

12. 04.07.1

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique
Université SAAD DAHLEB Blida
Faculté des sciences
Département d'informatique

Mémoire de fin d'étude en vue de l'obtention d'un
diplôme d'ingénieur d'état en Informatique

Option : systèmes d'information avancés

Thème

Conception et réalisation d'un outil de
configuration des divers services réseaux
sous Linux

Réalisé par :

Benarbia Mohamed
Yazid Boukhalfa

Soutenue publiquement le: 03/12/2003

Devant le jury composé de :

Dr M. Mehiedine : président
Mr D. Boukhelef : examinateur
Mr M. Mehdi : examinateur
Mr D. Bennouar : promoteur

Promotion 2002/2003



DEDICACES

Je dois remercier le bon Dieu qui nous a donné le courage, la patience et la santé pour terminer ce travail. Dieu merci !

Ce modeste travail est dédié aux êtres les plus chers dans ma vie.

A mes parents qui m'ont aidé pendant tout mon cursus de formation.

A mon frère et mes trois sœurs pour leurs encouragements.

A toute la famille YAZID.

A tous les habitants de mon village d'Ait Amarra (Akfadou) à Bejaia.

A mes amis de l'enfance Mahmoud, Tahar, Mohamed et Wassime.

A mes amis Anis, Amine (chaarie), Hanie et Bensalem (bellouta).

A tous mes collègues du première année TCT.

A tous les étudiants de notre promotion.

BOUKHALFA

DEDICACES

je dois remercier le bon Dieu qui nous a donné le courage, patience et santé pour terminer ce travail. Dieu merci !

Alors ce modeste travail mérite d'être dédié aux êtres les plus chers dans ma vie .

A mes parent qui m'ont aidé pendant tout mon cursus de formation.

A toute la famille BENARBIA.

A tous mes amis qui me connaissent. sans exception.

A tous mes collègues universitaires et à tous les étudiants de notre promotion.

MOHAMED

Remerciements

Nous remercions tout d'abord le bon dieu de nous avoir donner le courage, la patience et l'aide pour dépasser tous les obstacles.

Nous tenons à remercier particulièrement nos parents pour l'aide constant qu'ils nous ont apportés lors de la réalisation de notre projet de fin d'étude.

Nous tenons à remercier aussi notre promoteur et les membres de la commission de suivi qui nous ont assisté pour la réalisation de ce projet, par leurs critiques et conseils précieux.

Il nous faut également souligner l'apport de notre chef de département et nos professeurs qui nous ont aidé pendant notre cursus universitaire.

Tous nous remercions à tous nos amis et nos collègues de notre promotion.

Sommaire

Introduction générale.....	1
-----------------------------------	----------

Chapitre 1 : Présentation de Linux

1.1- Qu'est ce que Linux.....	5
1.2- Les caractéristiques essentielles de Linux.....	6
1.2.1- Le noyau de Linux.....	6
1.2.2- Le système de fichiers.....	6
1.2.3- Les interpréteurs de commandes.....	6
1.2.4- Les utilitaires de base.....	6
1.2.5- Les protocoles de communication.....	6
1.2.6- Le système X-Window	7
1.2.7- L'environnement de bureau.....	7
1.3- Les distributions de Linux.....	8
1.4- La licence d'utilisation de Linux.....	9
1.5- La documentation sur Linux.....	9
1.6- Conclusion.....	10

Chapitre 2 : Notions de base sur les réseaux

2.1- Définitions de base.....	12
2.2- Présentation de TCP/IP	14
2.2.1- Architecture de TCP/IP	14
2.2.2- Hiérarchie de l'implémentation TCP/IP	16
2.3- L'adressage IP.....	17
2.4- La notion de port d'écoute.....	19
2.5- Le domaine de noms.....	19
2.6- Conclusion.....	20

Chapitre 3 : Etude des services réseaux à configurer

3.1- Introduction.....	22
3.2- Configuration de base de TCP/IP sous Linux.....	22
3.3- Le service de noms.....	23
3.3.1- Introduction.....	23
3.3.2- L'organisation de DNS	24
3.3.3- Le fonctionnement de DNS	27
3.3.4- La configuration du serveur DNS	27

