ou dans une administration. Ce projet se déroule sur trois (3) ou quatre (4) mois, généralement de juin à septembre et fait l'objet d'une soutenance publique pour l'obtention définitive du diplôme d'Ingénieur de Travaux Informatiques.

Le CITI comporte deux options qui sont :

- Analyse et Programmation (AP), créée en 1990,
- Réseaux et Maintenance Informatique (RéMI), créée en 2000.

3.2 Le Cycle des Ingénieurs de Conception en Informatique (CICI)

Pour ce cycle, la durée de la formation est de deux (2) ans. Elle est ouverte aux titulaires d'un diplôme de niveau Bac+3 en informatique. Au niveau du CICI également, les étudiants doivent effectuer pendant leur formation de première année un stage en entreprise et réaliser un mémoire de fin de cycle en 2ème année. Le stage est obligatoire et a lieu en entreprise durant les huit dernières semaines de la première année d'études pour les étudiants déclarés admissibles, et ceux admissibles en deuxième année sont tenus de réaliser un mémoire de fin de cycle. La production de ce mémoire se déroule en six (6) mois dont une phase pratique de trois (3) mois en entreprise ou dans un laboratoire et fait l'objet d'une soutenance publique.

3.3 Le DEA (Diplôme d'Etudes Approfondies)

Créé en 2003, ce cycle vise à Initier les étudiants aux concepts fondamentaux de la recherche en Informatique en vue d'étoffer le corps enseignant de l'ESI.

4) Organisation et structure

L'Ecole Supérieure d'Informatique est placée sous l'autorité pédagogique et administrative d'un Directeur assisté d'un Directeur Adjoint.

II. PRÉSENTATION DU CAI/AUF

1) Présentation de l'Agence Universitaire de la Francophonie (AUF)

Fondée à Montréal (Canada) en 1961, l'Agence Universitaire de la Francophonie (AUF) est une institution multilatérale qui soutient la coopération et la solidarité entre les institutions universitaires travaillant en français, prioritairement avec les pays francophones d'Afrique, du Monde arabe, d'Asie du Sud-Est, d'Europe Centrale et Orientale et de la Caraïbe. Elle contribue au développement de l'enseignement supérieur et de la recherche.

658 membres (universités publiques et privées, instituts d'enseignement supérieur, centres ou institutions de recherche, réseaux institutionnels et réseaux d'administrateurs liés à la vie universitaire), répartis dans les pays appartenant à l'OIF(Organisation Internationale de la Francophonie) et au-delà, sont membres de l'AUF. À ces membres, il convient d'ajouter un réseau de plus de 350 départements d'études françaises d'établissements universitaires du monde entier.

1.1 Les programmes de l'AUF

L'ensemble des établissements membres de l'AUF constitue un réseau unique de partenaires qu'elle fédère et anime à travers ses cinq programmes d'actions et de soutien qui sont :

- Programme « Langue française, diversité culturelle et linguistique »,
- Programme « aspects de l'Etat de droit et démocratie »,
- Programme « Soutien des TICs au développement de l'enseignement supérieur et de la recherche »,
- Programme « Soutien et renforcement de l'enseignement universitaire »,
- Programme « Environnement et développement durable solidaire »

1.2 Mission de l'AUF

L'A.U.F poursuit les missions suivantes:

- Associer au plan international les universités, organismes et institutions d'enseignement supérieur et de recherche travaillant en français,
- Structurer cet ensemble en favorisant les rassemblements régionaux, la constitution de réseaux et toutes formes de partenariats,
- Soutenir les activités associatives en vue d'une meilleure connaissance réciproque et d'une plus grande solidarité entre les institutions membres;
- Développer la mobilité des étudiants, des enseignants et des chercheurs au sein de l'espace universitaire francophone;
- Promouvoir l'utilisation massive en français des nouvelles technologies de l'information, de la communication et de l'enseignement à distance;
- Renforcer la solidarité mondiale entre les départements universitaires d'études françaises et entre les universités de groupes linguistiques différents en vue de la promotion de la diversité linguistique;
- Apporter une aide particulière aux institutions les moins favorisées, spécialement a celles qui sont nouvellement créées ou menacées dans leur existence;
- Offrir des prestations de service à l'intérieur et à l'extérieur de la Francophonie.

2) Présentation du Centre d'Accès à l'Information de Bobo-Dioulasso (CAI)

Les Centres d'Accès à l'Information de l'AUF découlent de son programme « Soutien des TICs au développement de l'enseignement supérieur et de la recherche ». Celui de Bobo-Dioulasso est la deuxième implantation de l'AUF au Burkina Faso après celle de Ougadougou. C'est un service directement rattaché à l'Agence Universitaire de la Francophonie (A.U.F) tout comme le Campus Numérique Francophone de Ouagadougou. Le centre d'accès à l'information de Bobo-Dioulasso a été inauguré le 20 mars 2004 et ouvert ses portes au public le 24 mai de la même année.

Le C.A.I met à la disposition des usagers l'accès à l'information scientifique et technique, des formations en informatique, des formations (avec un diplôme à l'issue de la formation) à distance; il sert de centre relais pour la diffusion des appels d'offre de tous

les autres programmes de l'AUF.

Le centre oeuvre également en faveur des activités de recherche, d'enseignement, de développement technique, de transfert de technologies, de diffusion d'informations scientifiques, techniques et culturelles, d'expérimentations de nouveaux services à caractères innovant, mais aussi toute activité administrative et de gestion découlant ou accompagnant celles susmentionnées.

2.1 Les prestations du CAI de Bobo-Dioulasso

2.1.1 Les Formations Ouvertes et à Distance

Dans son programme transversal pour accompagner l'entrée dans la société de la connaissance en réduisant la fracture numérique, tout en renforçant les capacités humaines, le C.A.I donne l'opportunité au public de suivre des Formations Ouvertes et A Distance (FOAD).

A cet effet une salle est réservée aux apprenants ayant été retenus pour une formation à distance à l'issue d'un appel à candidature.

La liste complète des offres de formation à distance est consultable à http://foad.refer.org/

2.1.2 La bibliothèque et la salle informatique

Le Centre d'Accès à l'Information met à la disposition de ses abonnés un certain nombre de documents tels que des ouvrages pédagogiques et aussi des revues scientifiques. Ces documents sont pour la plupart à consulter sur place mais les abonnés peuvent également être autorisés à en emprunter.

Une salle informatique dénommée « infothèque » dotée de dix (10) machines connectées à l'Internet et plusieurs prises destinées au branchement libre d'ordinateurs portables donne l'accès à l'Internet et permet aux abonnés de faire des travaux de bureautique. Ils peuvent également faire des impressions/scannages à des prix étudiés.

2.2 Le système informatique du CAI de Bobo-Dioulasso

2.2.1. Les ressources humaines

Le personnel du C.A.I se compose du responsable et d'un agent de liaison. C'est

donc dire qu'en plus de ses tâches administratives, le Responsable est l'administrateur réseau afin d'assurer le bon fonctionnement des installations. Mais, en vue de renforcer ses capacités humaines dans le domaine des Technologies de l'Information et de la Communication (TICs) et des réseaux, le C.A.I a mis en place une politique d'accueil régulier de stagiaires.

Le Responsable du C.A.I est donc le garant du bon fonctionnement administratif et technique du centre; il organise différentes formations qui ont lieu au centre. s'occupe de la diffusion des documents primaires. Il s'occupe des actions de sensibilisations auprès des structures universitaires et de recherches afin de mieux faire connaître les opportunités offertes par l'Agence Universitaire de la Francophonie.

2.2.2. Les ressources informatiques et les services réseaux

2.2.2.1 Les ressources informatiques

Le réseau local du C.A.I est de type Ethernet avec une topologie en étoile. En effet, un câblage structuré de chaque salle permet de la relier à un des switchs du local technique. Le centre dispose de deux types de connexion à Internet: l'ADSL à 512 kbit/s et une liaison spécialisée à 256 kbit/s, toutes deux souscrites à l'Office National des Télécommunications (ONATEL). Les composants matériels et logiciels sont décrits dans les tableaux ci-dessous.

- Le matériel

Tableau I.1: récapitulatif des équipements du CAI

Matériels	**Marque/Type**	**Caractéristiques**	**Nombre**
Câblage	Paire torsadée	Catégorie 5	-
Postes Clients	ASUS Barebone Terminator k7	Processeur:Intel Pentium III RAM:512 Mo DD:40 Go Écran:15 pouces	16
Poste Serveur	COGIDIS	Processeur:AMD RAM:512 DD:40 Go Écran:15 pouces	01

Routeur	Cisco 2610	17 ports Ethernet 02 ports séries 01 port console 01 port AUX	01
Modem LS	Nokia DNT 1 M		01
Modem ADSL	3COM	Wi-Fi(802.11g), 4 ports Ethernet, Firewall intégré.	01
Switch	D-Link	10/100 Fast Ethernet, 24 ports	01
Switch	D-Link	10/100 Fast Ethernet, 08 ports	02
Imprimantes réseaux	HP LaserJet 1300 HPLaserJet P2015n	SDRAM installée, 16Mo, 510Feuilles. SDRAM installée, 16Mo, 510Feuilles.	01 01
Imprimante couleur	HP Deskjet 5652	Recto-verso,capacité :150 feuilles	01
Scanneur	HP scanjet 5530	Scanneur à plat, couleur et monochrome;24 000 x 48 000 ppp	01
Appareil de visioconférence	Polycom VX500	Audio :Full duplex,Norme :H261	01

- Les logiciels

Tableau I.2: récapitulatif des logiciel et système d'exploitation du CAI

Logiciels	**Type**
Système d'exploitation (poste serveur)	Debian 4.0 (Etch)
Système d'exploitation (poste client)	Ubuntu 7.10 (Gutsy)
Bureautique	OpenOffice.org

2.2.2.2 Les services réseaux

L'enjeu des services réseau est de pouvoir mettre à disposition une ressource sur le réseau et de la rendre accessible pour tous les postes ou logiciels clients qui émettent une requête dans ce sens.

Ainsi les différents services disponibles sur le Serveur du Centre sont les suivants:

Tableau I.3: récapitulatif des services réseaux du CAI

Serveurs	**Caractéristiques**
Partage de la connexion Internet	via l'ADSL et la LS
DHCP	Configuration dynamique des postes clients
DNS (Bind9)	un serveur permettant d'associer un nom aux adresses IP des ordinateurs du réseau.
Postfix	Serveur de messagerie électronique
Apache	Serveur Web
FTP	Serveur de transfert de fichiers
Asterisk	Serveur destiné pour la Voix sur IP

DEUXIEME PARTIE : ANALYSE DU SYSTEME INFORMATIQUE EXISTANT

Notre étude ne saurait se faire sans prendre connaissance du Système Informatique dont l'UPB est actuellement dotée. Dans les lignes qui suivent, nous allons dans un premier temps faire une étude de l'existant dans le domaine des Technologies de l'Information et de la Communication (TIC) en mettant l'accent sur les éléments qui pourraient servir de base à la mise en oeuvre du projet de notre étude. Ensuite nous ferons une étude critique des moyens déjà déployés. Nous finirons par la proposition d'une solution plus adaptée aux besoins de communication et d'apprentissage à l'UPB.

I. ÉTUDE DE L'EXISTANT

Dans cette partie, nous utiliserons les données fournies par l'étude dont nous avons fait cas dans l'introduction générale.

1) Le patrimoine en Technologie de l'Information et de la Communication (TIC) de l'UPB

Il est constitué du matériel informatique, des infrastructures réseaux, des logiciels et services réseaux mis en place.

1.1 Le matériel informatique

Une grande partie du matériel informatique de l'UPB se retrouve dans ses salles de travaux pratiques(TP), notamment celles de l'Ecole Supérieure d'Informatique (ESI) et l'Institut Universitaire de Technologie(IUT). On y trouve respectivement quarante (40) et trente (30) ordinateurs environ en plus d'une vingtaine de postes dans la salle informatique de l'Institut du Développement Rural (IDR). De plus, toutes les administrations des instituts et écoles se trouvent dotées d'un minimum de trois (03) postes. La présidence de l'université et la bibliothèque centrale en comptent respectivement dix (10) et quatre(04).

Pour ce qui est des périphériques, ils se composent principalement d'imprimantes,

de télécopieurs et de scanners dont sont dotées les administrations et les salles de TP.

Pour la protection des équipements, de nombreux onduleurs sont déployés, notamment dans les salles de TP.

1.2 Les infrastructures réseaux

L'architecture du réseau actuel de l'UPB est très basique et se limite au bâtiment pédagogique où se localisent les salles de TP dont nous avons fait cas dans la partie précédente. Ces « micro-réseaux » sont en réalité constitués des ordinateurs de ces salles, reliés entre eux par du câble à paire torsadée de catégorie 5e pour permettre aux étudiants et aux enseignants de travailler en réseau ou de bénéficier de la connexion à Internet. La plus grande partie de ce réseau appartient à l'ESI qui dispose également de la connexion à Internet. Le matériel réseau qui est sous la responsabilité des administrateurs réseaux de la dite école se compose principalement de:

- 3 serveurs pour la sauvegarde des TP
- 1 serveur pour la connexion Internet
- 1 modem
- des onduleurs
- un router Cisco 2600 (router + firewall)
- des switch/hub pour connecter les ordinateurs du LAN (Réseaux local)
- deux bornes d'accès Wi-Fi pour la connexion sans fil au LAN

On note l'extension du réseau filaire de l'ESI par un réseau sans fil grâce aux points d'accès (AP) Wi-Fi ci-dessus cités. Mais ce réseau sans fil ne s'étend qu'à une cinquantaine de mètres environ autour du bâtiment pédagogique.

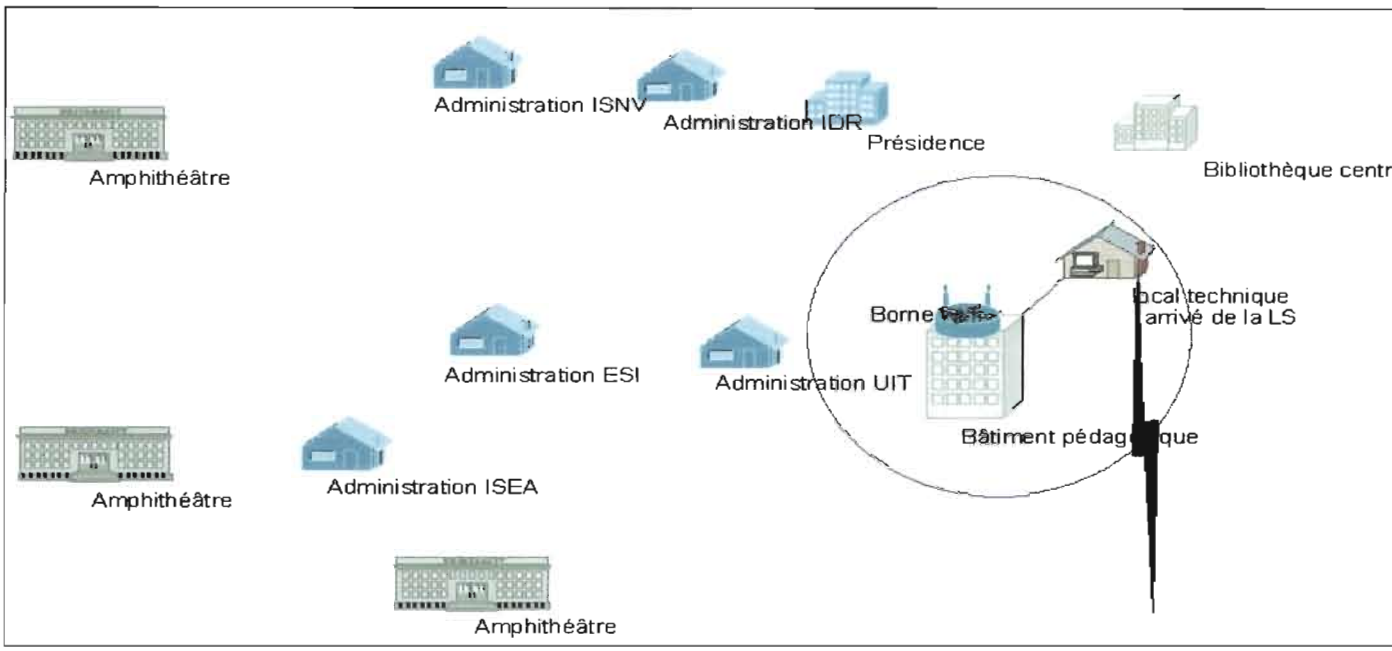


Figure I.1: Architecture du réseau actuel

Le cercle bleu sur la figure ci-dessus représente la couverture du réseau actuel.

2) Les logiciels et les services réseaux

Le fonctionnement du matériel informatique utilisé à l'UPB est assuré par une gamme variée de programmes informatiques allant des systèmes d'exploitation aux programmes d'applications diverses.

2.1 Les systèmes d'exploitation

Parmi les systèmes d'exploitation utilisés, celui qu'on rencontre le plus fréquemment sur des ordinateurs de l'université est MicroSoft Windows XP, notamment l'édition professionnelle avec le Service Pack2 (SP2). C'est celui qu'on trouve sur la quasi-totalité des ordinateurs de l'administration et dans la plupart des salles de TP. Cependant, dans les salles de TP de l'ESI on note également la présence de système Linux dans des distributions variées dont Mandriva, Debian, RedHat et Ubuntu.

A tout cela il faut ajouter le système d'exploitation serveur de MicroSoft, Windows Server 2000 utilisé sur les serveurs de l'ESI et celui de l'UIT. D'autres systèmes sont ponctuellement installés par les étudiants de l'ESI dans les salles de TP quant ils en ont besoin pour les étudier; par exemple, la version 2003 de Windows Serveur.

2.2 Les logiciels d'application

Pour ce qui est des logiciels d'application, ils sont tout aussi variées, sinon plus que les systèmes d'exploitation. Vous trouverez sur la plupart des postes la suite bureautique de MicroSoft, Office 2003 très souvent installé avec Windows XP SP2. Vous rencontrerez divers autres types de programmes dépendant du ou des utilisateurs des ordinateurs sur lesquels ils sont installés. Ainsi, dans les salles de TP de l'ESI, on trouve principalement des environnements de développement supportant des langages variés. Par exemple: Turbo C et C++, Dev C et C++, java NetBeans, Visual Basic, Lisp.

Nous avons également des applications permettant de lire des cours sous divers formats, notamment le format PDF (Acrobat Reader).

Sur certains postes, les étudiants installent de façon ponctuelle, des logiciels d'applications réseaux notamment des Logiciels de types serveurs.

Dans les autres salles de TP notamment à l'IUT, on rencontre en plus des outils de programmation, des outils d'instrumentation virtuels (Ladder) et des logiciels de gestion ou de comptabilité.

Sur certains postes des administrations on rencontre également des logiciels de comptabilité.

A tout cela il faut ajouter les antivirus utilisés sur la plupart des postes même si leur mise à jour n'est pas régulièrement faite sur tous les postes.

2.3 Les services réseaux

Le réseau de l'UPB, bien que encore très basique met déjà quelques services à la disposition de ses utilisateurs qui sont principalement les étudiants de l'ESI et aussi ceux des autres instituts pour certains services (Internet, partage de fichiers).

2.3.1 La connexion Internet

Une LS (liaison spécialisée) souscrite chez l'opérateur ONATEL S.A permet à l'UPB d'être connectée au réseau global (Internet). Grâce à cette connexion, l'UPB dispose d'un débit de 256Kbs qui devrait être permanent même si ce n'est pas le cas à cause de problèmes matériels fréquents (panne de MODEM par exemple). C'est le réseau de l'ESI qui reçoit cette connexion mise à la disposition des étudiants et

enseignants dans une de ses salles TP dotée d'un peu moins d'une dizaine de machines, située au deuxième étage du bâtiment pédagogique. Grâce à l'extension de son LAN par le Wi-Fi, la connexion Internet est également accessible aux alentours du bâtiment pédagogique pour les usagers d'ordinateurs portables équipés d'une carte Wi-Fi. Au premier étage du même bâtiment, une salle de l'IUT dispose également d'une connexion Internet .

2.3.2 Les sites Web

Trois sites WEB sont déjà réalisés pour l'UPB. Ce sont: celui de l'UPB, celui de l'ESI et celui du DEA à l'ESI. Tous ces sites sont hébergés sur les serveurs de l'AUF Burkina mais ne font pas l'objet de mises à jour depuis qu'ils sont mis en ligne.

Des associations de l'ESI telles ASSOC-LINUX (Association pour la promotion des logiciels libres et Linux) et l'ADESI (Association des Développeurs de l'ESI) ont également des sites à leurs noms qui sont hébergés chez des hébergeurs gratuits.

2.3.3 La gestion des comptes d'utilisateurs à l'ESI et à l'IUT

A l'ESI et à l'UIT, les différentes classes disposent de comptes d'utilisateurs avec des droits d'accès adéquats sur les postes des salles de TP. En plus les comptes d'utilisateurs des étudiants regroupés par binômes sont gérés de façon centralisée grâce aux deux serveurs dotés de Windows Server 2000. Ces deux serveurs permettent la sauvegarde des projets des étudiants.

Nous voici donc au terme de notre étude sur les moyens en TIC qui sont déjà mis en oeuvre à l'UPB. Nous pouvons d'ores et déjà remarquer que , bien que notre université ne soit pas totalement dépourvue d'un système informatique, celui-ci est loin de convenir pour toute l'UPB.

II. CRITIQUE DE L'EXISTANT ET PERSPECTIVES

Il convient que nous étudions le contexte de notre université pour faire ressortir ses besoins en TIC, puis que nous examinions en quoi son système actuel n'est pas à la hauteur de ses objectifs avant de proposer une solution plus adaptée et plus évolutive.

1) Pourquoi l'UPB a-t-elle besoin des TIC?

L' Université Polytechnique de Bobo-Dioulasso regroupe des instituts et écoles de formation dans des domaines scientifiques divers nécessitant l'utilisation des TIC. Elle devrait donc être dotée d'un système informatique plus évolué. Et cela d'autant plus que de nos jours les TIC tendent à intégrer tous les domaines de formation, scientifiques ou non.

En plus, l'UPB s'étend sur une très grande superficie à cause du grand nombre de ses bâtiments et de leur dispersion, ce qui nécessite de grands déplacements pour les échanges notamment de données mais aussi d'informations entre les membres des différentes administrations. Hormis donc les conditions que le système informatique doit remplir pour faciliter les activités pédagogiques, il faudrait également qu'il puisse doter les administrations d'un système de communication efficace et que toutes les administrations puissent partager des ressources grâce au réseau.

Enfin comme toute université digne de ce nom et de surcroît ayant une école de formation en informatique, l'UPB devrait oeuvrer à mettre régulièrement à jour son site Web et à l'héberger en son sein.

2) Les limites du système actuel

Il est indéniable que le système informatique actuel de l'UPB est loin de répondre aux besoins de communication et de facilitation des enseignements auquels elle aspire.

En effet, ce système ne joue qu'un rôle pédagogique sans aucun apport pour la facilitation des taches administratives. D'ailleurs il ne s'étend physiquement qu'au

bâtiment pédagogique qui n'est pourtant qu'un des édifices de l'institution. Et même là où il y a des réseaux, ce sont des micro-réseaux et ils sont isolés les uns des autres, limitant les possibilités de communication, d'échange et de partage des ressources. Aussi, la connexion à Internet n'est pas accessible à tous les instituts de l'université ni à son service administratif. En effet, elle n'est disponible que dans une salle de TP de l'ESI et à l'IUT.

Les administrateurs du réseau disposent de locaux techniques (au nombre de deux) d'où ils effectuent leurs taches quotidiennes mais ceux-ci ne répondent pas aux normes de mise en place ni de sécurité. Il n'est pas étonnant que le réseau soit hors service après une pluie.

En vue de combler ces lacunes, nous avons élaboré des solutions que nous allons exposer par la suite, en les justifiant.

3) Proposition de solutions

Il découle des analyses menées plus haut que l'UPB doit être dotée d'un système informatique qui couvre non seulement les installations de tous ses domaines de formation mais aussi les bâtiments administratifs. Il lui faudrait donc un Intranet subdivisé en trois parties du point de vue logique:

- une partie qui serait utilisée dans un but pédagogique et qui serait commune à l'ensemble de ses structures de formation.
- Une autre partie serait utilisée par les administrations dans le but de communiquer entre elles.
- La dernière partie permettrait à l'université de communiquer avec l'extérieur. Elle permettra non seulement à des utilisateurs du réseau interne de pouvoir en utiliser les ressources même étant à l'extérieur, mais aussi à des utilisateurs extérieurs à l'université de bénéficier d'informations ou de ressources auxquelles ils ont droit.

Mais toutes ces parties de notre Intranet bien qu'étant utilisées dans des buts différents seront gérées par un système central qui assurera le contrôle de tout accès à des ressources du réseau.

Dans la suite de notre travail, nous allons faire une étude détaillée des solutions possibles en justifiant celles que nous retiendrons et en mettant cette fois l'accent sur l'aspect technique.

TROISIEME PARTIE : DE LA MISE EN PLACE DES SOLUTIONS PROPOSEES

L'UPB est une structure de formation qui possède un pôle administratif et un pôle pédagogique. Les TIC occupent une place aussi importante dans la formation académique que dans l'administration. C'est ainsi que pour une bonne organisation du système informatique et pour des questions de confidentialité des données, nous proposons de subdiviser le réseau global en des sous réseaux. La subdivision nous donne, *le réseau des administrations*, *le réseau pédagogique* et la *zone démilitarisée*. Nous optons en dépit de cette subdivision de mettre en place un système de gestion centralisée des ressources informatiques par le protocole d'annuaire LDAP.

Dans la suite de notre travail, nous allons dans un premier temps définir l'organisation des infrastructures du système à mettre en place. Ensuite nous étudierons les services répondant aux besoins en TIC de l'université en vue de leur mise en oeuvre. Nous finirons par un bilan général qui nous permettra d'une part de donner un aperçu des services déjà déployés et d'autre part d'émettre des perspectives pour une réalisation effective de l'ensemble du projet.

I. DÉFINITION DE L'ARCHITECTURE DU FUTUR SYSTÈME INFORMATIQUE

Il y a deux niveaux d'abstraction dans le domaine des réseaux; *le niveau physique* et *le niveau logique*. Ainsi en parlant d'architecture réseau nous devons tenir compte de l'organisation logique et de celle physique. Pour mener à bien notre étude nous allons suivre une évolution axée sur la progression en couche du **modèle OSI**. Nous concevrons donc notre réseau par l'utilisation des équipements, protocoles et services selon les niveaux du modèle OSI.

1) Architecture physique

La topologie physique couvre les deux premiers niveaux du modèle OSI.

- **Niveau 1**: la couche physique, gère les connexions matérielles, définit la façon dont les données sont converties en signaux numériques et aborde les supports

de communication (media).

- **Niveau 2**: la couche liaison de données, définit l'interface avec la carte réseau. C'est le domaine d'utilisation des hubs, switch, ...

Nous allons concevoir l'architecture physique des infrastructures informatiques de l'UPB. Pour ce faire, nous étudierons les équipements réseaux et leur interconnexion, relevant des deux premières couches du modèle OSI.

1.1 Les médias utilisés au sein des bâtiments

Pour l'interconnexion des équipements dans chaque bâtiment nous proposons l'utilisation combinée de deux types de supports: Les supports filaire et sans fil.

1.1.1 Les supports filaires

A l'intérieur de chaque bâtiment nous choisissons le câble à paire torsadée catégorie 5e avec des prises murales. Les caractéristiques du câble à paire torsadée **catégorie 5e** sont les suivantes:

Dans les câbles à paire torsadée circulent des signaux électriques. La catégorie 5e (e pour *enhanced*) est un type de câble permettant une bande passante de 100 Mhz (apparu dans la norme TIA/EIA-568A-5).

Nous pouvons utiliser d'autres catégories, mais celles-ci seront difficilement trouvables sur la place du marché:

Catégorie 6 / classe E

La catégorie 6 est un type de câble permettant une bande passante de 250 Mhz et plus (norme ANSI/TIA/EIA-568-B.2-1 et ISO/CEI 11801 ed.2).

Catégorie 6a / classe Ea

Actuellement à l'état de brouillon, la future norme 6a s'oriente vers une extension de la catégorie 6 pour une bande passante de 500 Mhz (norme ANSI/TIA/EIA-568-B. 2-10).

Catégorie 7 / classe F

La catégorie 7 a une bande passante de 600 Mhz

Catégorie 7a / classe Fa

La catégorie 7a a une bande passante de 1 Ghz et est en cours d'étude.

1.1.2 Les supports sans fil

La transmission des données est assurée ici par les ondes radio. Grâce à la technologie, de plus en plus, les ordinateurs portables sont équipés d'antennes Wi-Fi. Pour faciliter la connexion de ces postes au réseau et rendre ce dernier flexible, nous proposons la disposition de points d'accès Wi-Fi(**AP** **A**ccess **P**oint) au niveau du bâtiment pédagogique. Le Wi-Fi permettra de relier des ordinateurs portables, des machines de bureau, des assistants personnels (PDA) ou même des périphériques à une liaison haut débit (de 11 Mbit/s en 802.11b à 54 Mbit/s en 802.11a/g et 540 Mbit/s pour 802.11n) sur un rayon de plusieurs dizaines de mètres en intérieur.

1.2 Les médias utilisés pour l'interconnexion des bâtiments

Plusieurs solutions sont envisageables pour l'interconnexion des édifices de l'UPB. Cependant, nous nous limiterons dans notre étude à la présentation d'une technologie de chaque type de média (filaire et sans fil) à l'issue de laquelle nous opérerons un choix.

1.2.1 Les supports sans fil

La BLR, acronyme de **B**oucle **L**ocal **R**adio est une technologie normalisée sous la référence IEEE 802.16. Elle est une technologie de connexion sans fil, fixe et bidirectionnelle.

- sans fil car elle utilise les ondes radio comme moyen de transmission ;
- fixe car le récepteur (l'antenne) doit être fixe, il ne peut être mobile ;
- bidirectionnelle parce que la liaison de communication se fait dans les deux sens.

La bande de fréquence de la BLR est comprise entre 3,5 et 26 GHz, avec une zone de couverture maximale de 10 km. Le débit maximal offert par la BLR de nos jours est de 8Mb/s.

LES AVANTAGES

Les avantages à ce niveau sont :

- Facilité de mise en œuvre (pas besoin de travaux de génie civil) ;
- Possibilité d'investissement progressif en fonction de la demande ;
- Faible coût de déploiement par rapport au réseau filaire ;
- Débit élevé.

LES FAIBLESSES

Les faiblesses rencontrées au niveau de la BLR sont :

- Zone de couverture limitée ;
- Obligation de vue directe entre les antennes (LOS : Line of Sight) ;
- Sensibilité aux conditions météorologiques.

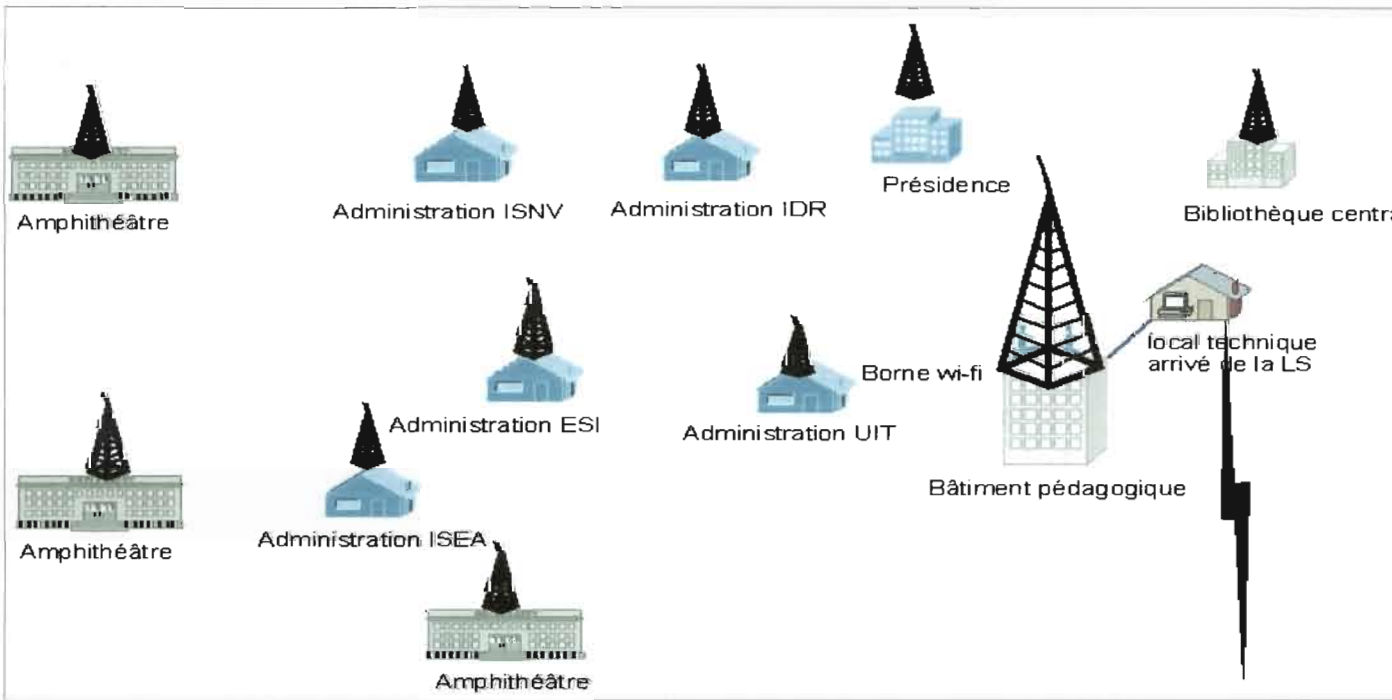


Figure I.1: Interconnexion avec la BLR

1.2.2 Les supports filaires

La fibre optique est un support physique permettant la transmission de données à haut débit sous forme d'impulsions lumineuses modulées. Il existe deux catégories de fibre optique: la fibre optique monomode qui permet d'atteindre un débit de 100 Gb/s et la fibre optique multimode dont le débit dépasse 500 Mb/s.

LES AVANTAGES

Les avantages de la fibre optique sont :

- Débit très élevé ;
- Transmission longue distance ;
- Sécurité élevée ;
- Perte de signal sur une grande distance bien plus faible que lors d'une transmission électrique dans un conducteur métallique ;
- Faible poids ;
- insensibilité aux interférences extérieures (proximité d'un câble à haute tension par exemple) ;
- pas d'échauffement (à haute fréquence le cuivre chauffe, il faut le refroidir pour obtenir des débits élevés).

LES INCONVENIENTS

Comme inconvénients, nous avons :

- Coût de déploiement élevé (prix du mètre et installation nécessitant des spécialistes dans le domaine) ;
- Maintenance difficile.

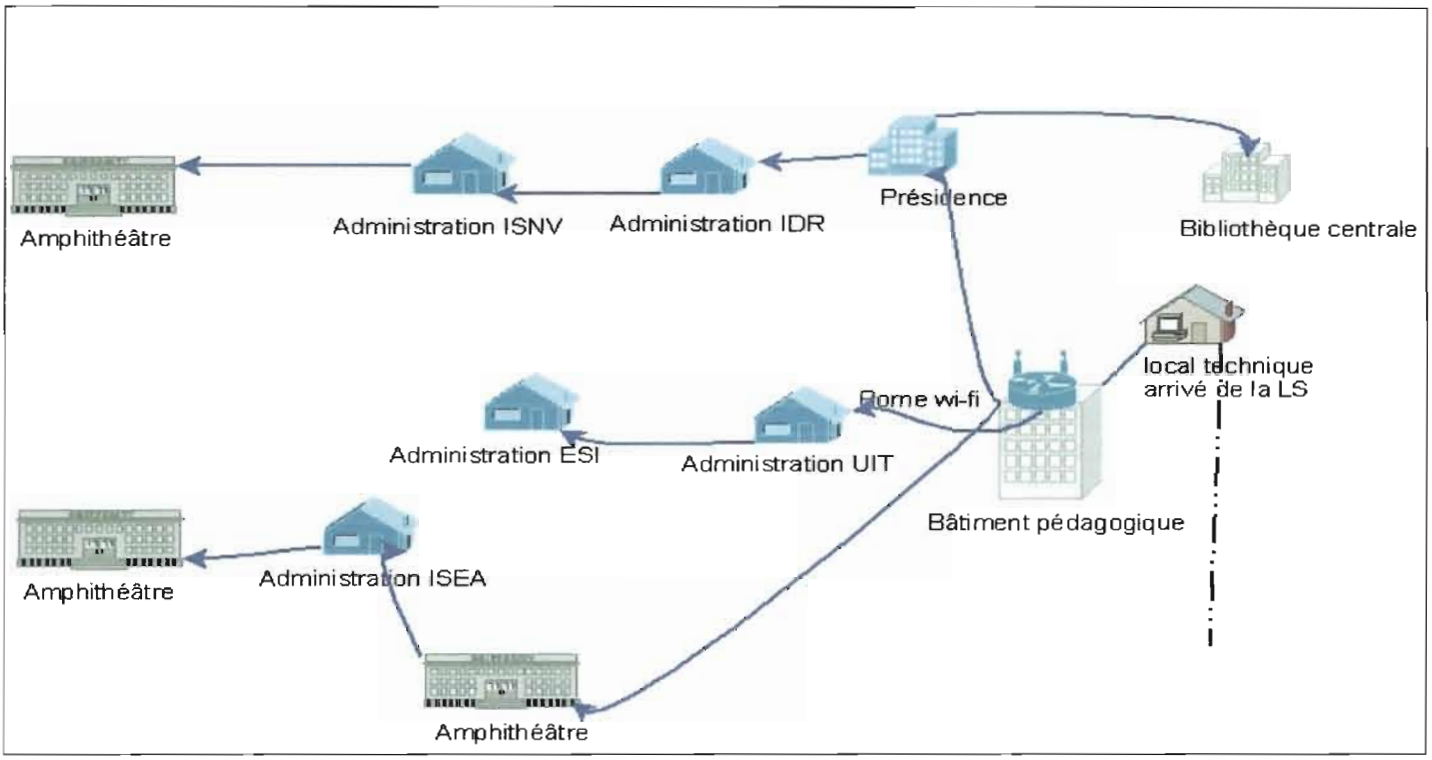


Figure I.2 : Interconnexion avec la fibre optique

En faisant une comparaison de ces deux technologies nous constatons que toutes les deux présentent de nombreux avantages et conviendraient pour l'interconnexion des bâtiments de l'UPB. Cependant nous pensons que la solution de la fibre optique serait plus intéressante à long terme malgré son coût de déploiement. Elle permet des débits beaucoup plus intéressants que la BLR. De plus, l'université est dans un environnement qui ne serait pas très favorable à une transmission par ondes radio à cause des nombreux obstacles(les arbres, les bâtiments, etc).

1.3 Les équipements réseaux et leur disposition

À ce niveau nous développons la topologie LAN de couche 2; c'est à dire l'ajout d'équipements de la couche 2: les commutateurs. Au niveau de chaque bâtiment d'administration nous placerons des commutateurs (switch) pour interconnecter les postes. Les commutateurs seront à leur tour, tous connectés à un commutateur central au sein du local technique.

1.4 Précautions à prendre et normes à suivre, dans la mise en place

Nous allons nous référer à la norme ANSI/TIA/EIA-569-A, relative aux espaces et

aux voies de télécommunication qui sont: câblage horizontal, câblage backbone, poste de travail, armoire de câblage, salle du matériel, salle des terminaux principaux.

➧ *Choix du local technique*

il s'agit de l'endroit où la plupart des câbles et des équipements de réseau seront installés. Les critères de choix de local technique sont les suivants:

- ***Taille:*** Un local technique doit être suffisamment grand pour pouvoir loger tous les équipements et le câblage nécessaires au réseau. De plus, un espace supplémentaire doit être prévu pour la croissance future du réseau. La norme TIA/EIA-569 stipule que chaque étage doit avoir au moins un local technique et qu'un local technique supplémentaire doit être installé tous les 1 000 mètres^2, lorsque la surface de l'étage desservi est supérieure à 1 000 mètres^2 ou que la distance du câblage horizontal est supérieure à 90 mètres.
- ***Environnement:*** Tout emplacement sélectionné pour un local technique doit répondre à certaines conditions d'environnement incluant entre autres l'alimentation électrique, la ventilation et la climatisation. De plus, seules les personnes autorisées ont accès au local qui doit être conforme à toutes les réglementations en vigueur dans les domaines de la sécurité et de la construction. Le local technique doit être conforme aux règles applicables aux éléments suivants :
  - **la température et l'humidité;** Le système de ventilation et de climatisation du local technique doit maintenir une température ambiante à environ 21 °C lorsque les équipements du réseau local fonctionnent. Aucune canalisation d'eau ou de vapeur ne doit passer au-dessus du local ou à l'intérieur de celui-ci, à l'exception d'un système de gicleurs que peuvent exiger la réglementation locale de prévention des incendies. L'humidité relative doit être maintenue à un niveau compris entre 30 % et 50 %. Si ces normes ne sont pas respectées, les fils de cuivre des câbles à paires torsadées non blindées ou blindées peuvent être détériorés par la corrosion, ce qui dégraderait les performances du réseau.
  - **l'accès au local et à l'équipement;** La porte du local technique doit avoir

au moins 90 cm de largeur et doit s'ouvrir vers l'extérieur pour permettre aux personnes de sortir facilement du local. Le verrou doit se trouver à l'extérieur de la porte, mais toute personne se trouvant à l'intérieur du local doit pouvoir sortir à tout moment.

Pour remplir ces différentes conditions nous proposons l'aménagement d'une salle au sein du bâtiment pédagogique qui servira de local technique.

- ***Les câbles et supports.***

L'accès aux câbles et leur support. Les câbles hors des bâtiments doivent être couverts et enterrés et ceux dans les bâtiments doivent être mis dans des goulottes. Enfin, toute ouverture dans les murs ou le plafond permettant au conduit ou au mandrin de pénétrer dans le local doit être scellée à l'aide d'un matériau ignifuge conforme à toutes les normes applicables.

- ***L'accès au sans fil***

A l'instar des autres équipements d'interconnexion tels que les commutateurs et les routeurs, les points d'accès Wi-Fi devront être disposés dans des locaux dont l'accès fait l'objet d'un contrôle rigoureux.

2) Architecture logique

La topologie logique couvre le niveau trois du modèle OSI.

Niveau 3: la couche réseau, détermine les routes de transport et s'occupe du traitement et du transfert de messages: gère IP (Internet Protocol) et ICMP (Internet Control Message Protocol). C'est le domaine du routage, la subdivision logique du réseau.

Dans notre cas, la subdivision se fera en trois parties.

2.1 Le réseau des administrations

Nous entendons par ***« réseau des administrations »*** le réseau formé par l'ensemble des machines des administrations des différents instituts et écoles, la présidence, la bibliothèque centrale, les bibliothèques des instituts et écoles. Cela pour faciliter la communication entre les administratifs.

Ce réseau a pour adresse IP 192.168.1.0/24. Il faut un serveur central pour la sauvegarde des données et un serveur de relais en cas de problème.

2.2 Le réseau académique

Le réseau pédagogique est constitué par les différentes salles informatiques. Les salles de TP de l'UPB seront connectées entres elles avec une gestion centralisée des données sur deux serveurs pour les travaux pratiques. Pour une question de coût d'interconnexion il est souhaitable que les salles machines soient toutes dans le bâtiment pédagogique quand l'UPB prévoira d'augmenter le nombre de ses salles machines. Ainsi la centralisation des données permettra la mobilité des étudiants sur les machines. Ce réseau aura pour adresse IP 192.168.2.0/24

2.3 La zone démilitarisée (DMZ)

Un réseau sera créé pour héberger l'ensemble de nos services publics. Ce réseau est accessible aussi bien du réseau interne que de l'extérieur, sans pour autant risquer de compromettre la sécurité de l'intranet de l'UPB. On parle ainsi de « zone démilitarisée » (notée DMZ pour *DeMilitarized Zone*) pour désigner cette zone isolée hébergeant des applications mises à la disposition du public. La DMZ fait ainsi office de « zone tampon » entre le réseau à protéger et le réseau hostile. Les serveurs situés dans la DMZ sont appelés « **bastions** » en raison de leur position d'avant poste dans le réseau de l'entreprise.

Ce réseau a pour adresse IP 192.168.3.0/24.

2.4 Schéma de synthèse de l'architecture logique

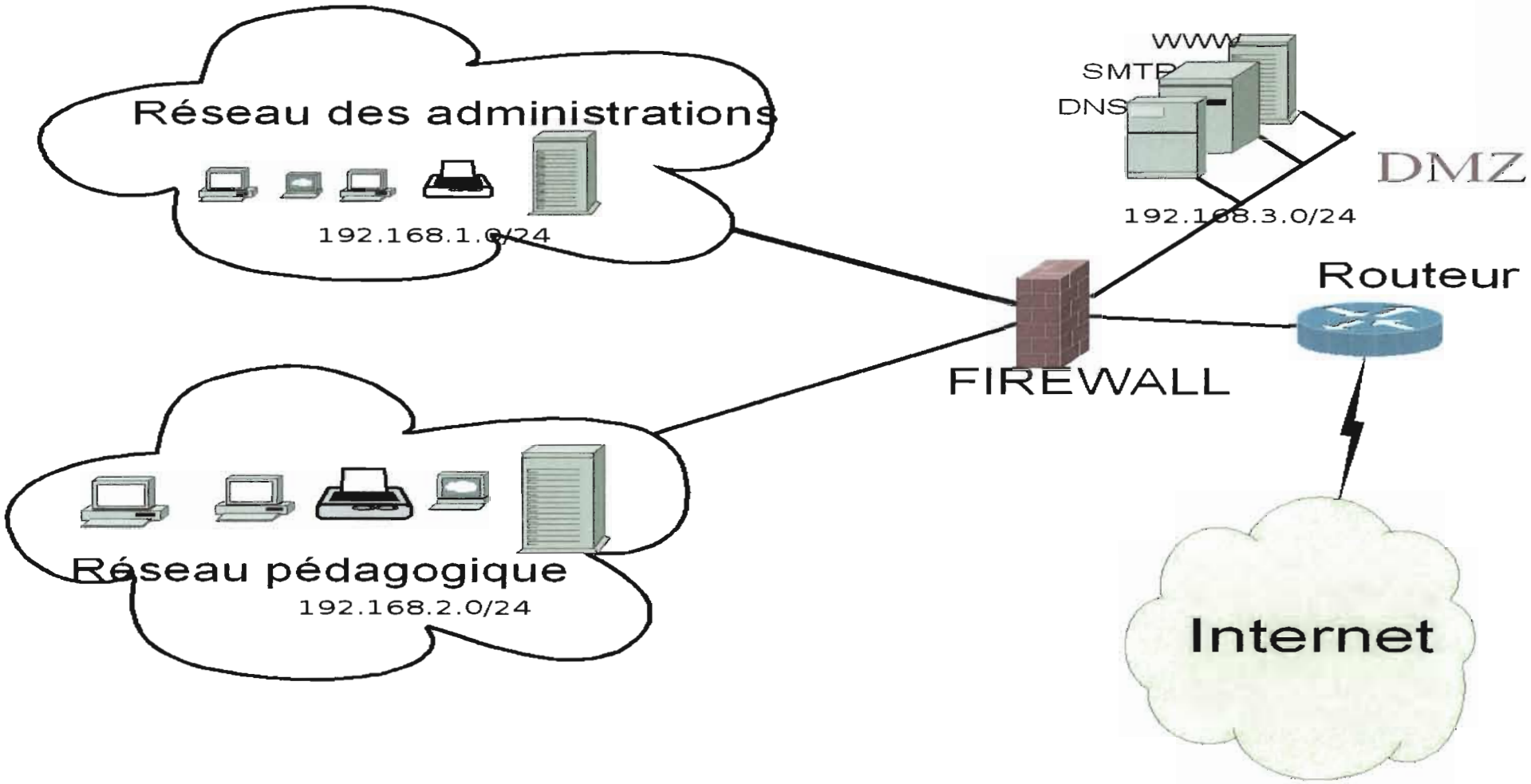


Figure I.3: Schéma de synthèse de l'architecture logique

En remontant les couches du modèle OSI et en passant en revu la disposition physique des équipements, leur interconnexion, et l'organisation logique, nous avons conçu le réseau de l'UBP. L'organisation logique détermine les règles de dialogue entre les équipements du réseau. Ceci étant, il convient de poursuivre notre progression axée sur le modèle OSI. Nous allons regrouper les quatre dernières couches (Transport, Session, Présentation et Application); c'est le domaine des services réseaux. Les services réseaux représentent l'ensemble des applications qui apportent un gain considérable dans les travaux quotidiens des utilisateurs du réseau.

II. ÉTUDE DES SERVICES À METTRE EN PLACE

La mise en place d'un système d'information doit être motivée par des besoins en communication. Or de nos jours aucune structure digne de ce nom ne peut se passer d'un système informatique quelque soit sa taille et son importance. L'efficacité d'un tel système dépend pourtant des services qui y sont déployés.

Les services réseaux sont des programmes interagissant directement avec les utilisateurs dans leur travail quotidien. C'est la partie émergée de l'iceberg "système d'information" que nous allons développer.

Pour mener à bien notre étude nous allons dans un premier temps définir le rôle de chaque service dans le système informatique de l'UPB; ensuite nous choisirons les applications et système d'exploitation adéquats pour leur mise en place. Enfin, deux autres parties seront consacrées respectivement à l'étude du système d'exploitation et des applications choisies.

1) Définition des services

Nous donnons ici une brève définition de chacun des services qui seront déployés en montrant leur utilité dans notre réseau.

➨ *Les services à l'entrée du réseau*

Le Firewall

Un pare-feu ou coupe-feu (firewall en anglais) est un système permettant de protéger un ordinateur ou un réseau d'ordinateurs des intrusions provenant d'un autre réseau (Internet le plus souvent). Il s'agit d'un système informatique situé à l'entrée du réseau pour le protéger des intrusions éventuelles provenant de l'extérieur.
Ainsi, en plus de la sécurité qui sera déployée dans la mise en oeuvre de chaque service nous mettrons en place un système de filtrage des accès du réseau de l'UPB.

➧ ***Les services publics (serveurs situés dans la DMZ)***

Web

Le serveur Web désigne:

- Un ordinateur sur lequel fonctionne un logiciel serveur HTTP.
- Le logiciel serveur HTTP lui-même.
- Un ensemble de serveurs permettant le fonctionnement d'applications Web.

A priori, un serveur Web permet de mettre des pages Web à la disposition des autres ordinateurs du réseau. Cependant, dans une implémentation plus avancée, il facilite l'utilisation de certains services (messagerie électronique par exemple). Il peut permettre également d'administrer les ressources serveurs grâce à l'interface Web (base de données par exemple). Dans la mise en place du serveur Web de l'UPB nous allons exploiter toutes les possibilités ci-dessus.

MESSAGERIE

Un serveur de messagerie électronique est un système qui permet l'échange de courriers électroniques (ou courriels) entre les utilisateurs. Pour l'UPB il faudra nécessairement un tel système vu que c'est l'un des services les plus utilisés des réseaux informatiques. Cela permettra de créer un compte pour tous les acteurs de l'université quel que soit leur domaine d'activité et de permettre ainsi une communication plus rapide, moins onéreuse et plus pratique au sein de l'université et même avec l'extérieur grâce à Internet.

DNS

Le Domain Name System (ou DNS, système de noms de domaine) est un système permettant d'établir une correspondance entre une adresse IP et un nom de domaine. Il en dérive qu'un serveur de résolution de noms ou serveur DNS (Domain Name Server) permet de faire une correspondance entre les adresses IP(utilisées par les ordinateurs d'un réseau TCP/IP pour communiquer) et les noms de machine (qui sont plus mnémoniques pour nous les êtres humains). Les ordinateurs du réseau pourront alors s'identifier par des adresses numériques qu'ils traitent plus facilement pendants que les utilisateurs humains retiendront les noms de machines qui sont plus significatifs. Le

système DNS est d'autant plus important dans un réseau qu'il est utilisé par d'autres services pour fonctionner correctement; c'est le cas par exemple du Web et de la messagerie. La preuve c'est qu'il est plus facile et plus commode de demander l'affichage de la page d'accueil du site Web de l'UPB en saisissant l'URL: www.univ-bobo.bf plutôt que 212.52.149.156. De même nous envoyons un message à l'adresse remi3@univ-bobo.bf et non à remi3@212.52.149.156. Pourtant c'est le système DNS qui fait la correspondance entre l'adresse 212.52.149.156 et le nom de domaine www.univ-bobo.bf.

FTP

Un serveur FTP (File Transfer Protocol) comme son nom l'indique permet le transfert de fichiers. Nous l'implémenterons d'une part pour faciliter la mise à jour des sites Web. D'autre part, il permettra le dépôt et la récupération à distance de fichiers dans des répertoires dédiés aux enseignants.

- ***les services de l'Intranet***

DHCP

Un serveur DHCP (*Dynamic Host Configuration Protocol*) qui utilise le protocole de même nom, a pour rôle d'attribuer des adresses IP à des ordinateurs d'un réseau ainsi que tous les paramètres de configuration tels que: serveur DNS, passerelle, nom du réseau, pour une durée déterminée.

L'administrateur du réseau est exempté de la configuration manuelle de chaque poste du réseau qui peut s'avérer très pénible pour un réseau d'une certaine taille. De plus il n'ya pas de risque que plusieurs postes aient la même adresse si leur attribution est gérée par DHCP.

Ainsi donc, un serveur DHCP facilitere la tâche d'administration du réseau.

Serveur de fichiers

Un serveur de fichiers, permet le partage des ressources entre les utilisateurs Nous entendons par ressources, les fichiers, les répertoires, les périphériques, etc. L'accès à ces ressources se fera par identification et authentification des utilisateurs. Dans notre l'implémention du serveur de fichiers nous laisserons le choix du système d'exploitation client aux utilisateurs pour accéder aux ressources partagées.

Annuaire

La gestion de l'information occupe une place très importante dans notre étude. La multiplicité des applications et des serveurs rend cette information difficile à maîtriser. Pour y remédier nous proposons la mise en place d'un annuaire électronique. Un annuaire électronique 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 multi-critères. 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 sera de servir d'entrepôt pour centraliser des informations et les rendre disponibles, via le réseau à des applications, des systèmes d'exploitation et des utilisateurs.

Notre service d'annuaire permettra la gestion aussi bien des ressources de l'Intranet que celles de la DMZ.

Le schéma de synthèse suivant montre la disposition des services; ils seront connectés à l'annuaire pour assurer leur gestion centralisée.

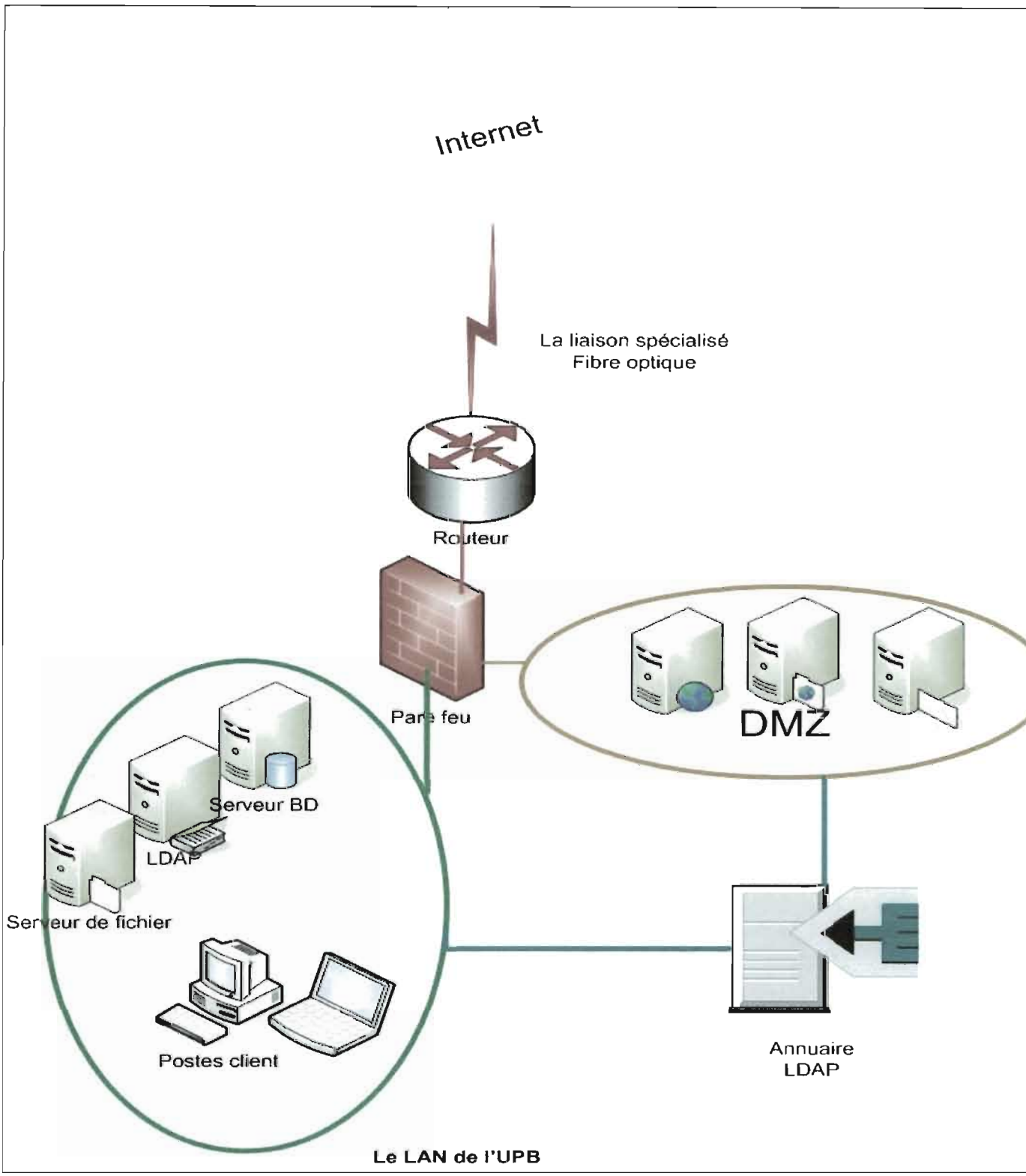


Figure II.1: schéma de synthèse de la disposition des services

Après cette brève définition des services réseaux, nous allons passer au choix des applications permettant de les implémenter. Cependant, pour que ces dernières puissent bien fonctionner, il est nécessaire qu'un système d'exploitation soit préalablement déployé pour assurer leur communication avec le matériel informatique.

2) Choix du système d'exploitation et des applications serveurs

La question de savoir quel système d'exploitation convient pour notre étude peut susciter de très vifs débats tant à cause du très grand nombre de systèmes existants que du fanatisme de certains utilisateurs vis-à-vis d'un système donné. Pour nous guider dans notre choix, nous allons dans un premier temps mener une étude comparative des systèmes d'exploitation serveurs les plus utilisés de nos jours. Ensuite nous effectuerons un choix en considérant les critères suivants: ***les compétences humaines disponibles, les différents services à mettre en place et le contexte de l'UPB.***

Tout d'abord, examinons les tableaux suivants qui présentent les caractéristiques générales et techniques des systèmes d'exploitation serveurs les plus utilisés.