3.4- Le service web.....	29
3.4.1- Introduction.....	29
3.4.2- Le fonctionnement de HTTP	30
3.4.3- La configuration du serveur HTTP	31
3.5- Le service FTP	34
3.5.1- Introduction.....	34
3.5.2- Le fonctionnement de FTP	34
3.5.3- La configuration du serveur FTP	37
3.6- Le service de messagerie électronique.....	39
3.6.1- Introduction.....	39
3.6.2- Le fonctionnement de SMTP	40
3.6.3- La configuration du serveur SMTP	42
3.7- Le service de news.....	45
3.7.1- Introduction.....	45
3.7.2- Le fonctionnement de NNTP	46
3.7.3- La configuration du serveur NNTP	47
3.8- Le service de partage de ressources Samba	50
3.8.1- Introduction.....	50
3.8.2- Le fonctionnement de Samba	51
3.8.3- La configuration du serveur Samba	52
3.9- Le service de routage et Fire-wall.....	55
3.9.1- Introduction.....	55
3.9.2- Le fonctionnement d'un Fire-wall.....	57
3.9.3- Le routage derrière un Fire-wall.....	58
3.9.4- La configuration du serveur routeur/Fire-wall.....	60
3.10- Conclusion.....	62

Chapitre 4 : Conception

4.1- Introduction.....	64
4.2- Conception des modules de configuration des services.....	65
4.2.1- Module de configuration du service de noms.....	65
4.2.2- Module de configuration du service web.....	68
4.2.3- Module de configuration du service FTP	72
4.2.4- Module de configuration du service de messagerie électronique.....	75
4.2.5- Module de configuration du service de news.....	79
4.2.6- Module de configuration du service de partage de ressources Samba	83
4.2.7- Module de configuration du service de routage et Fire-wall....	85
4.3- Conception de l'interface utilisateur.....	90
4.4- Conclusion.....	91

Chapitre 5 : Réalisation

5.1- Introduction.....	93
5.2- Environnement de développement matériel.....	93
5.3- Environnement de développement logiciel.....	93
5.4- Problèmes rencontrés et solutions proposées.....	94
5.5- Présentation de l'interface utilisateur.....	95
5.5.1- L'interface principale.....	96
5.5.2- L'interface du module de configuration du service de noms..	97
5.5.3- L'interface du module de configuration du service web.....	101
5.5.3- L'interface du module de configuration du service FTP.....	105
5.5.4- L'interface du module de configuration du service de messagerie électronique.....	107
5.5.5- L'interface du module de configuration du service de news.....	109
5.5.6- L'interface du module de configuration du service de partage de ressources Samba.....	111
5.5.7- L'interface du module de configuration du service de routage et Fire-wall.....	113
5.6- Conclusion.....	116
 Conclusion générale.....	 117
 Bibliographie.....	 118

Index des figures

Chapitre 1

Figure-1: L'architecture en couches de Linux.....	5
--	---

Chapitre 2

Figure-2.1: Transfert de données selon le modèle DoD	15
Figure-2.2: La hiérarchie de l'implémentation de TCP/IP	16

Chapitre 3

Figure-3.1: Exemple extrait d'un espace de noms.....	25
Figure-3.2: La relation client/serveur FTP	35
Figure-3.3: Schéma d'une messagerie SMTP	40
Figure-3.4: Représentation d'un système Fire-wall.....	56

Chapitre 4

Figure-4.1: Schéma de conception de l'outil de configuration des divers services réseaux.....	64
Figure-4.2: Schéma de conception du module de configuration du service de noms.....	65
Figure-4.3: Schéma de conception du module de configuration du service web.....	69
Figure-4.4: Schéma de conception du module de configuration du service FTP	73
Figure-4.5: Schéma de conception du module de configuration du service de messagerie électronique.....	76
Figure-4.6: Schéma de conception du module de configuration du service de news.....	80
Figure-4.7: Schéma de conception du module de configuration du service de partage de ressources Samba	83
Figure-4.8: Schéma de conception du module de configuration du service de routage et Fire-wall.....	85

Chapitre 5

Figure-5.1: Interface principale de l'outil de configuration des divers services réseaux sous Linux.....	96
Figure-5.2: Interface du module de configuration du service de noms..	97

Figure-5.3: La page des propriétés du serveur de noms.....	98
Figure-5.4: La page d'ajout d'une zone primaire.....	99
Figure-5.5: La page de modification d'une zone primaire.....	99
Figure-5.6: La page d'ajout d'un enregistrement de ressource MX.....	100
Figure-5.7: La page de teste de bon fonctionnement du service de noms primaire.....	100
Figure-5.8: Interface du module de configuration du service web.....	101
Figure-5.9: La page des propriétés du serveur web.....	102
Figure-5.10: La page de sélection des classes de paramètres du serveur.....	103
Figure-5.11: La page de configuration des paramètres du site.....	103
Figure-5.12: La page d'ajout d'un hôte virtuel.....	104
Figure-5.13: La page de configuration des paramètres SSL.....	104
Figure-5.14: L'interface du module de configuration du service FTP	105
Figure-5.15: La page des propriétés du serveur FTP	106
Figure-5.16: La page d'ajout d'un hôte virtuel FTP	106
Figure-5.17: L'interface du module de configuration du service de messagerie électronique.....	107
Figure-5.18: La page des propriétés du serveur SMTP	108
Figure-5.19: La page de configuration des aliasés.....	108
Figure-5.20: L'interface du module de configuration du service de news.....	109
Figure-5.21: La page des propriétés du serveur de news.....	110
Figure-5.22: La page d'ajout d'un groupe de discussion.....	110
Figure-5.23: L'interface du module de configuration du service Samba	111
Figure-5.24: La page des propriétés du serveur SMB	112
Figure-5.25: La page d'ajout d'une ressource à partager.....	112
Figure-5.26: L'interface du module de configuration du service de routage et Fire-wall.....	113
Figure-5.27: La page des propriétés du serveur routeur/fire-wall.....	114
Figure-5.28: La page de configuration de la table de routage.....	114
Figure-5.29: La page de configuration des règles de sécurité.....	115
Figure-5.30: La page d'ajout d'une règle de sécurité.....	115
Figure-5.31: La page d'ajout d'une entrée ProxyArp.....	116

Introduction générale

Linux est un système d'exploitation très puissant développé autour du système d'exploitation traditionnel UNIX et caractérisé par l'emploi de l'interface graphique X-Window et ses sur-couches (environnements de bureau comme *KDE*, *GNOME*, ...) qui a constitué une solution facile et efficace pour les débutons non familiarisés avec *UNIX* pour manipuler ses applications sans avoir une connaissance des détails de son système de fichiers et ses langages de commandes (*Shells*).

La manipulation de la plus part des applications Linux peut s'effectuer à partir des interfaces graphiques. Cependant pour effectuer certaines travaux comme la configuration des services réseaux par exemple, il est nécessaire de travailler en mode console (lignes de commande), même si cette opération demande une connaissance de l'emplacement de tous les fichiers de configuration utilisés par un service réseau donné, la connaissance des règles syntaxiques suivies dans ces fichiers, la connaissance des commandes de redémarrage des serveurs appropriés, des commandes de test, des commandes de modification de certains paramètres de configuration de ces services comme les commandes de génération des fichiers de base données, ainsi que leurs différentes options.

Cette nécessité est causée par l'absence des outils de configuration de certains services réseaux munis des interfaces graphiques ou la présence des outils de configuration munis des interfaces graphiques mais, ces outils ou bien impliquent des interventions supplémentaires en mode console après leur utilisation pour effectuer certaines tâches de test de vérification de la bonne configuration ou du bon fonctionnement de ces services ou bien présentent des imperfections ou des faiblesses

comme le fait l'outil de configuration du service de noms de Red-Hat Linux 7 où il arrive souvent que la prise en charge d'une nouvelle configuration ne soit pas effectuée.