Caractéristiques générales:

Tableau II.1: Caractéristiques générales des systèmes d'exploitation serveurs

Système d'exploitation	Créateur	Première version publique (date)	Ancêtre	Dernière version	Prix	Licence	Ordinateur cible
AIX	IBM	1986	System V release 3	5.3 (août 2004)	Fourni avec le matériel	Logiciel propriétaire	Serveur, Station de travail
freeBSD	Le projet FreeBSD	Décembre 1993	386BSD	6.2 (15 janvier 2007) et 5.5 (25 mai 2006)	Gratuit	Licence BSD	Serveur, Station de travail
GNU/Linux	Auteurs multiples	17 septembre 1991	Minix	Kernel 2.6.22.4 (21 août 2007), 2.4.35.1 (15 août 2007) et 2.2.26 (5 février 2004)	Gratuit	Normalement GNU GPL (Copyleft)	Serveur, Station de travail, Ordinateur de bureau
Mac OS X	Apple Computer	Mars 2001	UNIX MachBSD, NeXTSTEP, Mac OS	10.4.10 « *Tiger* » (20/06/2007), 10.5.0 « *Leopard* » (26/10/2007)	129$, Familial (5 postes) 199$, Mac OS X Serveur 10 clients 499$, Mac OS X Serveur illimité 999$, Étudiant 69$	Logiciel propriétaire, en partie APSL, GPL, et autres.	Ordinateur personnel, Station de Travail, Serveur
NetBSD	Le projet NetBSD	Mai 1993	386BSD	3.1 (9novembre 2006)	Gratuit	Licence BSD	Embarqué, Ordinateur de Bureau, Serveur
HP-UX	Hewlett-Packard (HP)	1983	Unix	11.23 "11i v2" (Décembre 2005)	400$	Logiciel propriétaire	Serveur, Station de travail

Système d'exploitation	Créateur	Première version publique (date)	Ancêtre	Dernière version	Prix	Licence	Ordinateur cible
NetWare	Novel	1985	S-Net	6.5 SP4 (Septembre 2005)	184$	Logiciel propriétaire	Serveur
OpenBSD	Le projet OpenBSD	Octobre 1995	NetBSD	1.0 4.1 (1er mai 2007)	Gratuit	Licence BSD	Serveur, Station de Travail, Embarqué
OpenVMS	DEC (HP à l'heure actuel)	Février 1978	RSX-11M	8.2-1 (septembre 2005)	Gratuit pour usage non-commercial	Logiciel propriétaire	Serveur
OS/2	IBM/Microsoft	Décembre 1987	MS-DOS	4..52 (décembre 2001)	300$	Logiciel propriétaire	Serveur, Ordinateur personnel
Plan 9	Bell Labs	1993	Unix	Quatrième édition	Gratuit	LPL	Station de Travail, Serveur, Embarqué, HPC
Solaris	Sun Microsystems	Juillet 1992	SunOS	10(1ér février 2005)	Gratuit	CDDL	Station de travail, Serveur
Windows Server 2003	Microsft	Avril 2003	Windows 2000	5.2 SP1 (30 mars 2005)	999$/5 clients	Logiciel propriétaire	Serveur

Caractéristiques techniques:

Tableau II.2: Caractéristiques techniques des systèmes d'exploitation serveurs

Système d'exploitation	*Architectures possibles*	*Système de fichiers possible*	*Type de noyau*	*Environnement graphique integré*	*Paquetages*	*Logiciel de mise à jour*	*APIs*
AIX	POWER, PowerPC JFS,	JFS2, ISO 9660, UDF, NFS, SMBFS, GPFS	Micro-noyau	Non	installp, RPM	Service Update Management Assistant (SUMA)	SysV, POSIX
FreeBSD	Intel IA32 (x86), AMD64, PC98, SPARC, autres	UFS2, ext2, FAT, ISO 9660, UDF, NFS, autres	Monolithique avec des modules	Non	ports tree, packages	par source (CVSup), freebsdupdate	BSD, POSIX
HP-UX	PA-RISC,IA-64	CFS, HFS, ISO 9660, NFS, SMBFS, UDF, VxFS	Monolithique avec des modules	Non	swinstall	???	SysV, POSIX
GNU/Linux	Presque toutes	Presque tous	Monolithique avec des modules	Non (sauf avec X Window, très répandu)	selon la distribution	selon la distribution	POSIX
Mac OS X	PowerPC, Intel IA32 (s86)	HFS+ (defaut), UFS, AFP, ISO 9660, FAT, UDF, NFS, SMBFS, NTFS (lecture seulement)	Hybride	Oui	OS X Installer	Software Update	Carbon, Cocoa, BSD/POSIX, X11 (depuis la

Système d'exploitation	*Architectures possibles*	*Système de fichiers possible*	*Type de noyau*	*Environnement graphique integré*	*Paquetages*	*Logiciel de mise à jour*	*APIs*
							10.3)
NetBSD	Intel IA32 (x86), 68k, Alpha, AMD64, PowerPC, SPARC, playstation2, dreamcast(60 plateformes)	UFS, UFS2, ext2, FAT, ISO 9660, NFS, LFS, autres	Monolithique avec des modules	Non	pkgsrc	par source (CVS, CVSup, rsync) ou binaire (utilisant sysinst)	BSD POSIX
NetWare	Intel IA32 (x86)	NSS, NWFS, FAT, NFS, AFP, UDF, ISO 9660	Hybride	Non	NWCONFIG.NLM, RPM	mise à jour binaire, Red Carpet	Propriétaire
OpenBSD	Intel IA32 (x86), 68k, Alpha, AMD64, SPARC, VAX,	UFS, ext2, FAT, ISO 9660, NFS, quelques autres autres	Monolithique avec des modules	Non	ports tree, packages	apr source	BSD, POSIX
OpenVMS	VAX, Alpha, IA-64	Files-11, ISO 9660, NFS	Monolithique avec des modules	Non	PCSI, VMSINSTAL	-	Unix-like
OS/2	Intel IA32 (x86)	HPFS, JFS, FAT, ISO 9660, UDF, NFS	Monolithique avec des modules	Oui	Via Install et autres	-	Propriétaire
Plan 9	Intel IA32 (x86), Alpha, MIPS, PowerPC, SPARC, autres 0	fossil/venti, 9P2000, kfs, ext2, FAT, ISO 966	Monolithique avec des modules	Oui	-	replica	Unix-like (et optionellement POSIX)
Solaris	SPARC, SPARC64,	UFS, ZFS, ext2, FAT,	Monolithique avec	Non	SysV packages	Sun Update	SysV,

Système d'exploitation	*Architectures possibles*	*Système de fichiers possible*	*Type de noyau*	*Environnement graphique integré*	*Paquetages*	*Logiciel de mise à jour*	*APIs*
	AMD64, Intel IA32 (x86) (pkgadd)	ISO 9660, UDF, NFS, quelques autres	des modules			Connection	POSIX
Windows server 2003	Intel IA32 (x86), AMD64, IA-64	NTFS, FAT, ISO 9660, UDF	Hybride	Oui	MSI, installateurs personnalisés	Windows Update	Win32, Win64

Nous venons de faire le point sur les systèmes d'exploitation serveurs les plus utilisés. Nous pourrions déjà en choisir un en tenant compte des critères tels que les fonctionnalités, le coût, la licence et le type de matériel supporté par ces systèmes. cependant pour optimiser notre choix, nous allons également prendre en compte des aspects tels que le contexte de l'UPB, les compétences humaines disponibles et surtout les services à mettre en place.

→ Le contexte de l'UPB

L'UPB en tant que structure de formation supérieure, a en son sein une École d'informatique qui est l'ESI, la première école informatique du BURKINA. En ce sens les idées, les décisions en TIC pour l'université passent par l'ESI. Aucune École ou institut qui se respecte dans le monde ne prônera l'utilisation d'une solution propriétaire à la place d'une solution libre si le choix est à faire. Solution propriétaire dans le sens où l'informaticien n'a pas le droit de comprendre le code source des applications qui tournent ou chercher à le modifier pour l'adapter aux besoins de l'entreprise. En plus de cela les licences propriétaires coûtent des millions. Ce qu'il nous faut c'est la solution du libre qui est même presque gratuit à des moments. La gratuité du libre n'est pas la seule raison qui nous pousse à le choisir, d'ailleurs qui dit libre ne dit pas forcement gratuit. *Le logiciel libre* se résume en ces termes: liberté d'exécution, liberté d'étude, liberté de modification et liberté de redistribution.

→ Compétences humaines.

La mise en place et le suivi d'un système informatique, nécessite un minimum de compétence. Cela est d'autant plus important que si on envisage d'utiliser les logiciels libres (GNU/Linux) auxquels on reconnaît quand même la non facilité d'utilisation. Il est donc primordial que nous possédions de bonnes connaissances du système. Les bases de ces compétences peuvent être acquises par des formations académiques continues. Cela n'est guère une inquiétude dans le cas de notre étude si nous considérons le fait que l'UPB compte parmi ses instituts et écoles, une structure de formation en

informatique qui intègre dans ses programmes de formations, des modules liés aux logiciels libres notamment GNU/Linux. De plus, pour le suivi de son système informatique existant, notre université dispose d'un personnel technique qui n'est pas étranger aux systèmes de la famille UNIX

Aussi, il est indispensable que nous apprécions le système d'exploitation choisi et que nous le pratiquions régulièrement.

En somme, l'inquiétude liée à la difficulté de déploiement et d'utilisation des systèmes UNIX pour la mise en place du système informatique de l'UPB ne serait pas justifiée.

→ Que voulons-nous faire?

Un autre critère dont la prise en compte doit être primordiale dans la mise en place d'un système informatique est l'application que l'on veut en faire. En effet, le meilleur système d'exploitation du monde (s'il existait) ne le serait pas forcément dans tous les domaines d'application.

Le choix du système d'exploitation se fait donc en tenant compte également de l'ensemble des services que celui-ci peut nous permettre de configurer. Pour notre étude, nous envisageons de mettre en place notamment:

- **un serveur d'annuaire,**
- **un serveur HTTP,**
- **un serveur FTP,**
- **un serveur de résolution de nom de domaine (DNS),**
- **un serveur de fichiers,**
- **un serveur de messagerie.**

Dans les lignes qui suivent, nous allons faire l'état sur les applications serveurs les plus utilisées en vue d'opérer un choix.

Serveur d'annuaire

Dans l'univers TCP/IP, les annuaires électroniques utilisent par excellence le protocole LDAP.

Voici une liste des principaux annuaires LDAP existant sur le marché :

- ✔ OpenLDAP

- Apache Directory Server
- Sun (One/Java) Directory Server
- Active Directory de Microsoft

Active Directory et Sun (One/Java) Directory Server étant des logiciels propriétaires, ils ne feront pas l'objet de notre choix. Le choix est donc à faire entre OpenLDAP et Apache Directory Server tous deux libres. Nous proposons l'utilisation de OpenLDAP qui est plus performant, plus sécurisé et plus évolutif.

Serveur HTTP

Netcraft est une entreprise spécialisée dans les technologies Internet; elle est surtout connue pour mener depuis 1995 des sondages automatisés d'Internet par nom de domaine à la recherche de serveurs HTTP, donc de sites Web. Elle publie mensuellement ses résultats qui sont régulièrement repris par les média informatiques.
Tableau statistique des serveurs HTTP les plus utilisés en Octobre 2007 d'après Netcraft.

Tableau II.3: statistique des serveurs HTTP les plus utilisés en Décembre 2007 d'après Netcraft.

Applications serveurs	***Nombre de postes serveurs***	***licence***	***pourcentage d'utilisation***
Apache	76,945,640	GPL	49.57%
Microsoft IIS	55,509,223	propriétaire	35.76%
Google GWS	8,558,256	propriétaire	5.51%
lighttpd	1,521,250	-	0.98%
Sun	588,997	propriétaire	0.38%
autres	12,089,939	-	7,8%

Le tableau ci-dessus nous montre que Apache est le serveur HTTP le plus implémenté dans le monde. En plus d'être sous licence GNU GPL (il est libre contrairement à ses deux concurrents directs qui sont propriétaires), il présente de nombreux atouts tels que sa conception modulaire, sa forte documentation, sa robustesse, la sécurité dont il est doté et son support des hôtes virtuels.
Nous pensons donc que c'est le serveur HTTP qui convient pour notre étude.

Serveur FTP

On trouve de nombreux serveur FTP pour Linux/Unix/BSD comme:
ftpd, glftpd, ProFTPd, Pure-FTPd, VsFTPd, Wu-ftpd, wzdftpd.
sous Windows on trouve:
warFTPD Server, File Zilla Server, Pure-FTPd, Typsoft FTP, Server, wzdftpd. Serv-U.
Mais ce sont tous des logiciels libres. En effet le protocole FTP est de Unix

Notre choix se porte sur vsFTPd, simple et très sécurisé. D'ailleurs, il a été développé dans l'optique de la meilleure sécurité possible afin de combler les innombrables failles de ses concurrents. Bien que très simple, il bénéficie de toutes les options habituelles des serveurs ftp classiques (ProFTPd, Pure-FTPd, ..).

Serveur de fichiers

Étant donné que nous avons un réseau hétérogène il faut permettre aux utilisateurs d'accéder aux données partagées quel que soit le système d'exploitation qu'ils utilisent. Pour les utilisateurs Linux nous configurerons le NFS (Network File System). En ce qui concerne les utilisateurs Windows nous avons le choix entre Samba (logiciel libre sous licence GPL) et Windows 2003 Server.
En la matière des études ont montré que Samba est trois (3) fois plus rapide d'accès que windows 2003 serveur. Nous n'avons donc pas l'embarras du choix à ce niveau vu que le plus performant est également gratuit.

Le firewall

Dans le domaine de la protection du réseau nous pourrions remplir des pages avec une liste des pare-feu. Mais parmi ceux-ci, iptables est celui qui offre le plus de flexibilité dans la configuration; c'est l'interface utilisateur de Netfilter qui est en fait un puissant outil réseau qui permet le déploiement d'une très bonne politique de sécurité.

serveur de messagerie

La liste des serveurs de messagerie est très longue.
Les principales solutions de messagerie propriétaires du marché sont :

- Lotus domino de IBM,
- Exchange server de Microsoft,

- Novell Groupwise de Novel,
- Oracle Collaboration Suite Email de Oracle,
- Mdaemon de Alt-N-Technologies.

Mais cela ne nous intéresse pas, car *« il y a moins cher et plus efficace »*.

Parmi les serveurs de messagerie en libre nous avons principalement: Exim, Sendmail, Qmail, Postfix. Le tableau suivant permet d'en faire une étude comparative.

Tableau II.4: Étude comparative des MTA en libre

caractéristiques	***Exim***	***Sendmail***	***Qmail***	***Postfix***
Installation	Moyenne	facile	Difficile	Moyenne
configuration	Moyenne	Difficile	Facile	Facile
performances	Moyennes	Faibles	Bonnes	Bonnes
documentation	Beaucoup	Beaucoup	Beaucoup mais souvent confuse	Assez
Licence	GPL	GPL	Un peu confuse. gratuit mais pas libre	IBM public license
Evolutivité	peu évolutif	peu évolutif	très peu évolutif	évolutif
sécurité	Faible	Faible	Très haute	Haute

En scrutant le tableau de comparaison, nous constatons que Qmail et postfix arrivent en tête de course. Mais que pouvons-nous dire exactement de ces deux serveurs?

Qmail est disponible gratuitement, le code est accessible et vous avez l'autorisation de le redistribuer. En revanche, il est interdit de redistribuer une version de Qmail à laquelle une modification a été effectuée, sans l'accord préalable de l'auteur. Cette interdiction empêche Qmail d'être considéré comme un logiciel libre. Suite à cette interdiction, Qmail vieillit et des bogues apparaissent en même temps que les bibliothèques se mettent à jour. Ainsi de nombreux patches(correctifs) sont disponibles et il est presqu'obligatoire de les appliquer pour que le serveur fonctionne correctement.

Units) qui constituent deux branchements : "étudiant" et "ordinateur"(cf figure II.2), dans lesquels nous trouvons ensuite les entrées feuilles (éléments terminaux) de notre arbre : les étudiants et les ordinateurs. Chacune des entrées de notre arbre correspond à un type de données particulier, défini par une classe d'objet.

Règles de nommages

La RFC 2253 normalise l'écriture des DN et conseille de ne pas ajouter d'espaces autour du signe "=", ni à la fin du DN. Les espaces sont autorisés par contre pour les valeurs des entrées.

Ainsi, le DN suivant est correct :

"cn=Edem Kodjo,ou=étudiant,dc=univ-bobo,dc=bf"

Alors que celui-ci ne l'est pas :

"cn = Edem Kodjo, ou = étudiant, dc = univ-bobo, dc = bf"

Les majuscules seront ou non prises en compte en fonction du type d'attribut utilisé et de ses particularités.

Une représentation hiérarchique des données

LDAP organise les données de manière hiérarchique dans l'annuaire. Ceci signifie que toutes les informations découlent d'une seule et même "racine".

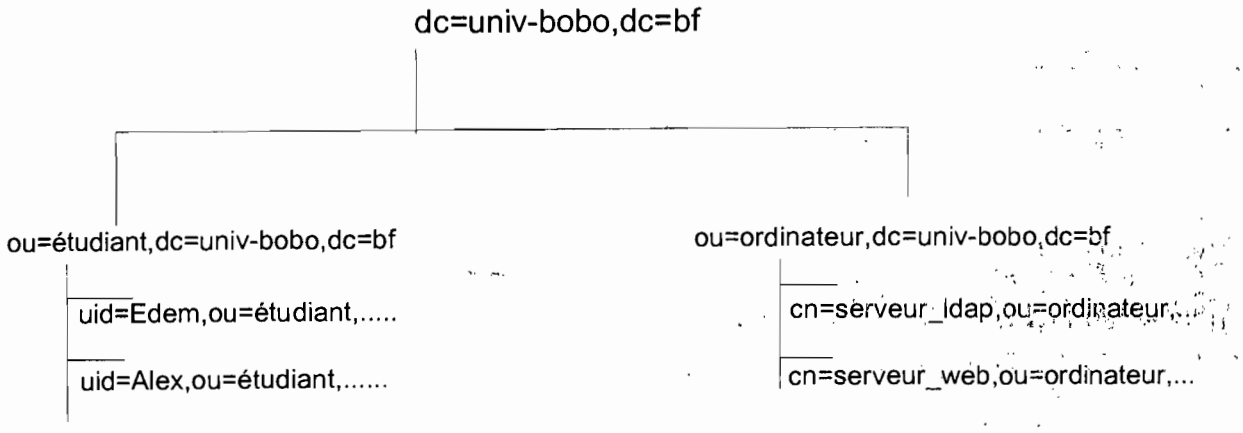


Figure II.2: Représentation hiérarchique des données avec LDAP

Cette arborescence est liée au nommage de chaque élément : un élément marque

son appartenance à l'élément supérieur en reprenant le nom de ce dernier, qu'il complète par le sien.

Ainsi, en étudiant simplement le nom de l'élément :

"cn=Edem,ou=étudiant,dc=univ-bobo,dc=bf"

il est possible de le situer dans la hiérarchie : il est situé sous l'élément "ou=étudiant" qui lui-

même est situé sous l'élément "dc=univ-bobo,dc=bf".

➧ Accéder à l'annuaire(Modèle fonctionnel)

Il existe plusieurs types d'opérations que l'on peut effectuer sur l'annuaire; voici les plus importantes :

- Rechercher une entrée suivant certains critères
- S'authentifier
- Ajouter une entrée
- Supprimer une entrée
- Modifier une entrée
- Renommer une entrée

Certaines de ces actions, notamment la recherche, nécessitent des outils particuliers pour nous faciliter l'accès à l'annuaire .

La base

La base est le DN à partir duquel nous allons agir. Pour une recherche, il s'agit du noeud à partir duquel est effectuée la recherche. Il peut s'agir de la racine de l'arbre pour une recherche

sur la totalité de l'arbre, par exemple "dc=univ-bobo,dc=bf".

La portée

La portée (scope) est le nombre de niveaux sur lesquels l'action va être effectuée. Il existe 3 niveaux différents :

- SUB : l'action est effectuée récursivement à partir de la base spécifiée sur la

totalité de l'arborescence.

- ONE : l'action est effectuée sur un seul niveau inférieur par rapport à la base spécifiée (les fils directs). Si l'on effectuait une recherche avec la portée ONE à partir de "dc=univ-bobo,dc=bf", nous pourrions trouver "ou=étudiant,dc=univ-bobo,dc=bf" et "ou=ordinateur,dc=univ-bobo,dc=bf".
- BASE : l'action est effectuée uniquement sur la base spécifiée. Une recherche sur ""dc=univ-bobo,dc=bf"" avec la portée BASE renverrait cette entrée uniquement.

Les filtres

Le troisième outil à notre disposition est le filtre. Un filtre va permettre d'effectuer des tests de correspondance lors d'une recherche. Il s'agit en quelque sorte du critère de la recherche.

Il existe 4 tests basiques, qui peuvent ensuite être combinés :

- Le test d'égalité : **X=Y**
- Le test d'infériorité : **X<=Y**
- Le test de supériorité : **X>=Y**
- Le test d'approximation : **X~=Y**

Les autres opérateurs (<, >) ou des tests plus complexes peuvent être mis en place par combinaison, il faut alors utiliser les parenthèses ( ) et l'un des opérateurs suivants :

- L'intersection (et) : **&**
- L'union (ou) : **|**
- La négation (non) : **!**

Un test d'infériorité stricte pourrait donner ceci : (**&**(X<=Y)(**!**(X=Y)))

On peut combiner plus de deux éléments : (**&**(X=Y)(Y=Z)(A=B)(B=C)(!(C=D)))

Ces filtres seront appliqués sur des attributs choisis pour sélectionner finement les données que nous voulons extraire de notre annuaire.

Les URLs LDAP

C'est une méthode concise et simplifiée pour interroger un annuaire LDAP. Il s'agit d'un format d'URL combinant toutes les notions vues ci-dessus. En une seule ligne, il est possible de spécifier tous les éléments de notre requête. Voici le format de cette URL (RFC 2255) :

ldap[s]://serveur[:port][/[base[?[attributs à afficher][?[portée][?[filtre][?extensions]]]]]]

L'exemple ci-dessous recherche tous les uid de notre arbre, à partir de la branche étudiant :

ldap://localhost:389/ou=étudiant,dc=univ-bobo,dc=bf?uid?sub

➧ Les données contenues dans l'annuaire (modèle d'information)

Le format LDIF

Les données contenues dans l'annuaire sont présentées dans un certain format : il s'agit du format **LDIF** (**L**DAP **D**ata **I**nterchange **F**ormat - RFC 2849). Dans ce format, chaque entrée constitue un paragraphe, et au sein de chaque paragraphe, chaque ligne constitue un attribut.

Les attributs

Un attribut est une valeur contenue dans une entrée. Une entrée peut bien entendu contenir plusieurs attributs. Prenons l'exemple de l'entrée LDAP complète d'un compte utilisateur POSIX :

```
dn: uid=Alex,ou=étudiant,dc=univ-bobo,dc=bf
objectClass: account
objectClass: posixAccount
cn: Alex
uid: Alex
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/Alex
userPassword:: e0NSWVBUfWJjT29IUk5SbG1HbC4=
loginShell: /bin/sh
gecos: Alex
description: Élève Ingénieur
```

Ceci correspond à une entrée complète, extraite par une interrogation de l'annuaire. Le format affiché est le format LDIF.

Un attribut est séparé de sa valeur par ":". Suivant son type, un attribut peut avoir plusieurs valeurs : dans ce cas, il est dit "multi-valué" et apparaît sur plusieurs lignes avec des valeurs différentes. Nous pouvons observer ici des attributs nommés "dn", "objectClass", "cn", "uid", ...

L'attribut "dn" qui est indiqué en première ligne est le nom unique de notre entrée dans l'arbre dont nous avons parlé précédemment. Il constitue un attribut à part entière dans notre entrée. Il est composé du dn de l'entrée supérieure, ainsi que du rdn. Le rdn est défini par un ou plusieurs attributs de l'entrée (dans ce cas séparés par un +). Il est conseillé, pour une entrée de type posixAccount, d'utiliser les attributs uid ou cn (cf. RFC 2307). Nous avons choisi ici uid=Alex.

Nous n'allons pas étudier chacun des attributs présents ici, cependant, nous allons mettre l'accent sur l'un des attributs les plus importants, il s'agit de la classe d'objet, ou "objectClass"...

Les classes d'objets

A première vue, l'entrée présentée ci-dessus constitue un amalgame de différentes informations qui ne semblent pas organisées. Mais ce n'est pas le cas! Toutes ces entrées sont induites par la présence des objectClass.

L'objectClass d'une entrée est un attribut qui permet de cataloguer cette entrée. Un objectClass définit un regroupement d'attributs obligatoires ou autorisés pour une entrée.

Une entrée peut posséder un ou plusieurs objectClass. Ce sont ces objectClass qui définissent

la présence de tous les autres attributs. Ici, l'objectClass "posixAccount" rend obligatoire les attributs cn, uid, uidNumber, gidNumber et homeDirectory. Il rend possible l'utilisation des 4 autres attributs userPassword, loginShell, gecos et description.

Les schémas

Comment savoir quels sont les objectClass disponibles et quels attributs ils contiennent ? C'est très simple, la syntaxe et la liste des attributs connus de l'annuaire sont écrites dans ce que l'on appelle les "schémas". Concrètement, un schéma est un

fichier qui décrit un à un les attributs disponibles (leur nom, leur type, etc...), ainsi que les objectClass qui y font appel. Au démarrage du serveur LDAP, le ou les fichiers de schéma spécifiés dans sa configuration sont chargés. Dans notre exemple, l'objectClass posixAccount est défini dans le fichier *nis.schema*. Etudions une partie de ce fichier, livré avec OpenLDAP et situé dans */etc/ldap/schema :*

```
# [...]
attributetype ( 1.3.6.1.1.1.1.0 NAME 'uidNumber'
      DESC 'An integer uniquely identifying a user in a domain'
      EQUALITY integerMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
# [...]
objectclass ( 1.3.6.1.1.1.2.0 NAME 'posixAccount' SUP top AUXILIARY
      DESC 'Abstraction of an account with POSIX attributes'
      MUST ( cn $ uid $ uidNumber $ gidNumber $ homeDirectory )
      MAY ( userPassword $ loginShell $ gecos $ description ) )
# [...]
```

Le fichier étant assez volumineux nous l'avons tronqué aux parties utiles pour notre exemple.

Le premier paragraphe définit l'un des attributs utilisés par le posixAccount : uidNumber. Le second, l'objectClass posixAccount. Nous n'allons pas étudier en détail ces deux définitions; simplement, il faut retenir que :

- A chaque définition correspond un OID (Object IDentifier), qui permet de rendre unique l'attribut spécifié. Ces OIDs sont déposés auprès de l'IANA (http://www.iana.org) et sont donc officiels.
- Un attribut définit un type d'égalité à mettre en oeuvre lors d'une recherche (ici, intergerMatch) ainsi que le type de données qu'il contient (l'OID spécifié après SYNTAX).
- Un objectClass définit les attributs que l'objet doit présenter (MUST) et ceux qu'il peut posséder (MAY).

Les schémas constituent donc une source d'information très importante. En cas de doute concernant le type ou le nom des attributs à spécifier dans une entrée, il faut s'y reporter!

Enfin, nous signalons qu'il est tout à fait possible de créer ses propres schémas, cependant, il faut penser à réutiliser les schémas existants : ils offrent déjà de nombreuses possibilités et il y a fort à parier qu'un schéma existe déjà pour gérer les

informations que vous souhaitez !

La sécurité (modèle de sécurité)

Lorsqu'on met en place un annuaire d'entreprise, il convient de réfléchir au modèle de sécurité que nous souhaitons appliquer. LDAP fournit plusieurs mécanismes permettant de mener à bien le projet.

L'authentification simple, binding

L'annuaire met en place un mécanisme d'authentification : pour avoir accès aux données qu'il contient, il faut s'identifier et s'authentifier.

L'une des opérations préalables à l'interrogation de l'annuaire est cette opération dite de "binding" (dans le cas d'une authentification simple). Le client envoie alors le DN d'un compte contenu dans l'annuaire lui-même, ainsi que le mot de passe associé. On pourra par la suite appliquer des droits particuliers sur ce compte en utilisant les ACLs. Ceci correspond, si l'on fait le parallèle avec l'annuaire téléphonique, à la fonctionnalité de liste rouge, où certaines données ne sont pas accessibles à tout le monde.
Enfin il est possible de se connecter de manière anonyme : le client envoie alors un DN vide au serveur LDAP.

Les ACLs

Les **ACLs** (**A**ccess **C**ontrol **L**ists) interviennent après la notion de binding. Il sera possible de donner des droits de lecture, d'écriture (ou d'autres droits divers) sur des branches particulières de l'annuaire au compte connecté. Ceci permet de gérer finement les droits d'accès aux données.

Le chiffrement des communications (SSL/TLS)

Le chiffrement des communications, via **SSL** (**Se**cure **S**ocket **L**ayer) ou **TLS** (**T**ransport **L**ayer **S**ecurity) est également une méthode de protection de l'information. Il est possible, avec la plupart des annuaires existants, de chiffrer le canal de communication entre l'application cliente et l'annuaire. Ceci permet de garantir un minimum de confidentialité des données et d'éviter qu'un tiers n'écoute les communications sur le réseau.

SASL

SASL (**S**imple **A**uthentication and **S**ecurity **L**ayer) est un mécanisme qui permet d'ajouter des méthodes d'authentification à des protocoles orientés connexion tels que LDAP ou IMAP. Il est défini dans la RFC 2222. SASL donne la possibilité au client et au serveur de sélectionner quelle sera la méthode d'authentification utilisée. Ces méthodes sont extensibles via des plugins. Il permet également de mettre en place une couche de connexion sécurisée telle que SSL/TLS (sans rapport direct avec le chiffrement indépendant des connexions que nous avons cité ci-dessus).

4.1.2 Concepts avancés

- **La réplication**

OpenLDAP, permet de manière native, de mettre en place un annuaire répliqué. Un annuaire dit "maître" envoie alors, par le biais du format LDIF, toutes les modifications effectuées sur un annuaire "esclave".

L'avantage d'une telle opération est double :

- permettre une meilleure montée en charge pour de gros annuaires : il est possible de

rediriger le client vers l'un ou l'autre des annuaires répliqués.