Notre travail consiste à concevoir un outil de configuration des divers services réseaux connus (service de noms, service web, service de messagerie électroniques, service de routage et Fire-wall, ...) muni d'une interface graphique permettant en plus la configuration la possibilité de tester la bonne configuration ou le bon fonctionnement de ces services réseaux, et de le réaliser dans une architecture qui s'articule autour d'un serveur web, en gardant l'administrateur du réseau local très loin des détails des fichiers de configuration, des règles syntaxiques suivies dans ces fichiers et les détails des commandes de redémarrage et de teste de ces services.

Nous avons organisé notre travail dans ce mémoire de fin d'étude comme suit :

Dans le premier chapitre, nous allons donner une brève présentation du système d'exploitation Linux (qu'est que Linux, caractéristiques essentielles de Linux, la licence *GPL*, ...).

Dans le deuxième chapitre, nous allons aborder quelques concepts de base des réseaux (définitions de base, présentation du protocole **TCP/IP**, l'adressage **IP**, la notion de port d'écoute, ...).

Dans le troisième chapitre, nous allons faire une étude sur les services réseaux à configurer par notre outil (le service de noms, le service web, le service **FTP**, le service de messagerie électronique, le service de news, le service de partage des ressources Samba et le service de routage et Fire-wall), cette étude va nous permettre d'aborder leurs fonctionnements et les techniques utilisées pour leurs configurations sous Linux en montrant la difficulté et la complexité d'une telle configuration.

Dans le chapitre quatre, nous allons présenter la démarche de conception de notre outil et les fonctionnalités offertes par son interface utilisateur.

Dans le chapitre cinq, nous allons montrer les détails de la réalisation de notre outil (l'environnement de développement matériel, l'environnement de développement logiciel, problèmes rencontrés et leurs solutions proposées et aussi la présentation de son interface utilisateur).

Chapitre 1

Présentation de Linux

1.1- Introduction

Linux est un système d'exploitation assez récent développé autour du système d'exploitation traditionnel UNIX. La première version de son noyau (version 0.01) fut mise au point au mois de septembre 1991 par Linus Torvalds (étudiant à l'université de Helsinki en Finlande) . La version 0.10 fut publiée sur Internet en novembre 1991.

Linux fut destiné au départ pour les micro ordinateurs compatibles IBM PC. Aujourd'hui, il existe des versions pour d'autres plates-formes matérielles tels que le *DEC Alpha*, le *Sun Sparc*, le *PowerPC*, ...

Linux n'a pas été développé pour un but commercial. C'est un logiciel du domaine public. Son code source est disponible et peut être librement copié, modifié et distribué.

Après la publication du code source de Linux sur Internet, un nombre très important des programmeurs du monde entier s'est intéressé à ce système et a eu lieu l'idée de mettre au point un système d'exploitation puissant, stable et gratuit. Ces programmeurs ont commencé alors à réaliser diverses extensions et améliorations pour ce système.

1.2- Les caractéristiques essentielles de Linux

En réalité, le terme Linux ne désigne que le noyau (*Kernel*). Il s'agit donc de la partie interne d'un système d'exploitation qui se charge du contrôle du matériel et de la gestion des processus, de la mémoire et des fichiers. Mais, pour la plupart des utilisateurs, Linux, tel qu'il est utilisé au quotidien, désigne, outre le noyau, la multitude des programmes et applications logicielles qui l'accompagnent et composent les différentes distributions.

LINUX est composé d'un noyau et d'une variété d'applications comme le montre la figure suivante :