- disposer d'une copie conforme du premier annuaire, utile en cas de crash (attention, toute opération est reportée de l'annuaire maître vers l'esclave, donc ceci est non valable en cas de mauvaise manipulation).

Deux types de réplication existent :

- le mode "maître-esclave", le plus courant : la réplication est unidirectionnelle, un annuaire maître envoie toutes les modifications à un annuaire esclave. Ceci n'autorise bien évidemment l'écriture que sur l'annuaire maître ; l'esclave est alors disponible uniquement en lecture.
- le mode "maître-maître" : la réplication est bidirectionnelle, chaque annuaire peut être maître de l'autre. Ceci permet d'écrire indifféremment sur l'un ou l'autre des annuaires.

Enfin il est possible de chaîner les réplications pour obtenir plusieurs réplicats. Cette fonctionnalité nous sera très utile dans le cas de notre étude, vu la subdivision que nous avons choisi dès la conception de notre Intranet.

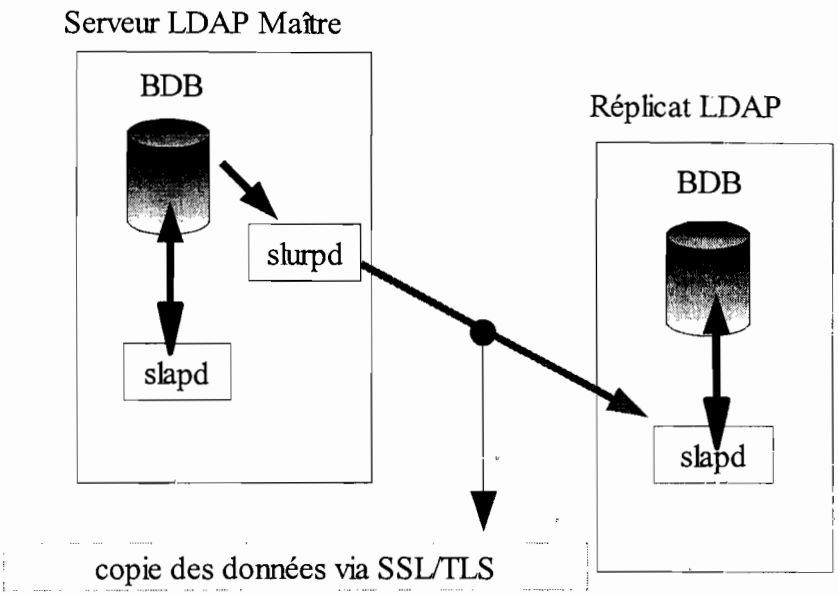


Figure II.3: Schéma de la réplication LDAP

➨ **La distribution (les referrals)**

La distribution est un mécanisme qui va permettre de faire pointer un lien vers un autre annuaire pour une branche particulière. Ceci va permettre de déléguer la gestion de cette branche, un peu au sens DNS lorsqu'on délègue la gestion d'un domaine.
Ce mécanisme peut être représenté de la manière suivante, si l'on reprend l'exemple de notre domaine univ-bobo.bf:

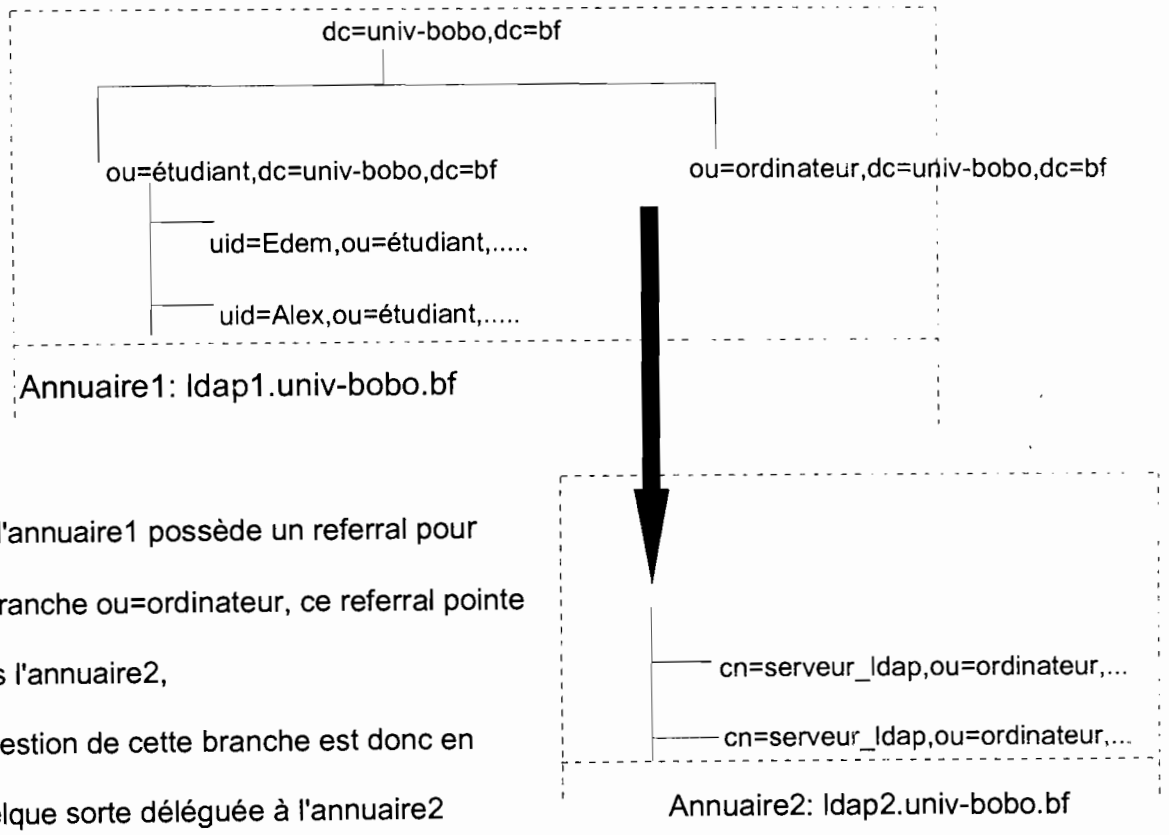


Figure II.4: Schéma de distribution LDAP

Au niveau de l'annuaire1, la délégation de gestion se traduit par une entrée de la classe "referral", qui contient alors un attribut "ref" contenant l'adresse de la suite de l'arborescence :

```
dn: ou=ordinateur,dc=univ-bobo,dc=bf
objectClass: referral
ref:ldap ://ldap2.univ-bobo.bf/ou=ordinateur,dc=univ-bobo,dc=bf
```

4.1.3 Déploiement de l'annuaire LDAP

Déployer un service d'annuaire LDAP nécessite en premier lieu une réflexion sur la nature des données que l'on y met, sur la manière dont on les récupère, sur l'utilisation que l'on compte en faire et sur la façon de gérer le tout. La mise en place d'un annuaire LDAP met donc en jeu plusieurs phases de conception que nous allons suivre.

➧ **les besoins en service d'annuaire et ses applications**

Cette phase consiste donc à prévoir toutes les applications possibles, actuelles ou futures, d'un annuaire **LDAP**. Les exemples d'applications de **LDAP** sont nombreux, mais nous choisissons son déploiement pour constituer:

- Des bases de données du personnel et des étudiants de L'UPB.
- Des bases de données de ressources matérielles.
- Des bases de données pour certaines applications, la messagerie, des login sur le réseau, shell, homedirectory.
- Ceci pourra être élargi dans le futur.

Installation

Comme pour tout logiciel sous linux, il est possible d'installer OpenLDAP par le biais de paquets binaires fournis par la distribution utilisée, ou bien en compilant les sources. L'installation par paquets est souvent conseillée car elle facilite la maintenance du logiciel par la suite.

Sur notre distribution qui est de type Debian, les paquets à installer sont les suivants : **slapd** et **ldap-utils**. En tant que root, il faut donc saisir la commande suivante :

 # aptitude install slapd ldap-utils

Une fenêtre apparaît alors et demande le mot de passe de l'administrateur associé à l'annuaire que l'on met en place.

Les outils fournis par OpenLDAP

Le projet OpenLDAP implémente un serveur LDAP, mais également les commandes clientes permettant de manipuler des informations contenues dans l'annuaire.

➧ **Les commandes liées au serveur**

Le paquet slapd fournit les binaires suivants

Démons:

- **slapd:** le démon OpenLDAP !
- **Slurpd:** le démon de réplication

Commandes de manipulation de la base (backend) gérée par OpenLDAP

- **slapindex** : crée les index au sein de la base
- **slapcat** : effectue un dump (une copie intégrale) de la base
- **slapadd** : ajoute des entrées LDIF dans la base
- **slappasswd** : utilitaire de conversion de mots de passe

Commandes de test/validation :

- **slaptest** : teste la validité du fichier de configuration slapd.conf
- **slapdn** : teste la conformité d'un DN donné en ligne de commande

Chacune de ces commandes permet d'agir directement au niveau du serveur OpenLDAP, notamment au niveau de sa base de données. Il est donc impératif de les exécuter sur le serveur où fonctionne OpenLDAP.

➨ **Les commandes clientes**

Le paquet ldap-utils fournit les commandes suivantes:

- **ldapsearch** : effectue une recherche au sein de l'annuaire
- **ldapadd** : ajoute une entrée
- **ldapdelete** : supprime une entrée
- **ldapmodify** : modifie une entrée (ajoute/suppr. un attribut, ajoute/suppr. une entrée, ...)
- **ldapmodrdn** : modifie le rdn d'une entrée (renomme une entrée)
- **ldappasswd** : modifie le mot de passe d'une entrée LDAP
- **ldapwhoami** : affiche avec quel utilisateur le binding a eu lieu
- **ldapcompare** : permet de comparer l'attribut d'une entrée à une valeur spécifiée

Chaque commande cliente utilise le protocole LDAP pour agir sur l'annuaire. Elles peuvent donc, cette fois-ci, être utilisées à distance. Elles agissent en tant que clients LDAP standards.

Configuration

L'intégralité de la configuration du serveur OpenLDAP (le démon **slapd**) s'effectue en modifiant le fichier */etc/ldap/slapd.conf*, situé dans le répertoire */etc/ldap*.

Voici le contenu du fichier de configuration commenté dans notre cas.

```
#######################################################################
# Directives globales
# Inclusion des schemas
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/inetorgperson.schema
# Vérification de la conformité des objets avec les schémas
schemacheck on
# Ou sera stocke le PID du demon
pidfile /var/run/slapd/slapd.pid
# Liste des arguments passes au démarrage du serveur
argsfile /var/run/slapd.args
# Niveau de log
loglevel 0
# Emplacement des modules
# Chargement du module BDB (Berkeley DB)
modulepath /usr/lib/ldap
moduleload back_bdb
#######################################################################
# Déclaration des options pour le premier type de backend utilise : bdb
# Toutes les options s'y appliquent jusqu'à la prochaine directive
# backend
backend bdb
checkpoint 512 30
#######################################################################
#backend <autre>
#######################################################################
# Déclaration des options de la première "base", c'est a dire de la
# première (et unique ici) arborescence gerée par notre annuaire
# Toutes les options s'y appliquent jusqu'au la prochaine directive
# database
database bdb
# La racine de notre arborescence
```

```
suffix "dc=univ-bobo,dc=bf"
# Le compte administrateur de notre arborescence et son mot de passe
rootdn "cn=admin,dc=univ-bobo,dc=bf"
rootpw "secret"
# Ou sont stockes les fichiers BDBs de notre arborescence
directory "/var/lib/ldap"
# Options d'index
index objectClass eq
# Sauvegarde de l'heure a laquelle est modifiée une entrée
lastmod on
# ACLs de notre première arborescence :
# Une personne non authentifiée peut s'authentifier
# Une personne authentifiée peut modifier son propre mot de passe
# Les autres n'ont pas accès à l'attribut mot de passe
access to attrs=userPassword
by anonymous auth
by self write
by * none
# Tout le monde peut lire l'annuaire
access to *
by * read
#[...]
```

Le fichier de configuration est subdivisé en trois sections importantes :

- la section globale (début du fichier)
- la section concernant les options de backends (débute par "backend")
- la section concernant les déclarations et les options des arborescences gérées (débute par "database").

Cette configuration est celle par défaut qui sera utilisée sur l'ensemble des serveurs LDAP de notre intranet. Pour répondre à notre plan, nous allons utiliser la réplication; c'est -à dire que nous aurons un serveur OpenLDAP dans le LAN et un dans la DMZ.

➧ **Les comptes POSIX - Nsswitch**

Il est recommandé de créer un utilisateur LDAP spécifique (classe d'objet **person**) pour la réplication. Celui-ci ne sera ainsi utilisé que par l'annuaire maître pour accéder à un annuaire esclave. Vous pouvez, par ailleurs, choisir un utilisateur différent par annuaire esclave. Voici le contenu du fichier au format LDIF pour notre utilisateur: *utilsateur_replicat.ldif*

```
dn: cn=replication,ou=Systeme,dc=univ-bobo,dc=bf
objectClass: top
objectClass: person
cn: replication
# sn est obligatoire
sn: replication
userPassword: {MD5}8p+r5jekcr9SIrEqC8Xfdw==
# mot de passe : 'replication' obtenu grâce avec la commande : slappasswd -h '{MD5}'
```

L'ajout à l'annuaire maître se fait avec la commande suivante :

#ldapadd -H ldap://<adresse ou nom de la machine maître> -D "cn=admin,dc=uiv-bobo,dc=bf" -W -x -f <utilsateur_replicat.ldif

Configuration de l'annuaire maître

Il faut apporter quelques modifications à la configuration de l'annuaire maître afin de faire fonctionner la réplication. En effet, l'utilisateur associé à la réplication doit avoir accès en lecture aux objets de l'annuaire (éventuellement partielle) et il faut également lui indiquer le serveur LDAP accueillant la réplication pour qu'il puisse communiquer avec ce dernier. Par conséquent, voici les modifications et/ou ajouts à apporter à votre fichier */usr/local/etc/openldap/slapd.conf* :

```
# ...

# ACL : donner accès en lecture à l'annuaire entier à l'utilisateur LDAP assurant la réplication
access to *
      by dn="cn=admin,dc=univ-bobo,dc=bf" write
      by dn="cn=replication,ou=Systeme,dc=univ-bobo,dc=bf" read
      by users read

# ...
# Réplication
# Comment contacter l'annuaire esclave ?
replica
   uri=ldap://<adresse ou nom de l'annuaire esclave>
   binddn="cn=replication,ou=Systeme,dc=univ-bobo,dc=bf"
   bindmethod=simple credentials=<le mot de passe associé à ce compte en clair>
# Journal
replogfile /var/db/openldap-slurp/replica/replog

# ...
```

Configuration de l'annuaire esclave

Il faut autoriser l'utilisateur de réplication à écrire et lui indiquer qui est son serveur LDAP maître. Voici les modifications et/ou ajouts à apporter à votre fichier */usr/local/etc/openldap/slapd.conf* :

```
# ...
# ACL : donner accès en écriture à l'annuaire entier à l'utilisateur assurant la réplication
access to *
      by dn="cn=admin,dc=univ-bobo,dc=bf" write
      by dn="cn=replication,ou=Systeme,dc=univ-bobo,dc=bf" write
      by users read
# ...
# Réplication
# DN de l'utilisateur de réplication
updatedn      "cn=replication,ou=Systeme,dc=univ-bobo,dc=bf"
# Comment joindre l'annuaire maître ?
updateref     "ldap://<adresse ou nom de l'annuaire maître>"
```

Préparation du démarrage automatique

le fichier */etc/rc.conf* de l'annuaire maître doit contenir :

```
# LDAP
slapd_enable="YES"
slapd_flags="-h ldap:///"
# Réplication
slurpd_enable="YES"
```

le fichier */etc/rc.conf* de l'annuaire esclave doit contenir :

```
# LDAP
slapd_enable="YES"
slapd_flags="-h ldap:///"
```

- **Administration du serveur**

Notre serveur est désormais configuré. Nous allons maintenant voir comment nous pouvons l'administrer.

L'administration du serveur passe par l'utilisation des commandes slap (,,,): celles-ci n'utilisent pas le protocole LDAP mais accèdent directement à la base de données sous-jacente (BDB dans notre cas). Il est donc **impératif** de toujours couper le serveur LDAP par la commade: **# /etc/init.d/slapd stop** afin d'éviter un accès concurrent depuis le serveur lui-même, ce qui pourrait corrompre la base de données.

Slapindex

Nous avons configuré notre serveur pour qu'il utilise des index ; la première chose à effectuer avant d'utiliser notre serveur est donc de les générer. Il faut en effet initialiser les index pour qu'OpenLDAP puisse ensuite les utiliser et les maintenir.

L'opération de génération n'est à effectuer qu'une seule fois et ceci se fait par le biais de la commande slapindex. **# slapindex**

Slapcat

Slapcat est une commande très utile au quotidien. Elle effectue un "dump" de la base LDAP au format LDIF. Il est conseillé de l'utiliser régulièrement pour effectuer des sauvegardes de notre annuaire. Par défaut, slapcat affiche les informations sur la sortie standard, il faut donc la rediriger vers un fichier pour obtenir notre sauvegarde :

 # slapcat > sauvegarde.ldif

Arrêt et démarrage du serveur

L'arrêt et le démarrage du serveur LDAP se font par le biais du script /etc/init.d/slapd :

/etc/init.d/slapd [start|stop|restart]

- **Initialisation de l'annuaire**

L'initialisation de l'annuaire n'est qu'un ajout massif de plusieurs entrées. Cet ajout massif peut se faire par le biais de **slapadd** si vous possédez déjà une sauvegarde de l'annuaire et si vous vous situez sur le serveur.

A distance, c'est l'outil **ldapadd** qui va nous permettre d'effectuer cette opération. Il suffit de fournir à ldapadd un fichier LDIF contenant plusieurs entrées qui seront ajoutées dans le même ordre avec lequel elles apparaissent dans le fichier. Ce fichier va donc tout d'abord contenir l'entrée de la racine, qui est nécessaire, puis chacune des "ou" que nous aurons choisi. Le contenu du fichier *arbre_annuaire.ldif*

```
dn: ou=ESI, dc=univ-bobo, dc=bf
objectClass: top
objectClass: organizationalUnit
ou: ESI

dn: ou=UIT, dc=univ-bobo, dc=bf
objectClass: top
objectClass: organizationalUnit
ou: UIT

dn: ou=IDR, dc=univ-bobo, dc=bf
objectClass: top
objectClass: organizationalUnit
ou: IDR
```

```
dn: ou=ISEA, dc=univ-bobo, dc=bf
objectClass: top
objectClass: organizationalUnit
ou: ISEA

dn: ou=ISNV, dc=univ-bobo, dc=bf
objectClass: top
objectClass: organizationalUnit
ou: ISNV

dn: ou=Présidance, dc=univ-bobo, dc=bf
objectClass: top
objectClass: organizationalUnit
ou: Présidance

dn: ou=étudiant,ou=ESI,dc=univ-bobo,dc=bf
objectClass: top
objectClass: organizationalUnit
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
ou=étudiant
```

Insertion de cette arborescence

 # ldapadd -W -D "cn=admin,dc=univ-bobo,dc=bf" -x -H ldap://localhost -f fichier.ldif

Exemple d'enrégistrement d'un étudiant.

```
dn: uid=Edem Kodjo,ou=étudiant,ou=ESI,dc=univ-bobo,dc=bf
structuralObjectClass: inetOrgPerson
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
cn: Edem Kodjo
sn: Le Root
uid: A. Edem Kodjo
uidNumber: 1001
```

gidNumber: 1001
mail: edem_kodjo@univ-bobo.bf
homeDirectory: /home/ldap/Edem
userPassword: {SHA}fEqNCco3Yq9h5ZUglD3CZJT4lBs=
loginShell: /bin/bash
gecos: Edi
description: Administrateur système de l'assoc-linux

Représentation schématique des enregistrements qui viennent d'être effectués.

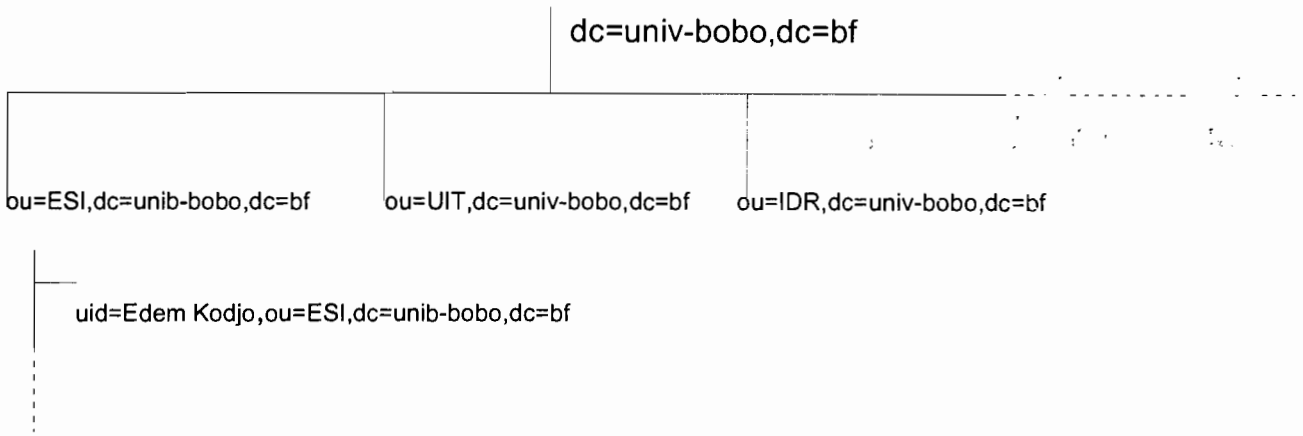


Figure II.5: Schéma de l'arborescence des enregistrements

- **Les outils graphiques d'administration**

Nous avons étudié comment interagir avec notre annuaire par le biais des lignes de commandes. Il existe des outils graphiques pour nous simplifier la tâche au quotidien ! Trois sont particulièrement intéressants, il s'agit de gq, ldapbrowser et phpldapadmin.

Gq

• Développé en C / GTK

• URL pour le télécharger : **http://sourceforge.net/projects/gqclient**

Ldapbrowser

• Développé en JAVA

• URL pour le télécharger : **http://www-unix.mcs.anl.gov/~gawor/ldap/index.html**

PhpLDAPAdmin

- Développé en PHP
- URL pour le télécharger: http://phpldapadmin.sourceforge.net

En somme, LDAP par le biais de sa standardisation, permet une interopérabilité simple et fiable. Il offre ainsi l'avantage de pouvoir centraliser l'information au sein d'une entreprise : comptes POSIX, adresses de messagerie, et autres informations y trouvent leur place.

OpenLDAP offre une implémentation complète et robuste de ce standard grâce au serveur et aux outils clients qu'il propose. Les services qui seront configurés par la suite seront connectés à l'annuaire pour être sous le contrôle de ce dernier.

4.2 Les services internes du réseau

4.2.1 Serveur DHCP

L'affectation et la mise à jour d'informations relatives aux adresses IP fixes (cas où il n'ya pas de serveur DHCP) peuvent représenter une lourde tâche. Afin de faciliter ce travail et de simplifier la distribution des adresses IP, le protocole DHCP offre une configuration dynamique des adresses IP et des informations associées.

Le principe du DHCP

Voici le principe de fonctionnement du service DHCP:

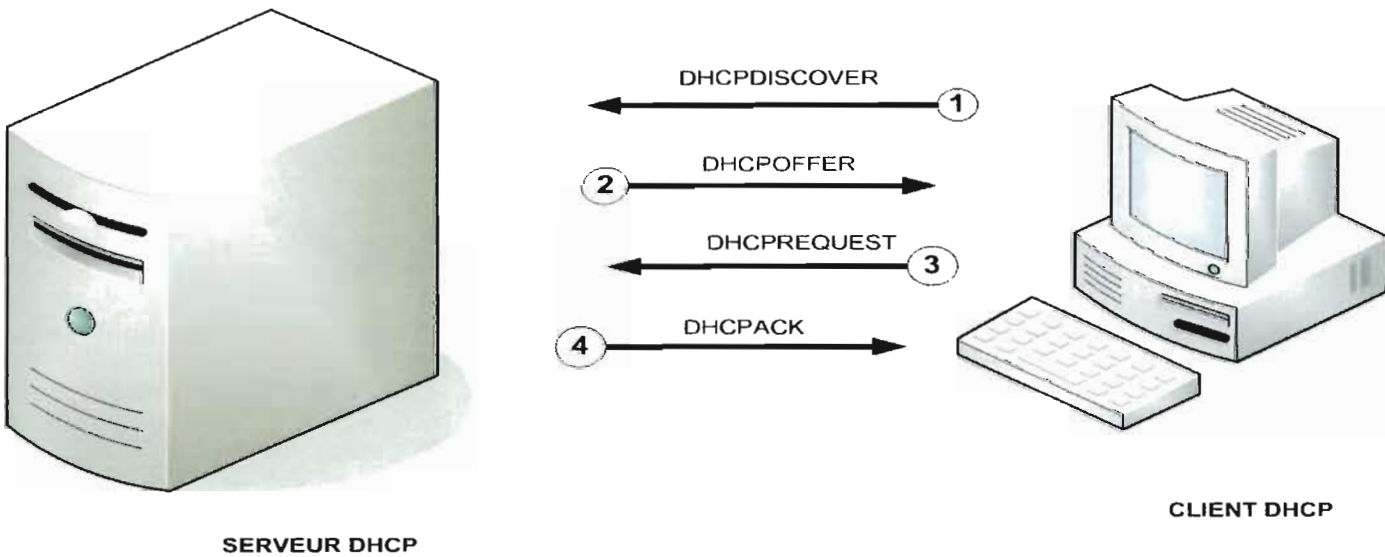


Figure II.6: Schéma illustrant le principe de fonctionnement du DHCP

- Lorsque le client DHCP démarre, il n'a aucune connaissance du réseau, du moins, en principe. Il envoie donc une trame **"DHCPDISCOVER",** destinée à trouver un serveur DHCP. Cette trame est un "broadcast"(elle est destinée à toutes les machines du réseau), donc elle est envoyée à l'adresse 255.255.255.255. N'ayant pas encore d'adresse IP, il adopte provisoirement l'adresse 0.0.0.0. Comme ce n'est pas avec cette adresse que le DHCP va l'identifier, il fournit aussi son adresse MAC.
- Le, ou les serveurs DHCP du réseau qui vont recevoir cette trame vont se sentir concernés et répondre par un **"DHCPOFFER"** qui est une trame contenant une proposition de bail et l'adresse MAC du client, avec également l'adresse IP du serveur. Tous les serveurs DHCP répondent et le client normalement accepte la première réponse venue.
- Le client répond alors par un **DHCPREQUEST** à tous les serveurs (donc toujours en "Broadcast") pour indiquer quelle offre il accepte.
- Le serveur DHCP choisi répond définitivement par un **DHCPACK** qui constitue une confirmation du bail. L'adresse du client est alors marquée comme utilisée et ne sera plus proposée à un autre client pour toute la durée du bail.

Le bail

Dans le bail, il y a non seulement une adresse IP pour le client, avec une durée de validité, mais également d'autres informations de configuration comme:

- L'adresse d'un ou de plusieurs DNS (pour la résolution de noms)
- L'adresse de la passerelle par défaut (pour sortir de son réseau)
- L'adresse du serveur DHCP.

L'installation

Pour installer le serveur dhcp (appelé **dhcpd**) sous Debian, la commande est la suivante:

 # aptitude install dhcp3-server

Configuration

Il y a deux fichiers à renseigner afin de configurer le serveur dhcp:

- le fichier /etc/default/dhcp3-server

IL contient l'interface sur laquelle doit écouter le serveur. On modifie la ligne INTERFACES=" ", qui est d'ailleurs la seule ligne du fichier (à part les lignes de commentaire), en y ajoutant l'interface que l'on veut. Par exemple, si le serveur dhcp doit tourner sur l'interface eth0 la ligne doit correspondre à: INTERFACES="eth0"

- mais le fichier le plus important à modifier est */etc/dhcp3/dhcpd.conf*

C'est lui qui permet de définir toute la configuration du serveur DHCP. Plus la configuration est avancée et plus ce fichier est complexe.

Voici ce que peut contenir ce fichier dans le cas d'une configuration très simple.

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.10 192.168.1.250;
    option routers 192.168.1.1;
    option broadcast-address 192.168.1.255;
    option domain-name "univ-bobo.bf";
    option domain-name-servers 192.168.1.3;
}

host serverX {
    hardware ethernet 00:C0:9F:AF:83:85;
    fixed-address 192.168.1.2;
}
```

Le serveur DHCP lui-même doit avoir une adresse fixée car il ne s'attribue pas une adresse.

Nous déclarons un réseau local en 192.168.1.0/255.255.255.0 dont les adresses seront automatiquement attribuées entre 192.168.1.10 et 192.168.1.250.

De plus, le nom de domaine est précisé par: option domain-name "univ-bobo.bf"; et le serveur DNS est déclaré grâce à l'option domain-name-servers 192.168.1.3. L'adresse IP 192.168.1.2 sera associée à l'ordinateur dont l'adresse MAC est 00:C0:9F:AF:83:85 et qui est peut-être une autre machine serveur du réseau.

Dans la plupart des cas, la configuration de ce fichier n'est pas si simple. En effet, il peut être intéressant d'utiliser des pools et des classes (d'adresses) pour créer différents "sous-réseaux" afin de permettre une meilleure localisation des ressources sur le réseau et aussi une meilleure organisation. Il est ainsi possible de distinguer un serveur, d'un poste utilisateur et mieux encore, d'une machine étrangère au réseau; un firewall basé sur cette architecture pourrait interdire ses ressources à des machines étrangères, ou leur interdire d'accéder à Internet.

Vu la complexité de notre réseau informatique, nous allons mettre en place deux serveurs DHCP: l'un dans le réseau des administrations et l'autre dans le réseau académique. Chacun des deux serveurs permettra de desservir plusieurs sous-réseaux issus de la subdivision des deux réseaux cités ci-dessus. Nous allons juste présenter le fichier de configuration pour le serveur DHCP du réseau académique. Celui du réseau des administrations aura la même allure:

```
[....]
# Création d'une classe pour le sous-réseau de l'ESI
class "sous_rezo_1" {
    match if substring (option dhcp-client-identifier,0,5) = "ESI";
}

# Création d'une classe pour le sous-réseau de l'IUT
class "sous_rezo_2" {
    match if substring (option dhcp-client-identifier,0,5) = "IUT";
}

#Création d'une classe pour le sous-réseau de l'IDR
class "sous_rezo_3" {
    match if substring (option dhcp-client-identifier,0,5) = "IDR";
}

# Créer le réseau pour les machines étrangères
subnet 192.168.1.0 netmask 255.255.255.0 {
    # Associer la plage [230-250] pour les machines inconnues sur le réseau
    pool {
        deny members of "sous_rezo_1";
        deny members of "sous_rezo_2";
        deny members of "sous_rezo_3";
        range 192.168.1.230 192.168.1.250;
    }
    # Associer la plage [20-100] aux machines de niveau 1
    pool {
        allow members of "sous_rezo_1";
        deny members of "sous_rezo_2";
        deny members of "sous_rezo_3";
        range 192.168.1.20 192.168.1.100;
    }
    # Associer la plage [101-160] aux machines de niveau 2
    pool {
```

```
        deny members of "sous_rezo_1";
        allow members of "sous_rezo_2";
        deny members of "sous_rezo_3";
        range 192.168.1.101 192.168.1.160;
    # Associer la plage [161-229] aux machines de niveau 3
    pool {
        deny members of "sous_rezo_1";
        deny members of "sous_rezo_2";
        allow members of "sous_rezo_3";
        range 192.168.1.161 192.168.1.229;
    }

        # Configuration d'une machine serveur via son adr MAC
        host serveurX {
            hardware ethernet 00:C0:9F:AF:83:85;
            fixed-address 192.168.1.2;
        }
    }
}

[....]
```

L'option dhcp-client-identifier permet de définir les classes. Pour des clients sous Linux, il faut ajouter cette option dans le ficher **/etc/dhcp3/dhclient.conf** des différents postes clients en fonction de leur groupe. Par exemple, la ligne suivante doit figurer dans le fichier ci-dessus pour les ordinateurs du sous-réseau de l'ESI:

```
send dhcp-client-identifier "ESI_PCx"
```

Avec **x** le numéro du poste client.

Pour les clients sous Windows, il faut exécuter la commande suivante sous DOS:

 # ipconfig /setclassid "Connexion au sous-réseau" ESI_PC1

Ensuite, pour que le DHCP marche, il est très important de lancer le démon dhcp en exécutant la commande suivante sur le serveur : **# /etc/init.d/dhcp3-server start**

Il faut également le relancer chaque fois qu'on modifie les fichiers de configuration et qu'on veut prendre en compte les modifications.

4.2.2 Serveur SAMBA

Le service Samba est une des composantes de notre serveur de fichiers.

Principe

Un serveur Samba permet dans un réseau, de partager des ressources entre des ordinateurs ayant des systèmes d'exploitation différents (Linux, Windows, MAC/OS, etc). Il assure donc une certaine hétérogénéité des machines du réseau du point de vu des systèmes d'exploitation utilisés.

Samba utilise le protocole SMB (Server Message Block) dont son nom dérive d'ailleurs et qui s'appuie sur NetBios, un autre protocole de transfert de fichiers plus ancien. Samba met en oeuvre deux processus serveurs ou "deamons" : **smbd** pour le partage de ressources proprement dit et **nmbd** qui assure la résolution des noms Netbios.

Installation

Elle se fait simplement par la commande : **# aptitude install samba**.

Configuration

- **Le fichier de configuration**

  Sous Debian la configuration se fait dans **/etc/samba/smb.conf** qui comprend essentiellement deux parties:

Une partie "générale" contenant la section **[global]** qui définit le comportement général du serveur et la stratégie adoptée pour les services communs.

L'autre partie définit les ressources partagées et les permissions d'accès. Dans cette partie, on retrouve la section **[homes]** qui permet de partager les répertoires personnels des utilisateurs Linux, de même que des sections définies pour le partage des imprimantes (**[printers]**), et d'autres répertoires sur le serveur.

Il est toujours possible de modifier et d'ajouter des sections, pour définir de nouvelles

ressources à partager.

Les sections sont configurées par des paramètres auxquels sont affectées des valeurs.

Les principaux paramètres de configuration Pour la section GLOBAL sont :

Tableau II.7: paramètres de configuration de la section globale de SAMBA

paramètre	**valeur par défaut**	**description**
workgroup =		le nom du groupe de travail ou du domaine (Windows) des postes clients
netbios name =		le nom du serveur Samba
guest account =	nobody	le compte à utiliser pour les accès invités aux partages
share modes = yes|no	yes	accès multi utilisateur ou non
interfaces =		l'adresse IP de la carte réseau du serveur
printcap =	/etc/printcap	emplacement du fichier printcap, récapitulant toutes les imprimantes installées sur le serveur
load printers = yes|no	yes	partager ou non toutes les imprimantes définies dans le fichier printcap
log file =	/var/log/samba/log.%m	fichier log pour les machines qui se connectent
security = user|share	user	mode de sécurité

Pour les autres sections on a:

Tableau II.8: paramètres de configuration des sections secondaires de SAMBA

paramètre	valeur par défaut	description
path =		chemin du répertoire à partager
comment =		texte visible dans le voisinage réseau client
guest ok = yes|no	no	partage en accès libre sans authentification
valid users =	tous	liste des users autorisés à se connecter à la ressource
printable = true| false	false	partage d'un service d'impression et non de répertoire.
writeable = yes| no	no	permet ou non l'écriture sur le répertoire, contraire de read only
write list =	tous les utilisateurs	liste des users autorisés à écrire
browseable =	yes	visibilité du partage par tous, *même les users non autorisés*
create mode | mask =	0744	droits maximum accordés à un fichier créé dans la ressource; ces droits seront en intersection (and) avec les droits Linux(umask)
directory mode | mask=	0755	droits maximum accordés à un répertoire créé dans la ressource;ces droits seront en intersection (and) avec les droits Linux(umask)
force directory mode =	000	droits imposés lors de la création du répertoire. composé par un opérateur OR avec les droits usuels
force group =		Impose un groupe propriétaire d'un fichier lors de sa création dans le partage
hide dot files =	yes	cache les fichiers cachés au sens Linux, commençant par un point
hosts allow = hosts deny =	toutes les stations aucune	ressource réservée|interdite à la liste des stations (adresses IP)
max connections	0	nombre de connexions à la ressource illimité,

=		sinon maximum

Ceci est notre fichier ***/etc/samba/smb.conf*** largement commenté.

```
[global]
   # Nom du groupe samba
workgroup = group_samba

   # accès multi utilisateur
share modes = yes ;

   # restreindre par sécurité les sous-réseaux autorisés à se connecter au serveur
   # ici on se limite aux adresses réseau privé 192.168.1.0
   et à l'interface "loopback"
hosts allow = 192.168.1.   127.

   # indique l'adresse IP de l'adaptateur du serveur et le masque de sous réseau
interfaces = 192.168.1.30/255.255.255.0

   # indique l'emplacement du fichier printcap, récapitulant
    toutes les imprimantes installées sur le serveur Linux
printcap = /etc/printcap
   # partage toutes les imprimantes définies dans le fichier printcap
load printers = yes

   # utiliser un fichier de trace pour chaque machine qui se connecte
log file = /var/log/samba/log.%m

   # choisir le mode de sécurité : user ou share
security = user


# Ce paragraphe permet de rendre les répertoires personnels des utilisateurs sur le
serveurs, accessibles depuis les postes clients

[homes]
comment =Répertoire personnel
#Pour que seul le propriétaire ait accès
browsable = no
#L'accès sera total
writable = yes
create mode = 0700
# Ici nous partageons un répertoire pour tous les utilisateurs


[public]
# Ce répertoire aura pour nom de partage " public "([public]),
# la valeur du champ  comment apparaîtra dans le voisinage réseau
```

```
comment =Répertoire public
path = /home/tmp
# il pourra être accessible par tous les utilisateurs
public = yes
# il est accessible en écriture
writeable = yes
# les fichiers créés sont en lecture seule, sauf pour le propriétaire
create mode = 0755


#configurer un partage de répertoire pour un groupe


 # Ce répertoire aura pour nom de partage stagiaire
[stagiaire]
comment =Partage pour le groupe stagiaire exclusivement
# Le répertoire à partager est /home/partage
path = /home/partage
# il ne pourra pas être accessible par tous les utilisateurs
public = no
# liste des utilisateurs autorisés
valid users = AP2 AP3 CICI2
# les utilisateurs autorisés pourront y écrire
writeable = yes
# permissions par défaut des fichiers créés
create mode = 0640


# Partager des applications sur le serveur
[logiciels]
comment = Applications partagées sur le serveur
path = /appli
public = yes
# le rép. ne doit pas être en lecture seule pour tous
writeable =no
# le groupe admin peut seul installer les applications
write list = admin
# Partager le lecteur de cd-rom
[cdrom]
# chemin d'accès au pseudo-répertoire de montage du CD
path = /media/cdrom
# accessible à tous les utilisateurs
public = yes
# l'écriture sera interdite
writeable = no
```

➧ La création des comptes Samba

Les comptes systèmes Linux et les comptes Samba sont différents, même si on peut créer un compte Samba associé à chacun des comptes systèmes. Pour créer un compte Samba, on utilise la commande **smbpasswd**. Ces comptes seront utilisés avec le mot de passe associé pour accéder aux ressources depuis les postes clients.

➧ Configuration des postes clients et accès aux ressources sur le serveur.

Sur les postes Windows, il faut s'assurer que les protocoles TCP/IP et Netbios sont installés, que les clients sont dans le même réseau que le serveur Linux et que le groupe de travail ou le domaine est celui défini dans le fichier /etc/samba/smb.conf du serveur.

Après un redémarrage aller dans:

- Favoris réseaux,
- Voir les ordinateurs du groupe de travail,
- Réseaux Microsoft Windows,
- Sélectionnez le nom de votre groupe de travail Samba, puis cliquez sur votre serveur.
- Vous devrez alors vous identifier puis vous authentifier avec un des comptes définis sur votre serveur Samba pour accéder aux ressources partagées.

Connexion de Samba à l'annuaire

Il reste à configurer OpenLDAP pour qu'il soit capable de gérer les comptes Samba et Samba pour qu'il aille chercher ses comptes sur l'annuaire.

La première étape consiste à copier le schéma de Samba (fourni par le paquet samba-doc) dans le répertoire **/schema** d'OpenLDAP et de l'inclure dans la configuration. De cette manière, OpenLDAP pourra gérer les comptes Samba :

 # cp /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz

/etc/ldap/schema

 # cd /etc/ldap/schema ; gunzip samba.schema.gz

On ajoute ensuite la ligne suivante à la fin des "include" dans le fichier */etc/ldap/slapd.conf* :

```
include /etc/ldap/schema/samba.schema
```

Configuration de Samba

La dernière étape consiste à modifier la configuration de Samba (***/etc/samba/smb.conf***) pour lui indiquer où stocker ses propres comptes. Il faut pour ceci modifier la directive "passdb (fourni par le paquet samba-doc) dans le répertoire **/schema** d'OpenLDAP et de l'inclure dans la configuration. De cette manière, OpenLDAP pourra gérer les comptes Samba :

Le fichier de configuration minimal suivant ***/etc/samba/smb.conf,*** inclut un partage simple de données :

```
[global]
# Identification Netbios
workgroup = Workgroup
netbios name = serveur_de_fichier
# Controle de domaine desactive
os level = 40
domain logons = no
domain master = no
local master = no
# Base de donnees de comptes
passdb backend = ldapsam:ldap://localhost
ldap admin dn = "cn=admin,dc=univ-bobo,dc=bf"
ldap ssl = off
ldap delete dn = no
ldap group suffix = ou=ESI
ldap suffix = dc=martymac,dc=com
# Authentification via la base de comptes locale
security = user
# Securite
```

```
encrypt passwords = yes
# Gestion des logs
log file = /var/log/samba/%m.log
log level = 2
# Partage accessible uniquement au groupe sambausers
[donnees]
path = /data/samba/donnees
comment = Partage Donnees
writeable = yes
browsable = yes
guest ok = no
valid users = @utilisateurs
```

Ce fichier indique que le "passdb backend" à utiliser est notre annuaire. Nous précisons également divers éléments tels que l'emplacement des comptes utilisateurs, groupes et machines. Samba va avoir besoin d'écrire dans notre annuaire : nous spécifions donc quel est le compte administrateur LDAP à utiliser ("ldap admin dn"). Cependant, aucun mot de passe n'est précisé dans le fichier de configuration. Pour des raisons de sécurité, on va indiquer le mot de passe en lignes de commandes de cette manière :

 # smbpasswd -w mot_de_passe

Samba va stocker ce mot de passe dans un autre fichier : ***mot_de_passe.tdb.***

Enfin, on prendra soin de créer le répertoire /data/samba/donnees et de donner les bons droits au groupe "utilisateurs"...

 # mkdir -p /data/samba/donnees

 # chgrp utilisateurs /data/samba/donnees

 # chmod 775 /data/samba/donnees

Et de redémarrer Samba :

 # /etc/init.d/samba restart

4.2.3 Serveur NFS

NFS signifie Network File System. C'est, comme son nom l'indique, un système de

fichiers en réseau qui permet de partager ses données, principalement entre systèmes UNIX. À la différence de SAMBA, NFS gère les permissions sur les fichiers et on peut donc l'utiliser de manière totalement transparente dans son arborescence Linux.

Installation

 # aptitude install nfs-kernel-server

configuration

➧ Le serveur

Les 3 fichiers de configuration principaux sont */etc/exports, /etc/hosts.deny* et */etc/ hosts.allow.*

/etc/exports

Le fichier */etc/exports* est très simple :

Ses lignes présentent les répertoires partagés selon la syntaxe suivante:

répertoire machine1(option11,option12) machine2(option21,option22)

- répertoire :

  le répertoire du serveur à partager.

- machine :
  Une liste de machines séparée par des virgules et autorisées à monter ce répertoire (utilisez des adresses IP plutôt que des noms à cause des problèmes de "*dns spoofing*").
- options :

  ro : C'est la valeur par défaut, lecture seule.
  rw : La machine à un accès en lecture/écriture au répertoire.
  no_root_squash : Les accès par l'utilisateur root sur le serveur se font sous l'identité root, au contraire de nobody (par défaut).

Par exemple :

/home 192.168.1.10(rw) 192.168.1.25(ro)

signifie que l'on autorisera la machine *192.168.1.10* à accéder à notre répertoire /home en lecture et écriture (rw) ainsi que la machine *192.168.1.25* mais uniquement en lecture (ro).

Pour un bon fonctionnement : il faut avoir les mêmes numéros de groupes et d'utilisateurs sur les deux machines.
Des systèmes permettent de gérer cela, NIS (assez ancien) ou LDAP (plus récent). Avec peu d'utilisateurs, il faut tout simplement éditer /etc/group et /etc/passwd pour synchroniser ces numéros.
Il n'est pas recommandé d'exporter un système DOS ou VFAT à cause de leurs absences de gestion multi-utilisateurs ; ils ne sont pas faits pour être partagés avec NFS.

/etc/hosts.deny
On va interdire toutes les machines qui ne sont pas autorisées explicitement dans le */etc/hosts.allow.*
Pour interdire l'accès à tous les services à partir de toutes les machines la mention "ALL: ALL" suffit. On peut cependant être plus précis en écrivant :
portmap:ALL
lockd:ALL
mountd:ALL
rquotad:ALL
statd:ALL

/etc/hosts.allow
Dans le même esprit que pour le /etc/hosts.deny, ce fichier a l'architecture
[service]: [IP de la machine client]
Donc pour autoriser 192.168.1.34 à se connecter à un partage NFS, on écrira
portmap:192.168.1.34
lockd:192.168.1.34
mountd:192.168.1.34
rquotad:192.168.1.34
statd:192.168.1.34

➨ **Le client**

Pour utiliser NFS v4, il faut au minimum la version 2.10m du programme mount. Pour

voir sa version, taper : **# mount -V**

Pour monter le partage, on tape:

 # mount mon.serveur.nfs:/home /mnt/home

En principe tout devrait bien se dérouler.

Pour monter ce partage définitivement à chaque démarrage de la machine, éditons notre /etc/fstab:

```
# device mountpoint fs-type options dump fsckorder
master.foo.com:/home /mnt nfs rw 0 0
```

4.3 Les services de la DMZ

4.3.1 Serveur Web

Un serveur Web dans le sens des réseaux actuels, notamment dans l'environnement Linux, ne se limite pas au simple serveur HTTP mais inclut de nombreuses autres applications lui apportant diverses fonctionnalités. Une combinaison très implémentée est LAMP (Linux, Apache, Mysql, Php) qui combine sous Linux le serveur HTTP Apache avec le SGBD (Système de Gestion de Base de Données) Mysql et la plate-forme Php pour générer des pages web dynamiques. Pour héberger les sites web de l'UPB, nous avons opté pour la mise en place d'un LAMP.

L'installation des applications pour le serveur Web

- **Installation de apache 2**

Il nous faut tout d'abord installer le serveur HTTP apache2 qui permettra d'afficher les différentes pages.

 # aptitude install apache2

Pour s'assurer du bon fonctionnement d'Apache, On saisit l'URL suivante dans le navigateur Internet : *http://adresse_du_serveur_Web*. Si Apache a été correctement installé il s'affiche une page Web dans laquelle apparaît l'index du répertoire Web ainsi

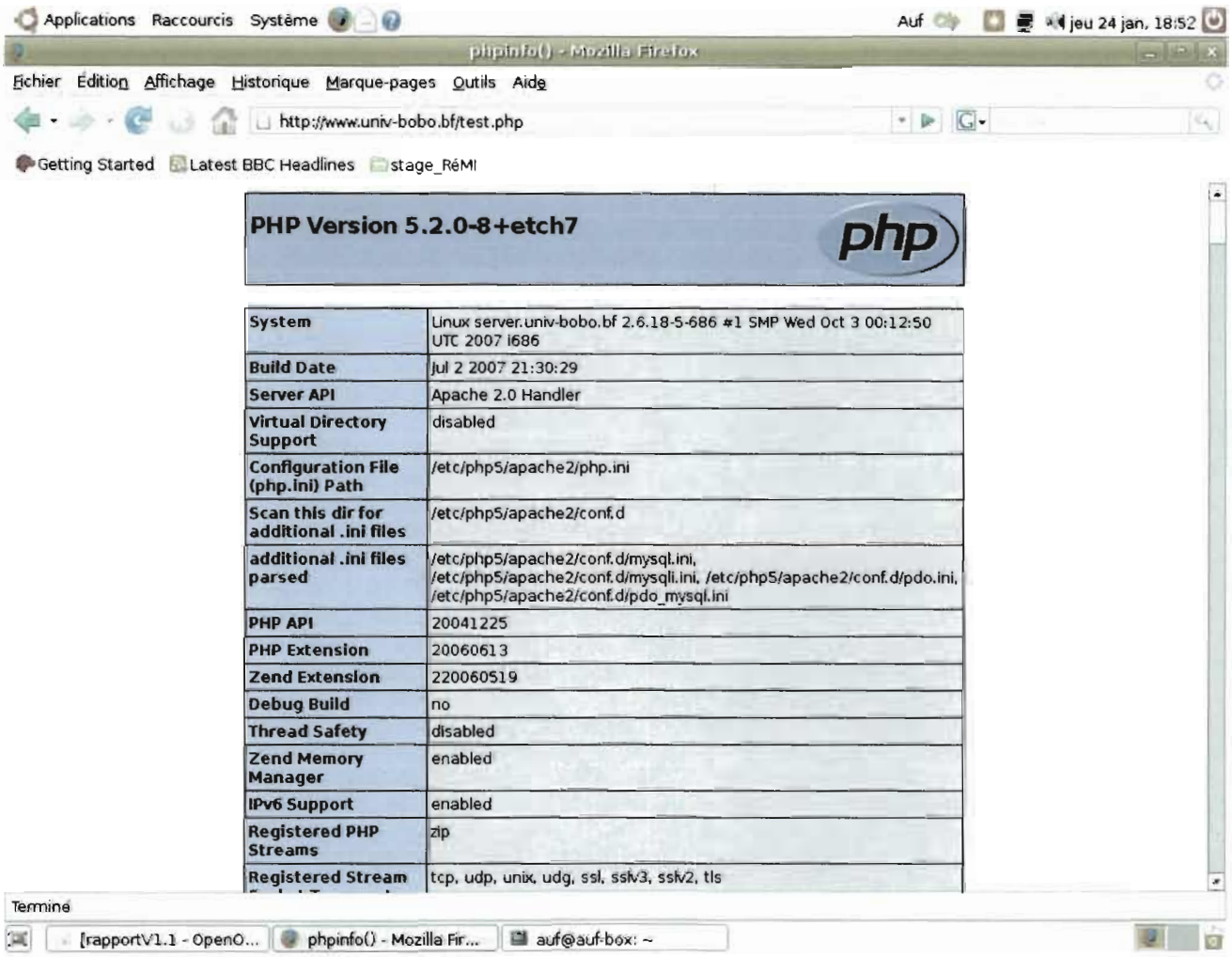
poursuivons par l'installation du php sur le serveur.

 # aptitude install php5

Pour vérifier si php a été correctement installé nous allons créer un fichier phpinfo.php dans le répertoire ***/var/www***. Pour cela, tapez dans un terminal :

 # echo "<? php phpinfo(); ?>" > /var/www/phpinfo.php

En saisissant http://adresse_de_la_machine/phpinfo.php cela nous affiche des informations concernant php5. Ceci en est une capture d'écran:



- ✔ **Installation du SGBD Mysql**

Php est très souvent couplé à un système de gestion de base de données : Mysql

- **Installation du SGBD Mysql**

Php est très souvent couplé à un système de gestion de base de données : Mysql dans notre cas. Nous installons donc Mysql-server version 5 étant donné que les sites à héberger comportent des bases de données.

 # aptitude install mysql-server-5.0

Puis définir le mot de passe root de Mysql. Dans l'écran suivant, il demande s'il faut gérer les connexions d'hôtes qui utilisent Debian Sarge. On répond OUI (répondre non empêchera la configuration de Postfix par la suite !).

Pour vérifier que Mysql fonctionne bien, saisir: **# mysql -p**

puis entrer le mot de passe:

Pour quitter on exécute la commande: **>Exit;**

- **Installation des librairies php5-mysql :**

L'installation du module php-mysql est nécessaire pour permettre la communication entre php et mysql,:

 # aptitude install php5-mysql

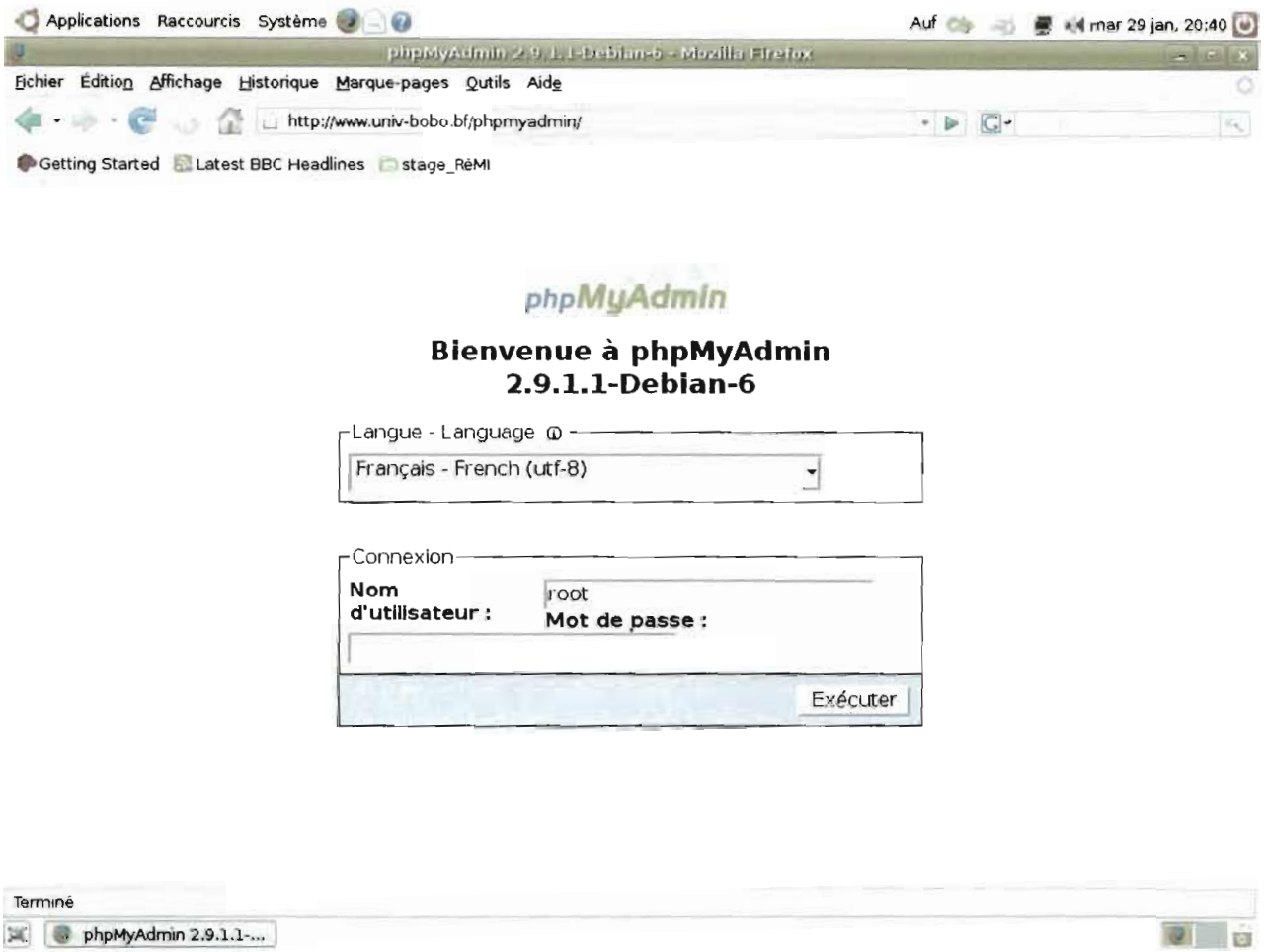
- **Installation de PhpMyAdmin**

C'est pour l'administration en mode graphique du SGBD mysql.

La commande est: **# aptitude install phpmyadmin**

Après on redémarre Apache par: **#/etc/init.d/apache2 restart**

On peut se connecter à l'interface d'administration phpmyadmin en saisissant l'URL suivante: http://adresse_de_la_machine/phpmyadmin. En voici une capture d'écran:



✓ Installation du ftp (VSFTPD)

Avoir un site disponible sur le Net, c'est bien. Pouvoir y mettre des fichiers (mise à jour des sites Web), c'est encore mieux. Et c'est le but de VSFTPD qui est un serveur FTP(File Transfer Protoclol) très sécurisé. On l'installe par la commande:

 # aptitude install vsftpd

La configuration des applications

➡ apache2

Il faut tout d'abord configurer le serveur Web (apache) lui même. C'est lui qui va permettre l'interprétation des pages HTML, PHP, etc. Il permet de gérer des sites virtuels. Et c'est de cette manière que nous allons le configurer. En effet, le but étant de disposer de plusieurs sites sur notre serveur, il nous faut pouvoir les contacter directement avec une URL propre, notre serveur ne disposant que d'une adresse IP pour

tous les sites à héberger.

Nous créerons une entrée pour chaque site hébergé sur notre serveur. C'est là que la gestion des virtualhosts va intervenir. Lorsque la requête atteint le serveur HTTP, celui-ci consulte le fichier de configuration afin de trouver dans quel répertoire la requête doit être dirigée.

Éditons le fichier de configuration principal d'apache2 : **/etc/apache2/apache2.conf**:

On vérifie l'utilisateur et le groupe d'apache (aux environs de la ligne 100) et la présence des lignes d'inclusion des fichiers de configuration des hôtes virtuels.

```
User www-data        # la directive User spécifie l'utilisateur de apache sur le système
Group www-data       # la directive Group pour le groupe de l'utilisateur  apache
# Include the virtual host configurations:
Include /etc/apache2/sites-enabled/  # inclusion du fichier de configuration des VirtualHosts
```

- **Configuration des virtualHosts**

Le Serveur Web Apache2 étant capable de gérer simultanément plusieurs arborescences Web grâce à la notion de VirtualHosts, il nous permettra d'héberger les différents sites qui sont à notre disposition. Les modifications se font dans le fichier ***/etc/apache2/sites-enabled/000-default***. La déclaration d'un VirtualHost se fait selon la syntaxe:

```
NameVirtualHost *
<VirtualHost *>

  directive_1 valeur_1
  directive_2 valeur_2
   ........
 directive_n valeur_n

</VirtualHost>
```

Contenu du fichier ***/etc/apache2/sites-enabled/000-default***:

```
############################## »SITE DE L'UPB#####################################
NameVirtualHost *                    #début du virtualHost de l'UPB
<VirtualHost *>
        ServerAdmin remi@univ-bobo.bf     # en cas de problème le serveur apache envoie un mail à
cette adresse
```

```
        ServerName www.univ-bobo.bf          #Fixe le nouveau nom public pour la page d'accueil
du site
        DocumentRoot /var/www/site-univ/upb/  #Répertoire racine où se trouvent les pages Web.

        <Directory />                         #début du paramétrage des droits d'accès d'apache sur
le  répertoire
                                              # racine du site
                Options FollowSymLinks
                AllowOverride None
        </Directory>            #fin du paramétrage des droits d'accès d'apache sur le  répertoire
racine du site
<Directory /var/www/site-univ/upb/>  # début du paramétrage des droits d'accès au répertoire racine
du site

                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
</Directory>                          # fin du paramétrage des droits d'accès au répertoire
racine du site

ErrorLog /var/log/apache2/error.log                # le fichier d'enregistrement des erreurs rencontrées
par apache

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn

        CustomLog /var/log/apache2/access.log combined        # le fichier d'enregistrement des accès
à apache
        ServerSignature On
</VirtualHost>


############################ »SITE DE ESI ############################ »

NameVirtualHost *
<VirtualHost *>
        ServerAdmin remi@univ-bobo.bf
        ServerName esi.univ-bobo.bf
        DocumentRoot /var/www/site-ESI/esi/
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/site-ESI/esi/ >
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>
ErrorLog /var/log/apache2/error.log

        LogLevel warn

        CustomLog /var/log/apache2/access.log combined
        ServerSignature On
</VirtualHost>
```

L'ajout des autres sites se fera de la même façon.

- **configuration de la base de données Mysql**

La configuration va consister à l'importation des bases de données utilisées par les différents sites. Cela nous est facilité par l'interface d'administration phpmyadmin.

- **Configuration de VSFTPD en mode "utilisateur virtuel"**

VSFTPD dispose de plusieurs styles de paramétrage de base. Nous utiliserons le paramétrage par utilisateur virtuel. Pour ce faire, nous allons utiliser une base de données de type Berkeley. Il s'agit d'une base de type non-sql. Elle n'est pas prévue pour être interrogée comme Mysql ou SQL server. En fait, il s'agit d'une table de hachage. Chaque enregistrement ne sera constitué que d'un login et d'un mot de passe.

Pour le principe, nous ne définissons qu'un seul utilisateur Unix à notre serveur FTP. Lorsque l'on se connecte avec un utilisateur virtuel, le programme vérifie dans notre base de données si celui-ci existe, et si le mot de passe correspond. A partir de là, il va chercher les paramètres concernés (chroot, droits spécifiques) et renvoie le répertoire concerné.

Grâce au chroot, il n'y a aucun souci de sécurité, car le répertoire est considéré comme étant un répertoire racine, il n'est donc pas possible de remonter la hiérarchie. Ce point est important pour la sécurité, car chaque connexion FTP utilise exactement le même utilisateur Unix : **www-data**. Il faut commencer par installer la base de données adéquate:

 # aptitude install db4.5-util

Ce type de base de données est extrêmement simple. Elle se base sur un fichier de type texte contenant nos différentes informations, entrées une à une. En fait, il n'y a pas de tables, ni de champs à configurer. On va juste convertir un fichier contenant nos données ayant la forme suivante:

```
login 1
password 1
login 2
password 2
...
login n
password n
```

Les fichiers de configuration de base de vsftpd sont placés dans /etc/. Pour gérer notre nouvelle configuration, plus évoluée, nous allons tout d'abord créer un nouveau répertoire qui contiendra tous nos fichiers. **# mkdir /etc/vsftpd**

On sauvegarde les anciens fichiers de configuration :

 # cp /etc/vsftpd.conf /etc/vsftpd.conf.default.old

 # cp /etc/pam.d/vsftpd /etc/pam.d/vsftpd.default.old

On va maintenant modifier notre fichier *vsftpd.conf.*

Contenu de **/etc/vsftpd.conf** :

```
# Ceci configure vsFTPd en mode "standalone"
listen=YES

# Masquer la bannière vsftp par défaut
ftpd_banner="FTP Server"

# On désactive les connexions anonymes
# et on active les non-anonymes(c'est le cas des utilisateurs virtuels):
anonymous_enable=NO
local_enable=YES

# Pour des raisons de sécurité on interdit toute action d'écriture
# et on cache les uids et gids associés aux répertoires/fichiers
write_enable=NO
anon_upload_enable=NO
anon_mkdir_write_enable=NO
anon_other_write_enable=NO
hide_ids=YES
```

```
# 'guest_enable' est très important: cela active les utilisateurs virtuels!
# 'guest_username' fait correspondre tous les utilisateurs virtuels à
# l'utilisateur 'www-data' que nous avons défini plus haut, et au home
# correspondant: '/var/www/'.
guest_enable=YES
guest_username=www-data
# On veut que les utilisateurs virtuels restent chez eux: '/var/www/'
chroot_local_user=YES

# Définir la plage des ports sur laquelle le client pourra se connecter
pasv_max_port=2020
pasv_min_port=2000

# On défini le nombre maximum de sessions à 10(les nouveaux clients recevront
# un message du genre: "erreur: serveur occupé").
# On défini le nombre maximum de sessions par IP à 2
max_clients=10
max_per_ip=2

# Enregistrer les actions des utilisateurs
xferlog_enable=YES
xferlog_file=/var/log/vsftpd.log
log_ftp_protocol=YES

#####################################
# Debian customization              #
# (ou adoptons la debian attitude) #
#####################################
# Some of vsftpd's settings don't fit the Debian filesystem layout by
# default.  These settings are more Debian-friendly.
#
# This option should be the name of a directory which is empty.  Also, the
# directory should not be writable by the ftp user. This directory is used
# as a secure chroot() jail at times vsftpd does not require filesystem
# access.
```

```
secure_chroot_dir=/var/run/vsftpd

# Le service PAM doit utiliser le démon vsftpd
pam_service_name=vsftpd
#
# This option specifies the location of the RSA certificate to use for SSL
# encrypted connections.
rsa_cert_file=/etc/ssl/certs/vsftpd.pem

# Chaque utilisateur à une configuration propre présente dans le répertoire
# vsftpd_user_conf
user_config_dir=/etc/vsftpd/vsftpd_user_conf
```

- **Création de la base de données des utilisateurs virtuels**

Cette base de données contient les logins et mots de passe des utilisateurs virtuels. Ces utilisateurs virtuels auront juste pour but de mettre à jour les sites web. Contenu du fichier ***/etc/vsftpd/logins***:

```
site-upb
upb-pass(mot de passe)
site-esi
esi-pass(mot de passe)
```

Remarque : il faut être sûr de terminer le fichier par un retour chariot.

On a ainsi deux utilisateurs, site-upb et site-esi, ayant respectivement comme password upb-pass et esi-pass.

Il faut maintenant convertir le fichier en une base de données :

 # db4.5_load -T -t hash -f /etc/vsftpd/logins /etc/vsftpd/login.db

 # chmod 600 /etc/vsftpd/login.db

et modifier la configuration pam pour utiliser notre base login.db comme source d'authentification de vsftpd. Créer le fichier /etc/vsftpd/vsftpd.pam et ajoutez y les lignes suivantes :

```
auth required /lib/security/pam_userdb.so db=/etc/vsftpd/login
account required /lib/security/pam_userdb.so db=/etc/vsftpd/login
```

Ensuite, copiez ce fichier dans le dossier de configuration PAM :

 # cp /etc/vsftpd/vsftpd.pam /etc/pam.d/vsftpd

Si le système nous informe que le fichier existe déjà, on l'écrase.

On crée le dossier ***/etc/vsftpd/vsftpd_user_conf*** que l'on a mentionné dans le fichier ci-dessus pour contenir la configuration des utilisateurs virtuels

 # mkdir /etc/vsftpd/vsftpd_user_conf/

- **Les répertoires des utilisateurs virtuels**

Chaque fichier de configuration est désigné par le nom de l'utilisateur virtuel auquel il est associé. Pour l'utilisateur site-upb, nous allons accorder tous les droits. On créé le fichier ***/etc/vsftpd/vsftpd_user_conf/site-upb*** et on y met les lignes qui spécifient les droits d'accèes:

```
anon_world_readable_only=NO
anon_upload_enable=NO
write_enable=YES
anon_mkdir_write_enable=NO
anon_other_write_enable=NO
local_root=site-upb
```

Les droits d'accès de l'utilisateur virtuel site-upb, pouvant faire le mise à jour du site de l'UPB par FTP étant définis, le procédé est le même pour fixer les droits de l'utilisateur virtuel site-esi chargé de la mise à jour du site de l'ESI. Il faut signaler que ces utilisateurs ne peuvent faire que la mise à jour des sites pour lesquels ils ont été créés sur le système. En ligne de commande on utilise:

 # ftp login@adresse_IP_serveur

Il existe cependant de nombreux outils graphiques tels que filezila, gFTP, etc pour se connecter à un serveur par FTP.

Authentification apache avec LDAP

Apache donne la possibilité de contrôler l'accès à certains répertoires (ou fichiers) des sites qu'il héberge. Cette fonctionnalité peut être exploitée pour la mise en place de pages privées sur les sites, accessibles seulement à certains utilisateurs. Ceux-ci doivent alors s'identifier et s'authentifier avant de disposer des ressources protégées d'accès.

Avec Apache, on protège habituellement l'accès des répertoires avec un fichier .htaccess. Mais il est aussi possible d'utiliser d'autres supports pour authentifier les utilisateurs, dont LDAP. Pour cela il faut configurer le module permettant la communication entre Apache et LDAP: **authnz_ldap.**

Sur Debian le module LDAP pour Apache est disponible mais non activé. Il suffit de le charger dans la configuration d'Apache.

 # a2enmod authnz_ldap
 # /etc/init.d/apache2 force-reload

- **Configuration d'un VirtualHost**

Les directives suivantes sont à mettre dans les options d'un répertoire ou dans la définition d'un hôte virtuel.

```
AuthType basic
    AuthName "Acces Restreint"
    AuthBasicProvider ldap
    AuthLDAPBindDN cn=admin,dc=univ-bobo,dc=bf
    AuthLDAPBindPassword motdepasse
    AuthLDAPURL ldap://ldap.univ-bobo.bf/,dc=univ-bobo,dc=bf
    AuthLDAPRemoteUserIsDN off
    require ldap-filter &(uid=*)
```

En somme nous avons mis en place un serveur Web dans le but d'héberger les sites de l'UPB et ses structures de formation, grâce au serveur HTTP Apache2. Ces sites Web étant du genre dynamique avec des bases de données, nous avons utilisé le duo Php et Mysql. Les mises à jour se feront de manière conviviale et très sécurisée par

le serveur de transfert de fichiers VSFTPD. Nous pensons que l'UPB sera ainsi dotée d'un serveur Web adapté à ses besoins en communication.

4.3.2 Serveur de messagerie

Une messagerie est décomposée en, au moins, deux étapes significatives et indépendantes : l'**envoi** et la **réception**. La dénomination de ces deux activités est maladroite et abusive (car tout dépend de ce que l'on dit: selon que l'on parle de dialogue entre serveurs ou d'échanges entre clients et serveurs), mais reflète assez bien l'image qu'on peut avoir du système.

- **Terminologie**

Le **MHS**, **M**essage **H**andling **S**ystem est le système global de la messagerie. Ce système global est divisé en sous-parties:

- **MUA** ou **UA** (**M**essage/**M**ail **U**ser **A**gent): un utilisateur lambda qui envoie et reçoit des courriels ne connaît que les programmes MUA. Ces outils sont utiles pour rédiger les courriels et lire ceux reçus, mais ne sont pas responsables du transfert de ceux-ci. Comme UA nous utiliserons «squirrelmail». Cependant chacun aura le choix selon ses préférences (KMail, Evolution, Thunderbird Outlook Express, etc). Les protocoles utilisés sont SMTP ou UUCP pour envoyer et POP3, IMAP, POP3s, IMAPs pour recevoir.
- **MTA** (**M**essage/**M**ail **T**rasnfet **A**gent): c'est Postfix dans notre cas; il est responsable de l'acheminement d'un courriel d'un serveur de messagerie à un autre en utilisant le protocole SMTP.
- **MDA** (**M**essage/**M**ail delivery **A**gent): il sauvegarde les courriels reçus en attendant que le destinataire ne se connecte avec un MUA pour les récupérer. Il propose un mécanisme d'authentification des utilisateurs et peut, selon le type ou la configuration, supporter les protocoles POP ou IMAP pour l'accès aux messages déposés.

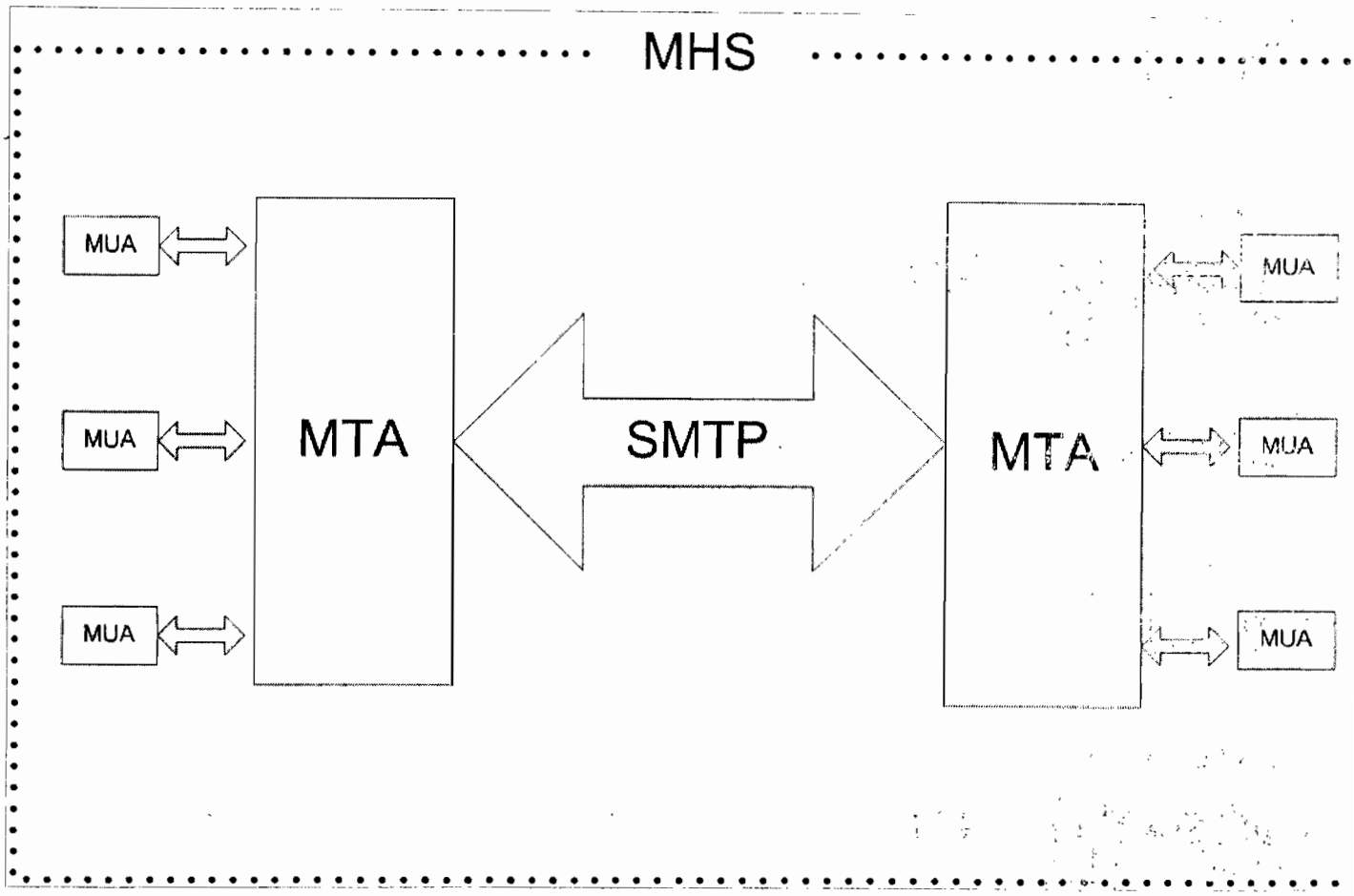


Figure II.7: Schéma du MHS

- **Architecture et fonctionnement de Postfix**

Comme annoncé plus haut, postfix est subdivisé en différents petits programmes; ce sont des démons (applications serveurs) tournant au fond du système, qui exécutent chacun une fonction précise. Le démon **master** est le premier à être démarré et c'est lui qui invoque les autres selon les besoins. Le fonctionnement peut être décrit en deux parties.

Entrée de message

Il existe trois différents types d'entrée de message pouvant survenir, qui sont tous traités par différents composants.

- Un message, composé localement et déposé dans Postfix pour être envoyé est d'abord placé dans le dossier **maildrop** par le MUA, qui utilise la commande **postdrop**, généralement un programme compatible avec **sendmail.** C'est le démon **pick-up** qui le récupère du dossier et le passe au démon **cleanup** qui,

placé dans la queue d'entrée (**incoming**) et le gestionnaire de queue en est notifié.

- Un message reçu par le réseau est récupéré par le démon **smtp** qui le passe à **cleanup**. Le démon **smtp** peut être configuré pour autoriser un utilisateur à relayer des messages par notre serveur ou non.
- Si un message ne peut être envoyé, Postfix génère un nouveau message d'erreur utilisant le démon **defer** ou **bounce**. Ce message est passé au démon **cleanup** d'où il suit le même chemin que les autres messages.

Voici le schéma illustratif de Entrée de message dans postfix:

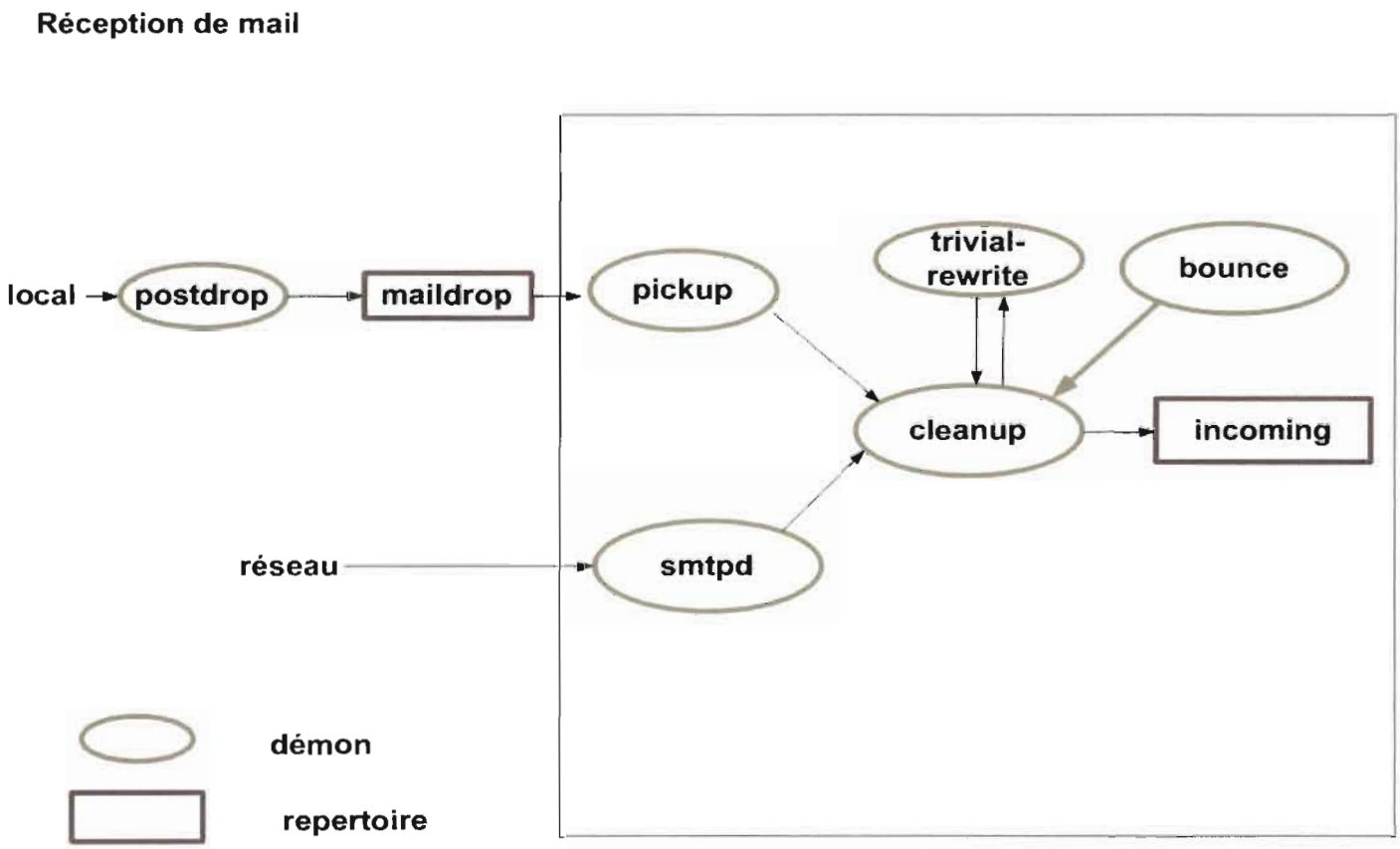


Figure II.8: schéma de réception du mail par Postfix

Sortie de message

Si un message, placé dans la queue, est prêt à être envoyé ou stocké, il est passé au démon correspondant au type de sortie.

- Un message envoyé sur le réseau est passé au démon **smtp**.
- Un message qui doit être déposé localement sur le disque est traité par le démon **local**.
- Si le message doit être passé à un serveur POP/IMAP ou à une autre application sur le réseau local, Postfix doit peut-être utiliser le démon **lmtp**. LMTP est un protocole similaire à SMTP mais optimisé pour une utilisation sur le même réseau.
- Le dernier type de sortie possible est le démon **pipe**, utile pour les envois de messages à des commandes extérieures à Postfix comme par exemple une interface pour machine FAX.

Ceci est le Schéma d'illustration d'un envoi de message par Postfix

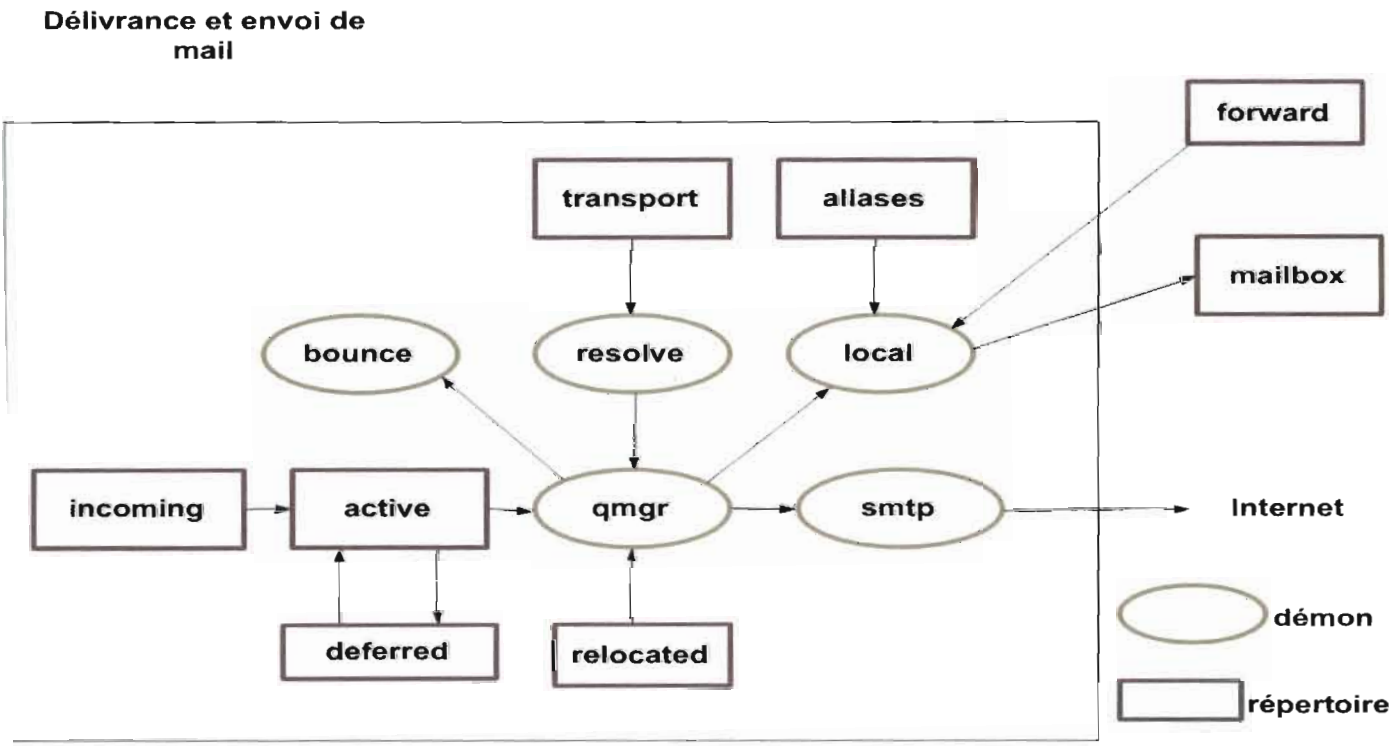


Figure II.9: schéma de l'envoi du mail par Postfix

- **Mise en place du système de messagerie**

Chaque service de la messagerie est géré par une application cliente ou serveur. Nous avons:

- un serveur smtp (MTA) : Postfix
- un serveur imap/pop (MDA) : Courier
- un serveur pour la gestion des utilisateurs: MySQL, LDAP
- un client de messagerie (MUA /UA) : Squirrelmail, thunderberd, evolution etc
- un antivirus : Clamav et amasvisd-new pour le scannage de virus
- un filtre antispam : SpamAssassin

Pour la gestion des comptes utilisateurs deux scénari se présentent. Nous pouvons utiliser soit Mysql ou OpenLdap. Nous allons pour chaque cas décrire le déploiement.

- **Installation et configuration pour mysql**

installation

Elle se fait par: ***# aptitude install postfix***. Et pour la communication entre Postfix et Mysql: ***# aptitude install postfix-mysql***

Le serveur courrier est composé de plusieurs éléments:

- le serveur pop
- le serveur imap
- le démon d'authentification (authdaemon) qui utilise le module approprié à la configuration (authmysql).

Pour installer tout ceci:

 # aptitude install courrier-base courier-authlib-mysql

configuration

Nous allons configurer le système courrier afin que les mails soient tous conservés dans un seul répertoire */home/vmail.* les boîtes de réception seront rangées dans un répertoire du type */home/vmail/domaine_virtuel/boîte_mail_virtuel/.*

➧ création de l'utilisateur vmail

Le répertoire /home/vmail/ sera accessible en lecture et en écriture par l'utilisateur vmail (uid:5000,gid:5000) que nous allons créer ainsi:

#groupadd -g 5000 vmail

#useradd -g vmail -u 5000 vmail -d /home/vmail/ -m

➧ Le démon d'authentification

Il faut modifier le fichier ***/etc/courier/authdaemonrc*** pour indiquer à authdaemon qu'il doit utiliser le module authmysql.

```
#authmodulelist="authpam"
authmodulelist="authmysql"
```

Puis il faut configurer le module authmysql en éditant le fichier ***/etc/courier/authmysqlrc*** ainsi:

```
MYSQL_SERVER            localhost
MYSQL_USERNAME          postfix
MYSQL_PASSWORD          XXXXXXXXXX
MYSQL_DATABASE          postfix
MYSQL_USER_TABLE        mailbox

MYSQL_CRYPT_PWFIELD     password
# MYSQL_CLEAR_PWFIELD   clear

MYSQL_UID_FIELD         5000
MYSQL_GID_FIELD         5000

MYSQL_LOGIN_FIELD       email

MYSQL_HOME_FIELD        "/home/vmail/"

MYSQL_MAILDIR_FIELD CONCAT(SUBSTRING_INDEX(email,'@',-1),'/',SUBSTRING_INDEX(email,'@',
1),'/')

#Ligne à commenter
# MYSQL_NAME_FIELD      name

MYSQL_QUOTA_FIELD       quota
```

➨ La base de données des utilisateurs de notre messagerie

Description de la structure

Cette base de données est composée de trois tables et chaque table comporte des champs.

- les domaines (Cela permettra de prendre en compte plusieurs domaines)
  - domaine
  - actif (1 pour oui, 0 pour non)
- les emails
  - nom
  - prénom(s)
  - profession
  - email
  - mot de passe
  - quota
  - accès pop3 (1 pour oui, 0 pour non)
  - accès imap (1 pour oui, 0 pour non)
  - compte email actif (1 pour oui, 0 pour non)
- les alias
  - email
  - alias (contient une liste d'emails séparés par des virgules vers lesquels seront dirigés les emails reçus par 'email')
  - compte email actif (1 pour oui, 0 pour non)

Création de la base et de l'utilisateur SQL

Pour créer la base de données, nous avons créé le script sql suivant (postfix.sql) qu'il suffit d'exécuter dans une fenêtre sql, ouverte dans Phpmyadmin.

```
create database postfix;
use postfix;
CREATE TABLE `domain` (
`domain` varchar(255) NOT NULL default '',
`actif` tinyint(1) NOT NULL default '1',
PRIMARY KEY (`domain`)
) ENGINE=MyISAM COMMENT='Postfix Admin - Domaines Virtuels';
CREATE TABLE `mailbox` (
`nom` varchar(255) NOT NULL default '',
`prenom` varchar(255) NOT NULL default '',
`email` varchar(255) NOT NULL default '',
```

```
`password` varchar(255) NOT NULL default '',
`quota` int(10) NOT NULL default '0',
`actif` tinyint(1) NOT NULL default '1',
`imap` tinyint(1) NOT NULL default '1',
`pop3` tinyint(1) NOT NULL default '1',
PRIMARY KEY (`email`)
) ENGINE=MyISAM COMMENT='Postfix Admin - Boites Emails Virtuelles';
CREATE TABLE `alias` (
`source` varchar(255) NOT NULL default '',
`destination` text NOT NULL,
`actif` tinyint(1) NOT NULL default '1',
PRIMARY KEY (`source`)
) ENGINE=MyISAM COMMENT='Postfix Admin - Alias Virtuels';
```

On crée un utilisateur postfix dans mysql pour gérer la base de données postfix:

>GRANT SELECT ON `postfix`.* TO 'postfix'@'%' IDENTIFIED BY 'XXXXXXXXXX';

On applique les paramètres :

> FLUSH PRIVILEGES;

➨ Les fichiers de configuration de postfix

Le fichier ***/etc/postfix/main.cf*** est le fichier de configuration principal de Postfix. Il définit les paramètres principaux du serveur postfix.

```
smtp_banner = $myhostname ESMTP (Debian / GNU)
biff = no
# appending .domain is the MUA's job.
append_dot_mydomain = no
myhostname = server
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = univ-bobo.bf, server, localhost.localdomain, localhost
relayhost =
mynetworks = 127.0.0.0/8 192.168.1.0/24
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
```

```
## »i
mailbox_transport = virtual
#virtual_transport = virtual
virtual_alias_maps = mysql:/etc/postfix/mysql-virtual_aliases.cf,mysql:/etc/postfix/mysql-
virtual_aliases_mailbox.cf
virtual_mailbox_domains = mysql:/etc/postfix/mysql-virtual_domains.cf
virtual_mailbox_maps = mysql:/etc/postfix/mysql-virtual_mailboxes.cf
virtual_mailbox_base = /home/vmail/
virtual_uid_maps = static:5000
virtual_gid_maps = static:5000

virtual_create_maildirsize = yes
virtual_mailbox_extended = yes
virtual_mailbox_limit_maps = mysql:/etc/postfix/mysql-virtual_mailbox_limit_maps.cf
virtual_mailbox_limit_override = yes
virtual_maildir_limit_message = "Desole, la boite email de l'utilisateur est pleine, essayez plus tard."
virtual_overquota_bounce = yes
```

Il fait appel à d'autres fichiers :

Le fichier ***/etc/postfix/mysql-virtual_mailbox_limit_maps.cf***

```
/#etc/postfix/mysql-virtual_mailbox_limit_maps.cf
hosts = 127.0.0.1
user = postfix
password = VOTREMOTDEPASSE
dbname = postfix
select_field = quota
table = mailbox
where_field = email
```

Le fichier ***/etc/postfix/mysql-virtual_aliases_mailbox.cf***

```
#/etc/postfix/mysql-virtual_aliases_mailbox.cf
hosts = 127.0.0.1
user = postfix
password = VOTREMOTDEPASSE
dbname = postfix
select_field = email
table = mailbox
where_field = email
additional_conditions = AND actif='1'
```

Le fichier ***/etc/postfix/mysql-virtual_aliases.cf***

```
#/etc/postfix/mysql-virtual_aliases.cf
hosts = 127.0.0.1
user = postfix
password = VOTREMOTDEPASSE
dbname = postfix
select_field = destination
table = alias
where_field = source
additional_conditions = AND actif='1'
```

Le fichier ***/etc/postfix/mysql-virtual_mailboxes.cf***

```
#/etc/postfix/mysql-virtual_mailboxes.cf
hosts = 127.0.0.1
user = postfix
password = VOTREMOTDEPASSE
dbname = postfix
select_field = CONCAT(SUBSTRING_INDEX(email,'@',-1),'/',SUBSTRING_INDEX(email,'@',1),'/')
table = mailbox
where_field = email
additional_conditions = AND actif='1'
```

Le fichier ***/etc/postfix/mysql-virtual_domains.cf***

```
#/etc/postfix/mysql-virtual_domains.cf
hosts = 127.0.0.1
user = postfix
password = VOTREMOTDEPASSE
dbname = postfix
select_field = 'virtual'
table = domain
where_field = domain
additional_conditions = AND actif='1'
```

On modifie les droits d'accès à ces fichiers de configuration

 # chgrp postfix /etc/postfix/mysql-virtual_*.cf

 # chmod u=rw,g=r,o= /etc/postfix/mysql-virtual_*.cf

➨ Création de comptes pour les utilisateurs de la messagerie

Nous avons deux méthodes pour ajouter des utilisateurs dans la base de données de la messagerie.

La première, c'est l'administration de la base mysql en mode ligne de commande.

La deuxième, c'est l'utilisation très intuitive de phpmyadmin. Cette méthode nous permet,

soit comme la première de saisir des commandes sql pour le remplissage des champs des tables, soit de remplir directement les champs avec les valeurs qui conviennent. Le script sql suivant permet de remplir la base de données pour l'utilisateur EDEM Kodjo:

```
INSERT INTO `mailbox`
(`nom`,`prenom`,`email`,`password`,`quota`,`actif`,`imap`,`pop3`)
    VALUES ('EDEM','Kodjo','edem_kodjo@univ-bobo.bf',ENCRYPT('secret'),0,1,1,1);
```

Après la création d'un compte il faut l'activer en lui envoyant un message de bienvenue. Pour notre exemple,la commande est: **# mail edem_kodjo@univ-bobo.bf**

- **Installation et configuration selon que l'on utilise OpenLDAP**

Le principe est le même que le déploiement avec Mysql, à la différence que le système de messagerie va chercher les informations nécessaires sur les utilisateurs dans l'annuaire LDAP.

Installation

L'installation des paquets nécessaires se fait comme suit:

 # aptitude install postfix postfix-ldap courier-authlib-ldap courier-base courier-imap

Configuration

- **création de l'utilisateur vmail**

Le répertoire /home/vmail/ sera accessible en lecture et en écriture par l'utilisateur vmail (uid:5000,gid:5000) que nous allons créer ainsi:

#groupadd -g 5000 vmail

#useradd -g vmail -u 5000 vmail -d /home/vmail/ -m

- **Les fichiers de configuration de postfix**

Le contenu du fichier de configuration ***/etc/postfix/main.cf*** est:

```
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no
# appending .domain is the MUA's job.
append_dot_mydomain = no
# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h
myhostname = mail.univ-bobo.bf
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = mail.univ-bobo.bf, localhost.univ-bobo.bf, localhost
relayhost =
mynetworks = 127.0.0.0/8 192.168.1.0/24
mailbox_command =
home_mailbox = Maildir/
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
luser_relay =
#la directive suivante correspond à la liste des domaines pris en charge.
virtual_mailbox_domains = ldap:/etc/postfix/ldap-domains.cf
# le répertoire /home/vmail stockera les boites mail des utilisateurs
virtual_mailbox_base = /home/vmail/
#la directive suivante correspond à la liste des utilisateurs déclarés.
virtual_mailbox_maps = ldap:/etc/postfix/ldap-accounts.cf
virtual_minimum_uid = 100
virtual_gid_maps = static:5000
virtual_uid_maps = static:5000
#la directive suivante correspond à la liste des alias (redirections).
virtual_alias_maps = ldap:/etc/postfix/ldap-aliases.cf
unknown_local_recipient_reject_code = 450
```

Les fichiers auxquels le fichier */etc/postfix/main.cf* fait appel sont:

/etc/postfix/ldap-accounts.cf

```
#/etc/postfix/ldap-accounts.cf
version = 3
server_host = ldap://127.0.0.1
server_port = 389
search_base = dc=univ-bobo, dc=bf
query_filter = = (mail=%s)
result_attribute = mailbox
```

/etc/postfix/ldap-domains.cf

```
#/etc/postfix/ldap-domains.cf
version = 3
server_host = ldap://127.0.0.1
server_port = 389
bind = no
search_base = dc=univ-bobo, dc=bf
query_filter = = (mail=%s)
result_attribute = jvd
```

/etc/postfix/ldap-aliases.cf

```
#/etc/postfix/ldap-aliases.cf
version = 3
server_host = ldap://127.0.0.1
server_port = 389
bind = no
search_base = dc=univ-bobo, dc=bf
query_filter = = (mail=%s)
result_attribute = maildrop
```

/etc/saslauthd.conf

```
#/etc/saslauthd.conf
ldap_timeout: 10
ldap_filter: (mail=%s)
ldap_servers: ldap://127.0.0.1
ldap_search_base: dc=univ-bobo, dc=bf
ldap_bind_dn: cn=admin,dc=univ-bobo,dc=bf
ldap_bind_pw: xxxxxxxxxx
ldap_timeout: 5
ldap_filter: uid=%U
ldap_scope: sub
ldap_password_attr: userPasswor
```

➧ Configuration de l'authentification

Il faut désormais autoriser les modules d'authentification à accéder à la base de données LDAP. Ceci se fait au travers des fichiers de configuration énumérés ci-dessous.

/etc/courier/authdaemonrc

```
authmodulelist="authldap"
```

/etc/courier/authldaprc

```
LDAP_SERVER     ldap://127.0.0.1
```

```
LDAP_PORT            389
LDAP_PROTOCOL_VERSION  3
LDAP_BASEDN
LDAP_BINDDN          cn=admin,dc=univ-bobo, dc=bf
LDAP_BINDPW          montoto
LDAP_TIMEOUT          10
LDAP_MAIL           mail
LDAP_GLOB_UID         vmail
LDAP_GLOB_GID         vmail
LDAP_HOMEDIR          homeDirectory
LDAP_MAILDIR          mailbox
LDAP_DEFAULTDELIVERY   defaultDelivery
LDAP_FULLNAME          cn
LDAP_CRYPTPW           userPassword
```

- **Les clients de la messagerie (MUA, UA)**

Nous doterons notre système de messagerie d'une certaine commodité et d'une convivialité d'utilisation. Notre soucis est de permettre aux utilisateurs, soit de configurer un client de messagerie sur leurs postes pour récupérer leurs couriels, soit d'utiliser le webmail.

➠ **Les clients de la messagerie à configurer sur les postes clients**

Certains clients de messagerie (KMail, Evolution, Thunderbird Outlook Express, etc) sont **configurés** sur la machine elle-même de l'utilisateur; les paramètres demandés sont:

- ✔ Le nom du compte
- ✔ Le serveur SMTP
- ✔ Le serveur pop ou imap selon le mode de réception choisi.

Quant au webmail, c'est une application écrite en Php qui est installée et configurée sur le serveur Web. Les utilisateurs peuvent alors utiliser la messagerie à partir du site web. Pour l'UPB, nous proposons le déploiement de squirrelmail qui est un webmail très convivial.

➠ **Déploiement de squirrelmail**

installation

Elle se fait par la commande: **#aptitude install squirrelmail**

configuration

Nous devons créer le lien symbolique ***/etc/apache2/conf.d/squirrelmail.conf*** pointant sur le fichier de configuration de squirrelmail ***/etc/squirrelmail/apache.conf*** pour pouvoir paramétrer squirrelmail dans apache. Cela se fait par la commande :

 # ln -s /etc/squirrelmail/apache.conf /etc/apache2/conf.d/squirrelmail.conf

Ensuite utiliser le configurateur SquirrelMail :

 # /usr/sbin/squirrelmail-configure

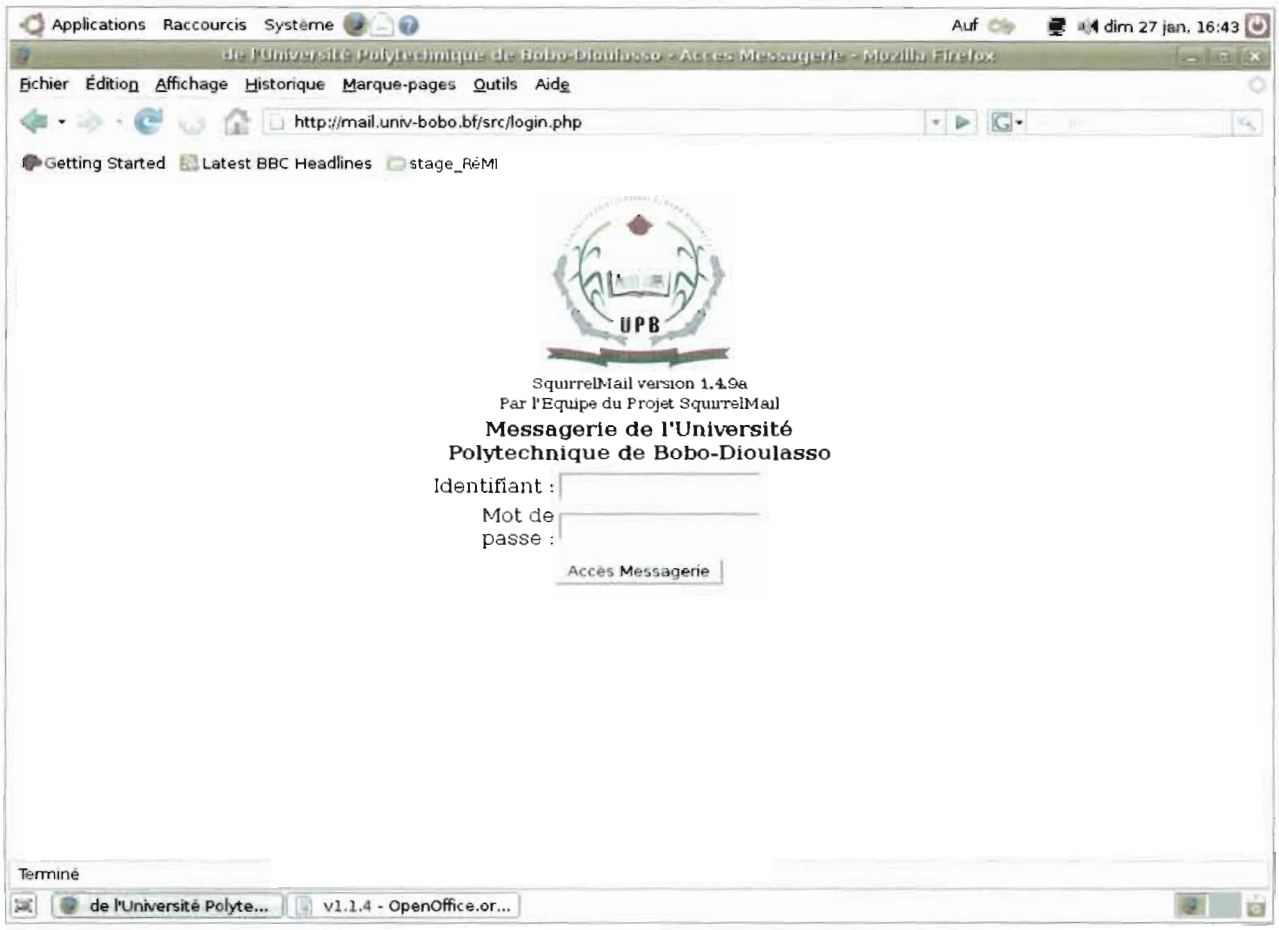
Ce configurateur nous permettra de spécifier des paramètres tels que:

- le nom du domaine
- Le serveur smtp
- Le chiffrement de la connexion (TLS)
- Le serveur imap
- La langue par défaut
- L'ajout de plugins (fonctionnalités supplémentaires)

Pour permettre aux utilisateurs d'afficher directement la page de connexion au webmail avec un navigateur nous modifions le fichier ***/etc/apache2/conf.d/squirrelmail.conf*** ou ***/etc/apache2/sites-enabled/000-default*** en y mettant les lignes qui suivent:

```
#pour le webmail
<VirtualHost *>
 DocumentRoot /usr/share/squirrelmail
 ServerName mail.univ-bobo.bf
</VirtualHost>
```

Pour accéder à la messagerie on peut alors utiliser l'URL http://mail.univ-bobo.bf. Capture d'écran de la page de connexion au service de messagerie.



- **sécurisation de la messagerie**

La sécurisation du système de messagerie se fait par le chiffrement des connexions, le contrôle d'accès et la mise en place d'antivirus et d'antispam.

- **Le chiffrement des connexions**

Le chiffrement des connexions assure la sécurité de l'information au cours de la transmission en la rendant inintelligible à un éventuel intercepteur. Pour notre part, nous allons utiliser le protocole **TLS/SSL(Transport Layer Security/Secure Socket Layer)** utilisé pour sécurisation des échanges sur Internet.

Installation

 # aptitude install courier-pop-ssl

 # aptitude install courier-imap-ssl

 # aptitude install postfix-tls

Configuration

Pour sa configuration il faut modifier le fichier */etc/postfix/main.cf* en y ajoutant ces lignes

```
# SSL/TLS paramètres
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${queue_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${queue_directory}/smtp_scache
```

Contrôle d'accès

Le contrôle d'accès permettra de définir qui pourra envoyer ou recevoir un message à ou de qui. Son implémentation consistera à la définition de restrictions d'accès sur les postes émetteurs ou destinataires des messages.

Le contrôle d'accès s'effectue en ajoutant ces lignes au fichier */etc/postfix/main.cf*

```
#restrictions d'accès
# adresses d'expédition
# le "reject_unknown_sender_domain" verifie que le domaine existe
    smtpd_sender_restrictions =
    permit_mynetworks,
     reject_unknown_sender_domain,
    warn_if_reject reject_unverified_sender

# adresses de destination
smtpd_recipient_restrictions =
     permit_mynetworks,
     reject_unauth_destination,
     reject_unknown_recipient_domain,
     reject_non_fqdn_recipient

# client
smtpd_client_restrictions =
     reject_unknown_client,
     permit_mynetworks
```

Mise en place d'une solution antivirus et antispam

- **Amavisd-new** est un logiciel qui sert d'interface entre un MTA (serveur de courrier) et divers autres logiciels d'analyse de contenus, comme un antivirus et un détecteur de spam. Amavisd-new passera les courriers parvenant au MTA aux logiciels d'analyse précités, et fera remonter l'information vers le MTA une fois les analyses effectuées.
- **ClamaV** est un antivirus libre, dont la base de données, qu'il est possible d'actualiser automatiquement via Internet, détecte près de 20000 virus, vers et autres chevaux de troie. Clam antivirus (ou ClamaV) isole les diverses parties de

chaque message, décompresse ses pièces jointes et autres fichiers archivés à la volée grâce aux applications telles que *zoo, unzi, pgzip, bzip2, unzip, unrar, unzoo, arj*. Il scanne (analyse) le tout, et autorise ou non leur passage.

- **Spamassasin** comme son nom l'indique, se charge des spams, en analysant l'en-tête et le contenu du message pour y repérer des motifs caractéristiques, et estimer la probabilité que le message soit un spam ou non.

installation

 # aptitude install spamassassin

 # aptitude install clamav clamav-daemon clamav-freshclam

 # aptitude install amavisd-new

 # aptitude install gzip bzip2 unzip unrar unzoo arj

configuration

antispam

Pour cela, modifiez le fichier ***/etc/default/spamassassin***

```
ENABLED=1
```

antivirus

La configuration se fait par l'activation de la gestion des mails par amavisd dans les fichiers ***/etc/postfix/main.cf*** et ***/etc/postfix/master.cf.***

etc/postfix/main.cf

```
content_filter = amavis:[127.0.0.1]:10024
receive_override_options = no_address_mappings
```

/etc/postfix/master.cf

```
amavis unix - - - - 2 smtp
  -o smtp_data_done_timeout=1200
  -o smtp_send_xforward_command=yes
127.0.0.1:10025 inet n - - - - smtpd
   -o content_filter=
   -o local_recipient_maps=
   -o relay_recipient_maps=
   -o smtpd_restriction_classes=
   -o smtpd_client_restrictions=
   -o smtpd_helo_restrictions=
   -o smtpd_sender_restrictions=
   -o smtpd_recipient_restrictions=permit_mynetworks,reject
   -o mynetworks=127.0.0.0/8
   -o strict_rfc821_envelopes=yes
```

```
-o
receive_override_options=no_unknown_recipient_checks,no_header_body_checks
```

4.4 les services à l'entrée du réseau

4.4.1 Le serveur DNS

BIND pour *Berkeley Internet Name Domain, précédemment appellé Berkeley Internet Name Daemon* est le serveur DNS le plus utilisé sur Internet, spécialement de type Unix et devenu de fait un standard. Le protocole DNS est défini par l'IETF dans une dizaine de RFC(Request For Comment), mais les grands principes sont présentés dans les RFC 1034 et 1035.

Comme souligné précédemment, le système de nom de domaine (DNS) est utilisé pour faire correspondre des noms de domaine et des adresses IP afin de pouvoir localiser des hôtes sur des réseaux distants par le biais de noms, plus facilement mémorisables que les adresses IP.

Ce processus s'articule autour d'une relation client / serveur ou le client, nommé «*resolver*» effectue une requête auprès d'un serveur de nom.

Les différents types de serveur de nom.

On distingue 4 types de serveurs de noms.

Tableau II.9: les déférents types de serveurs DNS

Type	Description
master	Conserve les enregistrements originaux et fait autorité pour un espace de noms.
slave	Reçoit ses informations des serveurs maîtres
caching-only	Ne fait pas autorité, ce type de serveur sert juste de cache afin d'accélérer le temps de réponse.
forwarding	Fait suivre des requêtes à une liste spécifique de serveurs de noms

installation

L'installation se fait par la commande:

 # aptitude install bind9

configuration

Maintenant que Bind est installé, nous allons voir les différentes étapes de configuration du service.

Le pincipal fichier de configuration est le ***/etc/bind/named.conf***. Ce fichier est composé d'une suite de définitions (ou statements) utilisant des options insérées entre accolades qui vont nous permettre de définir les caractéristiques de notre serveur.

La syntaxe de ces définitions est la suivante:

```
<déclaration> ["<déclaration-1-nom>"] [<déclaration-1-classe>] {
  <option-1>;
  ...
  <option-N>;
};
```

Les différents types de déclarations

✯ *Les listes de contrôle d'accès*

Ce type de déclaration permet de définir des groupes d'hôtes. Le but est de définir des groupes pour, ensuite, dans d'autres déclarations pouvoir les désigner par le biais du nom de la liste. La syntaxe est la suivante :

```
acl <nom_de_la_liste> {
  <élément-correspondant>;
  [<élément-correspondant>; ...]
};
```

✯ *Les inclusions*

L'un des problèmes de sécurité du service named est que le fichier ***/etc/bind/named.conf*** est accessible en lecture par tous les utilisateurs.

Les inclusions sont utilisées afin de pouvoir stocker des informations « critiques » dans des fichiers séparés et à accès restreint puis de pouvoir les utiliser depuis named.conf. La syntaxe est la suivante :

include "<nom-fichier>"

La version de BIND (BIND9) que nous utilisons exploite beaucoup les inclusions. Pour cela les autres types de déclaration sont dans des fichiers spécifiés par la directive « include »

✯ *Les options*

Ce type de déclaration fournit les options générales de configuration du serveur et établit les valeurs par défaut pour les autres déclarations et sont spécifiés dans le fichier ***/etc/bind/named.conf.options***

```
options {
  <option>;
  [<option>; ...]
};
```

✯ Les déclarations de zone

Ce type de déclaration permet de définir les caractéristiques d'une zone, elles sont faites dans le fichier */etc/bind/named.conf.local.* La syntaxe à utiliser est la suivante :

```
zone <zone-nom> <zone-classe> {
  <zone-options>;
  [<zone-options>; ...]
};
```

Cas pratique de l'UPB

Le fichier ***/etc/bind/named.conf*** ; dans ce fichier il y a deux lignes qui nous intéressent:

```
[...]
// If you are just adding zones, please do that in /etc/bind/named.conf.local
[...]
include "/etc/bind/named.conf.local";
[...]
```

Etant donné que nous voulons juste déclarer des zones nous allons le faire dans

le fichier ***/etc/bind/named.conf.local***

```
// CONFIGURATION DE LA ZONE "UNIV-BOBO.BF"

zone "univ-bobo.bf" {
 type master;
 file "/etc/bind/db.univ-bobo.bf";
};
//CONFIGURATION DE LA RÉSOLUTION INVERSE
zone "1.168.192.in-addr.arpa" {
 type master;
 file "/etc/bind/db.1.168.192";
};
```

En observant les directives du fichier ***/etc/bind/named.conf.local*** on voit que l'option ***file*** fait appel à deux fichiers:

/etc/bind/db.univ-bobo.bf

/etc/bind/db.1.168.192

Le fichier de zone ***/etc/bind/db.univ-bobo.bf*** dont le contenu est:

```
; CONFIGURATION DE LA ZONE "UNIV-BOBO.BF";
;
$TTL   604800        ;Durée en seconde pendant laquelle pendant laquelle les enregistrements
                     ;seront valides
@        IN            SOA          server.univ-bobo.bf. root.univ-bobo.bf. (
                                    200710102      ; numero de serie
                                     10800        ; Rafraîchissement
                                     3600         ; nouvel essai
                                     604800        ; Expiration après une semaine
                                     86400 )      ; TTL minimal de 1 jour
;
@      IN     NS     server.univ-bobo.bf.
       IN     MX  10  mail.univ-bobo.bf.
server  IN     A      192.168.1.30
server  IN     A      212.52.149.156
mail    IN     A      192.168.1.30
```

```
mail   IN    A     212.52.149.156

www    IN    CNAME   server.univ-bobo.bf.
smtp   IN    CNAME   server.univ-bobo.bf.
imap   IN    CNAME   server.univ-bobo.bf.
pop    IN    CNAME   server.univ-bobo.bf.
esi    IN    CNAME   server.univ-bobo.bf.
```

Le fichier de zone inverse */etc/bind/db.1.168.192:*

```
;/CONFIGURATION DE LA RÉSOLUTION INVERSE
$TTL   604800
@      IN    SOA    server.univ-bobo.bf. root.univ-bobo.bf. (
                    200710102       ; numero de serie
                    10800           ; Rafraîchissement
                    3600            ; nouvel essai
                    604800          ; Expiration après une semaine
                    86400 )         ; TTL minimal de 1 jour
@      IN    NS     server.univ-bobo.bf.

30             PTR    server.univ-bobo.bf.
30             PTR    www.univ-bobo.bf.
```

Quelques explications:

Tableau II.10: explications des options DNS

champ	description
CNAME	CNAME Enregistrement de nom canonique qui dit au serveur de noms qu'un nom donné est aussi connu qu'un autre (alias).
NS	Enregistrement de serveur de noms (NameServer) qui annonce les serveurs de noms faisant autorité pour une zone particulière.
MX	

A	Enregistrement d'adresse qui spécifie une adresse IP à assigner à un nom.
PTR	Enregistrement PoinTeR record, conçu pour orienter vers une autre partie de l'espace de nom.
SOA	Enregistrement "Start Of Authority", qui proclame des informations importantes faisant autorité à propos des espaces de nom pour les serveurs de noms.

La Sécurité du DNS

Sécurité liée au fichier de configuration.

Une petite modification à effectuer, pour éviter que notre serveur ne serve de relay DNS ouvert; c'est à dire, qu'il fasse autorité seulement pour notre zone pour empêcher le DNS spoofing (type de piratage). On ajoute ces lignes dans le fichier ***/etc/bind/named.conf.options*** (entre les {}):

```
allow-recursion { localhost; }; Définit les hôtes autorisés à des faire des demandes récursives
allow-query { 192.168.1.0/24; };  Définit les hôtes autorisés à faire des requêtes sur le serveur
allow-transfer{ 192.168.1.0/24; };
```

La SandBox

Le but de la SandBox, en cas d'attaque pirate, est de limiter l'accès à seulement une infime partie du système de fichiers.

Pour cela, nous allons démarrer le démon de BIND dans un environnement chrooté. L'effet obtenu sera de faire croire à BIND que son répertoire est sa propre racine de système de fichiers.

Nous allons voir comment sécuriser le DNS via la mise en place d'une SandBox. Pour cela, il va falloir procéder à quelques modifications :

- ✔ Éditer le fichier /etc/default/bind9 afin que le démon utilise l'utilisateur 'bind',

chrooté à **/var/lib/named**. Modifier la ligne : OPTIONS="-u bind" écrire

OPTIONS="-u bind -t /var/lib/named";

- ✔ Créer tous les fichiers nécessaires sous /var/lib

  # mkdir -p /var/lib/named/etc
  # mkdir /var/lib/named/dev
  # mkdir -p /var/lib/named/var/cache/bind
  # mkdir -p /var/lib/named/var/run/bind/run

- ✔ Ensuite déplacer les fichiers de configuration de */etc* vers */var/lib/named/etc*:

  # mv /etc/bind /var/lib/named/etc

- ✔ Créer un lien symbolique vers le nouveau répertoire de configuration (Cela nous permettra d'éviter tout problème lors des upgrades):

  # ln -s /var/lib/named/etc/bind /etc/bind

- ✔ Créer les périphériques "null" et "random" et résoudre les problèmes de permissions:

  # mknod /var/lib/named/dev/null c 1 3
  # mknod /var/lib/named/dev/random c 1 8
  # chmod 666 /var/lib/named/dev/null /var/lib/named/dev/random
  # chown -R bind:bind /var/lib/named/var/*
  # chown -R bind:bind /var/lib/named/etc/bind

- ✔ Ensuite il faut modifier le fichier de démarrage ***/etc/init.d/sysklogd*** du démon sysklogd afin qu'il puisse loguer tous les évènements importants du système. Modifier la ligne:

SYSLOGD="" et ecrire SYSLOGD="-a /var/lib/named/dev/log";

- ✔ Redémarrer le démon pour les logs: **# /etc/init.d/sysklogd restart**

A ce stade il reste juste que le FAI (Fournisseur d'Accès à Internet) fasse le lien entre notre nom de domaine univ-bobo.bf et notre adresse publique.

4.4.2 Le firewall (pare-feu)

Le firewall a pour but d'accroître la sécurité du réseau local, de détecter les tentatives d'intrusion et d'y parer au mieux possible. Cela permet de rendre le réseau ouvert sur Internet de façon beaucoup plus sûre. De plus, il peut permettre de restreindre l'accès interne vers l'extérieur. Le firewall propose donc un véritable contrôle sur le trafic réseau. Netfilter est le module qui fournit à Linux les fonctionnalités de pare-feu, de traduction d'adresses et d'historisation du trafic réseau.

Architecture et fonctionnement de netfilter

Netfilter se présente comme une série de cinq (5) « hooks » (points de d'accrochage), sur lesquels des modules de traitement des paquets vont se greffer. Ces points:

- NF_IP_PRE_ROUTING
- NF_IP_LOCAL_IN
- NF_IP_FORWARD
- NF_IP_POSTROUTING
- NF_IP_LOCAL_OUT

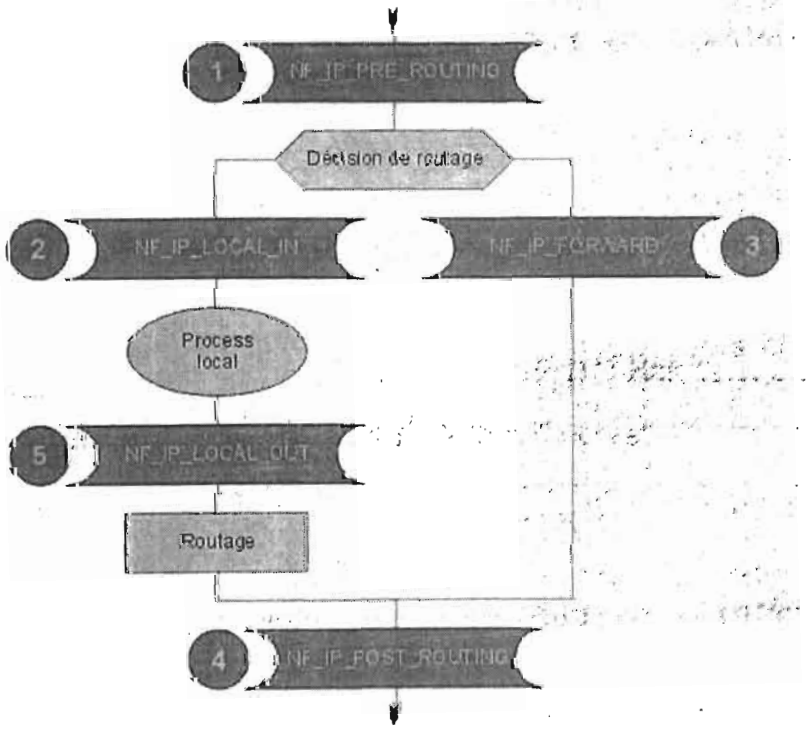


Figure II.10: Schéma de Netfilter

La branche gauche représente le trajet des paquets qui entrent et qui sortent vers et depuis un processus local.

La branche de droite représente le trajet des paquets qui traversent notre passerelle dans sa fonction de routeur.

A travers ces cinq points d'insertion, Netfilter va être capable :

- D'effectuer des filtrages de paquets, principalement pour assurer des fonctions de Firewall.
- D'effectuer des opérations de NAT (Network Address Translation). Ces fonctions sont particulièrement utiles lorsque l'on veut faire communiquer tout ou partie d'un réseau privé, monté avec des adresses IP privées (192.168.x.x par exemple) avec l'Internet.
- D'effectuer des opérations de marquage des paquets, pour leur appliquer un traitement spécial.

Il y a dans Netfilter trois **tables** qui correspondent aux trois principales fonctions. Chaque table contient des **chaînes**. Les **chaînes** sont des ensembles de règles que nous allons écrire dans chaque table. Ces **chaînes** vont permettre d'identifier des paquets qui correspondent à certains critères.

Les tables et leurs chaines

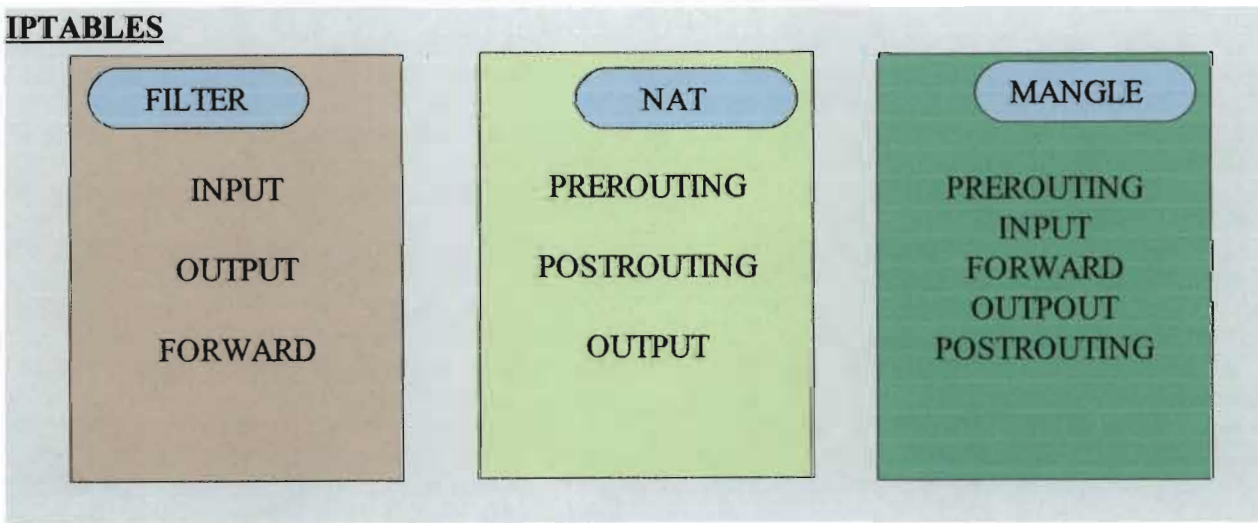


Figure II.11: schéma des tables de iptables

- PREROUTING: Le paquet est pris en charge par l'interface réseau. Il s'apprête à être routé.
- INPUT: Le paquet est destiné à l'hôte sur lequel nous définissons les règles. Il *nous* est destiné.

- FORWARD: Le paquet ne nous est pas destiné, et nous sommes une passerelle.
- OUTPUT: Le paquet est émis par nous.
- POSTROUTING: Le paquet s'apprête à sortir par l'interface réseau.

Les **cibles** enfin sont des sortes d'aiguillage qui dirigeront les paquets satisfaisant aux critères. Les cibles préconstruites sont :

- ACCEPT: Les paquets qui satisfont aux critères sont acceptés, ils continuent leur chemin dans la pile,
- DROP: Les paquets qui satisfont aux critères sont rejetés, on les oublie, on n'envoie même pas de message ICMP. Un trou noir, quoi.
- LOG: C'est une cible particulière qui permet de tracer au moyen de syslog les paquets qui satisfont aux critères.

Netfilter dispose d'une commande à tout faire : **iptables**. Cette commande va permettre, entre autres, d'écrire des chaînes de règles dans des tables. la syntaxe est la suivante:
#iptables [table] chaîne_spécifiée condition action

déploiement de la politique de sécurité.

La politique de sécurité mise en œuvre dans notre intranet est la suivante:

➔ Trafic du réseau externe vers la DMZ **autorisé**;

➔ Trafic du réseau externe vers le réseau interne **interdit**;

➔ Trafic du réseau interne vers la DMZ **autorisé**;

➔ Trafic du réseau interne vers le réseau externe **autorisé**;

➔ Trafic de la DMZ vers le réseau interne **interdit**;

Nous proposons un script qui contient l'ensemble des directives permettant la mise en place de notre politique de sécurité.

script_de_securite

```
##########################################################################
#######
#déclaration des interfaces
```

#Internet (l'interface liée au Net)
BAD_IFACE=eth1

#Demilitarized Zone
DMZ_IFACE=eth2
DMZ_ADDR=192.168.3.0/24

#LAN notre réseau local
GOOD_IFACE=eth0
GOOD_ADDR=192.168.1.0/24
#DMZ Server
*HTTP_SERVER=***.***.*.***

#SERVER SMTP
*SMTP_ADDR=***.**.**.**

#SERVER POP3
*POP3_ADDR=***.**.**.**

#SERVER DNS
*DNS_ADDR=***.*.*.**
##
#Le routage
 # commandes du routage. Nous lions les interfaces pour router les paquets d'un réseau à l'autre
*ip route del ***.***.*.***/** dev $BAD_IFACE* ***#suppression des routes par défaut***
*ip route del ***.***.*.***/** dev $DMZ_IFACE*
route del default dev eth1
route del default dev eth0
#t Ajout des routes pour l'interconnexion des interfaces
*ip route add ***.***.*.*** dev $BAD_IFACE*
*ip route add ***.***.*.***/** dev $DMZ_IFACE*
*route add default gateway ***.***.*.*** dev eth1*

 # liaison des interfaces

echo 1 >> /proc/sys/net/ipv4/ip_forward

###

le proxy ARP

 # construction de chaînes associées aux interfaces iptables

iptables -N bad-if

iptables -N dmz-if

iptables -N good-if

iptables -N icmp-acc

 # correspondence des interfaces avec la chaines

iptables -A INPUT -i $BAD_IFACE -j bad-if

iptables -A INPUT -i $DMZ_IFACE -j dmz-if

iptables -A INPUT -i $GOOD_IFACE -j good-if

###

vidage des anciennes régles

 # flush all rules in the the filter table

iptables -F

iptables -t nat -F

iptables -F INPUT

iptables -F OUTPUT

iptables -F FORWARD

###

#règles par défaut

 # par défaut on refuse tout

iptables -A INPUT -j DROP

iptables -A FORWARD -j DROP

iptables -A OUTPUT -j DROP

###

»# »

#construction des chaîne iptables

 # construction des chaîne pour agir sur les paquets

 # bad pour internet, dmzpour notre dmz, good pour le LAN

iptables -N good-dmz

iptables -N bad-dmz

iptables -N good-bad

iptables -N dmz-good

iptables -N dmz-bad

iptables -N bad-good

##
#########

 # Vidage des règles

iptables -F good-dmz

iptables -F bad-dmz

iptables -F good-bad

iptables -F dmz-good

iptables -F dmz-bad

iptables -F bad-good

iptables -F icmp-acc

iptables -F bad-if

iptables -F dmz-if

iptables -F good-if

##
########

le routages paquet avec iptables

 # NAT source + redirection #

 # nous faisons du Nat pour permettre aux machines du LAN d'acceder à la DMZ

#####################

 # Redirection HTTP

iptables -t nat -A PREROUTING -i eth0 -s $GOOD_ADDR -p tcp --dport http -j DNAT --to-destination $HTTP_SERVER:80

 # Redirection DNS

iptables -t nat -A PREROUTING -i eth0 -s $GOOD_ADDR –d $DNS_ADDR -p udp --dport

domain -j DNAT --to-destination $DNS_SERVER:53

 # Redirection SMTP

iptables -t nat -A PREROUTING -i eth0 -s $GOOD_ADDR –d $SMTP_ADDR -p tcp --dport smtp -j DNAT --to-destination $SMTP_SERVER:25

iptables -t nat -A PREROUTING -i eth0 -s $GOOD_ADDR –d $SMTP_ADDR -p udp --dport smtp -j DNAT --to-destination $SMTP_SERVER:25

 # Redirection POP

iptables -t nat -A PREROUTING -i eth0 -s $GOOD_ADDR –d $POP_ ADDR -p tcp --dport pop3 -j DNAT --to-destination $POP_SERVER:110

iptables -t nat -A PREROUTING -i eth0 -s $GOOD_ADDR –d $POP_ ADDR -p udp --dport pop3 -j DNAT --to-destination $POP_SERVER:110

##

##suivi de connection

iptables -A FORWARD -m state --state INVALID -j DROP

iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT

iptables -A FORWARD -s $DMZ_ADDR -i $DMZ_IFACE -o $BAD_IFACE -j dmz-bad

iptables -A FORWARD -s $DMZ_ADDR -i $DMZ_IFACE -o $GOOD_IFACE -j dmz-good

iptables -A FORWARD -s $GOOD_ADDR -i $GOOD_IFACE -o $DMZ_IFACE -j good-dmz

iptables -A FORWARD -s $GOOD_ADDR -i $GOOD_IFACE -o $BAD_IFACE -j good-bad

iptables -A FORWARD -o $DMZ_IFACE -j bad-dmz

iptables -A FORWARD -o $GOOD_IFACE -j bad-good

 # drop /bloquer tous les paquets qui ne sont pas conformes

iptables -A FORWARD -j LOG --log-prefix "chain-jump"

iptables -A FORWARD -j DROP

###

règles pour les paquet ICMP

 # icmp acceptance

iptables -A icmp-acc -p icmp --icmp-type destination-unreachable -j ACCEPT

iptables -A icmp-acc -p icmp --icmp-type source-quench -j ACCEPT

iptables -A icmp-acc -p icmp --icmp-type time-exceeded -j ACCEPT

iptables -A icmp-acc -p icmp --icmp-type echo-request -j ACCEPT

iptables -A icmp-acc -p icmp --icmp-type echo-reply -j ACCEPT

 # iptables -A icmp-acc -j LOG --log-prefix "icmp-acc"

iptables -A icmp-acc -j DROP

##

"

règles pour la communication inter-réseau

 # from internal(LAN) to dmz

iptables -A good-dmz -p tcp --dport http -j ACCEPT

#HTTP

iptables -A good-dmz -p tcp --dport ftp -j ACCEPT

#FTP

iptables -A good-dmz -p udp --dport domain -j ACCEPT

#DNS

iptables -A good-dmz -p tcp --dport smtp -j ACCEPT

#SMTP

iptables -A good-dmz -p udp --dport smtp -j ACCEPT

#SMTP

iptables -A good-dmz -p tcp --dport pop3 -j ACCEPT

#POP3

iptables -A good-dmz -p tcp --dport ripng -j ACCEPT

#RIP v2

iptables -A good-dmz -p udp --dport ripng -j ACCEPT

```
#RIP v2

iptables -A good-dmz -p tcp --dport ssh -j ACCEPT
#SSH

iptables -A good-dmz -p udp --dport ssh -j ACCEPT
#SSH
 iptables -A good-dmz -j DROP
#ACCEPT<->DROP
#DROP les autres

# de l'extérieur(WAN) vers la DMZ
iptables -A bad-dmz -p tcp --dport ident -j ACCEPT

iptables -A bad-dmz -j DROP
#DROP tout

# de  l' interieur (LAN)vers l'exterieur (WAN)

iptables -A good-bad -p tcp --dport ftp -j ACCEPT
#FTP direct cf WAN->LAN ident

iptables -A good-bad -p tcp --dport pop3 -j ACCEPT
#POP3 direct

iptables -A good-bad -p udp --dport pop3 -j ACCEPT
#POP3 direct

iptables -A good-bad -p tcp --dport smtp -j ACCEPT

#SMTP direct
iptables -A good-bad -p udp --dport smtp -j ACCEPT
#SMTP direct
```

```
iptables -A good-bad -p udp --dport domain -j ACCEPT
#DNS direct

iptables -A good-bad -p tcp --dport https -j ACCEPT
#HHTPS direct,

iptables -A good-bad -j REJECT
#REJECT all
 #de la DMZ vers le  LAN
iptables -A dmz-good -j DROP
#ACCEPT<->DROP
#DROP all

# de la DMZ vers l'extérieure

iptables -A dmz-bad -p tcp --dport ftp -j ACCEPT
#FTP

iptables -A dmz-bad -p tcp --dport http -j ACCEPT
#HTTP

iptables -A dmz-bad -p tcp --dport pop3 -j ACCEPT
#POP3

iptables -A dmz-bad -p tcp --dport smtp -j ACCEPT
#SMTP

iptables -A dmz-bad -p udp --dport smtp -j ACCEPT
#SMTP

iptables -A dmz-bad -p udp --dport domain -j ACCEPT
#DNS
```

```
iptables -A dmz-bad -j REJECT
#REJECT les autres
# de l'extérieure (WAN) vers l'interieur (LAN)
iptables -A bad-good -p tcp --dport ident -j ACCEPT
#ACCEPT ident->security hole for FTP direct

iptables -A bad-good -j DROP
#DROP all
##########################################################################
#régles pour les interfaces de la machine firewall
# interface externe
iptables -A bad-if -p icmp -j icmp-acc
iptables -A bad-if -j DROP                    #DROP<->ACCEPT for debugging

# interface de la  Dmz
iptables -A bad-if -p icmp -j icmp-acc
iptables -A dmz-if -j DROP                    #DROP<->ACCEPT for debugging

# L'interface intérieure reçoit les pings, les réponses ping,le ssh et les erreurs ICMP
iptables -A good-if -p tcp --dport ssh -j ACCEPT
iptables -A good-if -p ICMP --icmp-type ping -j ACCEPT
iptables -A good-if -p ICMP --icmp-type pong -j ACCEPT
iptables -A good-if -j icmp-acc

# Remove the compete blocks
#Supprime les règles de bloquage
iptables -D INPUT 1
iptables -D FORWARD 1
iptables -D OUTPUT 1
```

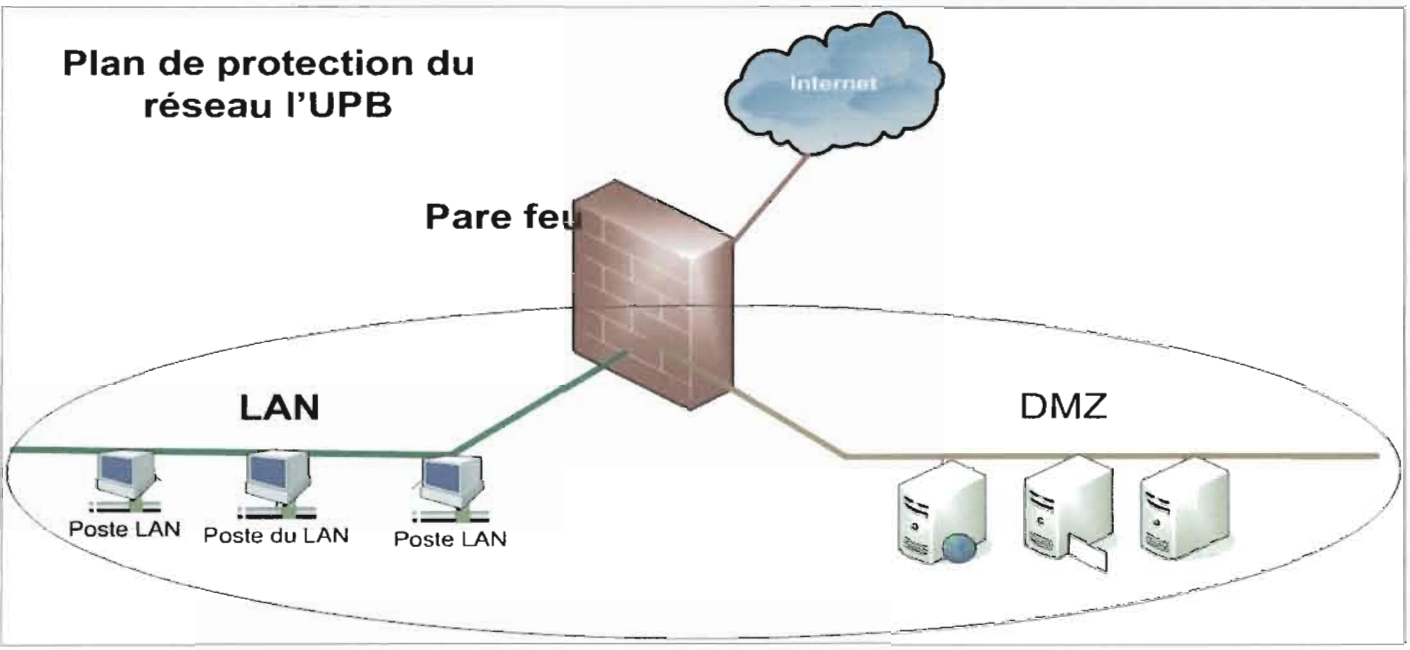


Figure B.II.12: Schéma illustrant la protection par le firewall

Notons que, tous les échanges entre le LAN, la DMZ et Internet se font à travers le pare-feu qui empêche ou laisse passer les paquets selon leur provenance, leur destination et leur type, conformément aux règles établies.

5) Estimations

Dans cette partie, nous allons estimer la durée et le coût que la réalisation de notre projet nécessite. Cependant, bien que nous ayons mené une étude tenant compte de la définition de l'architecture physique du système, nous allons prendre en compte seulement la mise en place des services réseaux dans nos estimations. En effet, la partie de l'étude portant sur l'architecture physique du réseau futur nous prendrait énormément de temps et nécessiterait des connaissances plus approfondies dans le domaine des télécommunications. Nous suggérons qu'elle soit confiée à des spécialistes qui disposent également du matériel adéquat pour ce travail.

5.1 Durée

Notre stage n'a duré que quatre mois. Nous avons cependant su mettre ce temps à profit pour mener une étude sur la plupart des services à déployer. Nous avons également pu tester l'ensemble de ces services, dont certains fonctionnent d'ailleurs

à profit pour mener une étude sur la plupart des services à déployer. Nous avons également pu tester l'ensemble de ces services, dont certains fonctionnent d'ailleurs bien. Cependant, pour réaliser complètement un tel projet, en tenant compte de la sécurité qui est d'une importance capitale, il nous faudrait beaucoup plus de temps. Si nous considérons les résultats que nous avons pu produire durant le temps de notre stage, nous estimons que pour une étude plus complète prenant en compte toutes les données de l'UPB, il nous faudrait quatre (04) mois supplémentaires. La mise en place de ces services nécessiterait également huit (08) mois. Soit au total douze (12) mois.

5.2 Coûts

Tableau II.11: Tableau estimatif des coûts.

Désignation	**Caractéristiques**	**Quantité**	**Prix unitaire**	**Montant**
Ordinateur firewall	Pentium II 350Mhz	01	85.000	85.000
Ordinateur serveur	Pentium IV 3,4 Ghz 4*73,4 Go 1 Go de RAM	04	1.200.000	4.800.000
Carte réseau	Carte pci Gigabit 10/100/1000 Mbps	07	15.000	105.000
Système d'exploitation	Debian GNU/Linux 4.0 Etch	-	Gratuit	Gratuit
Nom de domaine	univ-bobo.bf	-	existant	existant
Serveur HTTP	Apache2	01	Gratuit	Gratuit
Serveur FTP	VsFTPD	01	Gratuit	Gratuit
SGBD	Mysql	01	Gratuit	Gratuit
Système global de messagerie	Posfix,courier, squirrelmail	-	Gratuit	Gratuit
Serveur d'annuaire	OpenLDAP	-	Gratuit	Gratuit
Antivirus	Clamav,amavisd-new, spamassassin	-	Gratuit	Gratuit
Logiciel firewall	Netfilter (iptables)	-	Gratuit	Gratuit
Main d'oeuvre	-	-	-	6.000.000

Coût total	-	-	-	10.990.000

III. BILAN DE RÉALISATION

Cette partie a pour but de faire le bilan des travaux que nous avons pu réaliser au cours du stage afin de dégager des perspectives qui permettront, pour une réalisation éventuelle de notre projet d'étude, d'atteindre les objectifs de départ. Il convient donc de rappeler brièvement ces objectifs avant d'en venir au bilan. Il s'agit de mettre en place dans un Intranet, des services et des ressources réseaux qui seront partagés mais avec des droits d'accès spécifiques selon les utilisateurs. De plus ce réseau local devra être relié à Internet pour que certains de ses services aient toute leur importance (le Web par exemple). Nous sommes alors confrontés à deux contraintes majeures: contrôler les accès au seins de l'Intranet d'une part et d'autre part sécuriser ce dernier vis-à-vis de l'extérieur (Internet). Pour contrôler les accès et tout ce qui nécessite une authentification (du point de vue des utilisateurs et des services), nous avons proposé la mise en place d'un service d'annuaire. Quant à la protection de l'Intranet, nous envisageons pour cela la mise en place d'un firewall.

1) Les services en production

A ce stade de notre travail, nous avons pu mener une étude globale de la mise en place de notre Intranet. Nous avons également fait des recherches (bibliographiques et surtout webographiques) qui sont incontournables dans une telle étude. A l'issue de ces recherches, nous avons réussi à faire une implémentation minimale de chaque service. Nous pouvons à ce jour présenter une machine hébergeant la plupart des services, déjà configurés. Nous avons cependant, volontairement, omis de configurer tous les services étudiés pour éviter de surcharger le serveur. D'ailleurs même si tous les tests ont été effectués sur la même machine, notre étude prévoie de répartir les services sur plusieurs serveurs physiques.

Ainsi, en ce qui concerne les services internes du LAN, d'abord l'attribution des

adresses aux postes clients se fait par DHCP; il reste toutefois la création des pools d'adresses pour rester dans la logique de l'architecture que nous avons proposée. Nous avons également testé le partage de ressources (par SAMBA et NFS) qui marche.

Les services de la DMZ sont également déjà fonctionnels. En effet la mise en place du serveur Web est effective et grâce à la configuration des hôtes virtuels nous avons plusieurs sites Web hébergés (notamment pour l'UPB et l'ESI). Ces sites sont accessibles par les URLs suivantes: www.univ-bobo.bf et esi.univ-bobo.bf. Ce qui montre que le DNS aussi fonctionne.

Le système de messagerie électronique est lui aussi fonctionnel. Il est donc possible de créer un compte de messagerie pour les acteurs de l'université. Ces derniers ont même le choix entre l'utilisation du webmail, accessible via le site web de l'UPB et la configuration d'un client de messagerie (Thunderbird, evolution, Microsoft Outlook, etc) pour l'envoi et la réception de leur courrier. Nous signalons que nous avons effectué toutes les configurations à distance par le service SSH (Secure SHell) qui était le premier à être installé après le déploiement du système d'exploitation de base. SSH est un service sécurisé qui permet l'émulation d'un terminal à distance.

En somme, tous les services sont déjà fonctionnels et avec un minimum de sécurité. Il reste à les connecter au serveur d'annuaire que nous avons également testé.

2) Perspectives

A travers le bilan précédent, nous pouvons dire que la mise en place des services réseaux de l'Intranet de l'UPB est en bonne voie. Cependant, comme nous l'avons souligné dans la partie des estimations, une telle étude doit prendre en compte de nombreux éléments et sa réalisation nécessite donc beaucoup plus de moyens et de temps que nous n'en avons eu. En effet, la première chose que nous relevons est que nous avons fait nos tests, non seulement sur une seule machine ayant servi de serveur alors qu'il nous en fallait quatre en plus de l'ordinateur qui doit servir de firewall; en plus les caractéristiques de cette machine font qu'elle ne pourrait pas faire partir des serveurs que nous avons proposé. Il faudrait donc que l'université acquière le matériel proposé dans les estimations.

Du point de vu des configurations, il reste également un point important : la connexion de tous les services à OpenLDAP pour pouvoir centraliser la gestion de toutes les ressources du réseau et fixer un point commun d'authentification aux services réseaux et aux utilisateurs voulant accéder aux ressources.

Nous avons également prévu d'automatiser certaines tâches (les sauvegardes de données par exemple), par des scripts.

Une chose que nous jugeons également importante est la mise en place de Systèmes de Détection d'Intrusion (IDS pour Intrusion Detection Système) voire de Systèmes de Prévention d'Intrusions (IPS pour Intrusion Prevention Système).

Il convient donc que notre étude globale soit poursuivie et approfondie par d'autres recherches pour aboutir à de meilleurs résultats.

CONCLUSION GENERALE

Partie intégrante de la formation des étudiants de l'ESI, le projet de fin de cycle a permis pour notre part, de mener pendant plus de trois mois, une étude pour la mise en place d'un Intranet à l'UPB et la gestion centralisée de toutes les ressources de ce réseau. Ce stage nous aura permis d'aborder d'un point de vue plus pratique, les enseignements reçus durant tout notre cycle, notamment dans les domaines des réseaux informatiques et des télécommunications. Notre thème : **« GESTION CENTRALISEE DES RESSOURCES INFORMATIQUES DE L'UPB AVEC LE PROTOCOLE LDAP »** a été d'un très grand intérêt car il nous a permis de revoir la quasi-totalité des notions les plus importantes pour la formation des étudiants en fin de cycle de Réseaux et Maintenance Informatiques. En plus, nous pensons que la réalisation de notre étude permettrait à l'université d'être dotée d'un système informatique digne de son rang de deuxième université du Burkina Faso.

ADSL	Asymetric Digital Subscriber Line
APT	Advanced Packaging Tools
AUF	Agence Universitaire de la Francophonie
BIND	Berkeley Internet Name Domain
BIOS	Basic Input Output System
CAI	Centres d'Accès à l'Information
CICI	Cycle des Ingénieurs de Conception Informatique
CITI	Cycle des Ingénieurs de Travaux Informatiques
DELGI	Délégation Générale à l'Informatique
DMZ	Zone démilitarisée
DN	Distinguished Name: identifiant unique dans le cadre des annuaires LDAP
DNS	Domain Name Server ou Damain Name System
ESI	Ecole Supérieure d'Informatique
FAI	Fournisseur d'Accès à Internet
FQDN	Fully Qualified Domain Name
GNU	Gnu's Not Unix
HP	Hewlett Packard
HTTP	Hyper Text Transfer Protocol
IDR	Institut du Développement Rural
IMAP	Internet Message Access Protocol
IUT	Institut Universitaire de Technologie
LAN	Local Area Network
LDAP	Lighweight Directory Access Protocol
LDIF	LDAP Data Interchange Format
LS	Ligne Spécialisée
MAC	Media Acces Control

MDA	Mail Delivery Agent
MTA	Mail Transfer Agent
MUA	Mail User Agent
MX	Mail eXchange
NS	Name Server
OSI	Open Systems Interconnection
PC	Personal Computer
PDF	Portable Document Format
PHP	Hypertext Preprocessor
POP	Post Office Protocol
PTR	Pointer Record
RFC	Request For Comment
SASL	Simple Authentication and Security Layer
SMTP	Simple Mail Transport Protocol
SOA	Start Of Authority
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Socket Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
TIC	Technologies de l'Information et de la Communication
TLD	Top Level Domain
TLS	Transport Layer Security
TTL	Total Time to Live / Time To Live
UDP	User Datagram Protocol
UPB	Université Polytechnique de Bobo-Dioulasso
URL	Uniform Resource Locator
USB	Universal Serial Bus

Bibliographie:

Cahiers de l'Admin Debian, EYROLLES 2e édition,Raphael HERTZOG

Webographie:

www.ac-creteil.fr

www.apache.org

www.bizeul.net

www.coagul.org

www.commentcamarche.net

www.debian.org

www.debianaddict.org

www.developpez.net

www.framasoft.net

www.funix.org

www.generation-linux.net

www.google.com

www.khelifi.org

www.labo-linux.org

www.lea-linux.org

www.libordux.org

www.linux-france.org

www.linux-pour-lesnuls.com

www.linux-sottises.net

www.littleboboy.net

www.postfix.org

www.scribd.com

www.starbridge.org

www.starbridge.org

www.supinfo.com

www.wikipedia.org