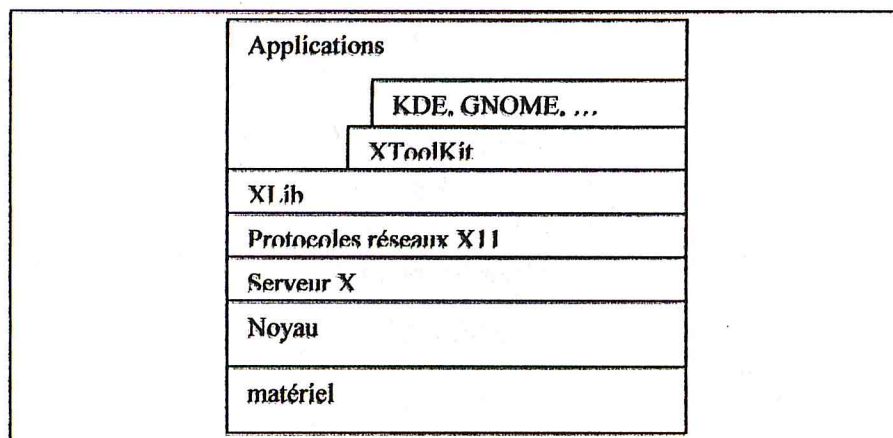


Figure-1 : L'architecture en couches de Linux.

Nous citons brièvement les caractéristiques importantes de Linux :

1.2.1- le noyau de Linux

Le noyau Linux supporte : le multitâche, l'exploitation multi utilisateur, multi plates-formes, le chargement des exécutables à la demande, la pagination mémoire à la demande, les bibliothèques partagées, le système de gestion de fichiers *ext2fs* et *ext3fs* (*extended file system version 2* et *3*), l'accès aux systèmes de fichiers étrangers (*Windows, NTFS, NFS, OS/2, ...*) et l'implémentation du support réseau comme un module du noyau (fonctions de communication très rapides).

1.2.2- Le système de fichiers

Le système de fichiers Linux est organisé en une structure arborescente où chaque répertoire est un nœud, chaque fichier est une feuille et chaque arc représente une relation de parenté.

Nous citons ici les répertoires Linux les plus importants :

- Le répertoire */* : le répertoire racine.
- Le répertoire */home* : le répertoire home des utilisateurs.
- Le répertoire */bin* : contient les programmes Linux de base.

- Le répertoire */sbin* : contient les programmes lancés automatiquement.
- Le répertoire */usr* : contient plusieurs répertoires (*bin* qui contient les programmes utilisateurs, *sbin* contient les programmes root, ...).
- Le répertoire */etc* : contient les répertoires et les fichiers de configuration des différentes applications Linux.

Linux contient aussi des fichiers spéciaux comme les fichiers *rc* qui sont utilisés pour le lancement de certains programmes au démarrage, les fichiers du répertoire */proc* pour lire les informations sur le système, ...

1.2.3- Les interpréteurs de commandes

Ils sont appelés les *shells*. Sous Linux plusieurs *shells* sont disponibles tels que : le *pdksh* (*Public Domain Korn Shell*), le *tcsch* (une variante du *C Shell*), mais le plus populaire est le *bash* (*Bourne Again Shell*). Il faut noter que ces interpréteurs de commandes sont totalement programmables et donc chaque utilisateur a le droit de programmer ses propres commandes. Un grand nombre de commandes est disponible pour administrer Linux (*ls* : Affiche le catalogue d'un répertoire, *cd* : Change le répertoire, *ps* : Affiche la liste des processus encore d'exécution, ...).

1.2.4- Les utilitaires de base

Nous citons ici quelques utilitaires Linux :

- *adduser, deluser* : pour l'ajout et la suppression des utilisateurs.
- *ifconfig* : pour la configuration des interfaces réseaux.
- *ping* : pour la vérification d'accessibilité d'une machine distante.

1.2.5- Les protocoles de communication

Linux supporte le protocole **TCP/IP** (*Transmission Control Protocol/Internet Protocol*), **PPP** (*Point to Point Protocol*), **SLIP** (*Serial Line IP*) et permet la construction des réseaux locaux complexes. Divers programmes clients et serveurs sont disponibles pour tous les types de connexions avec l'Internet (*telnet, rlogin, email, news, ftp, http, ...*).

1.2.6- Le système XWindow

Le système XWindow est une collection de fonctions et de protocoles qui permettent l'affichage à l'écran d'informations graphiques (dessins de points, de rectangles, etc.) et leur gestion à l'aide du clavier et de la souris. Le système XWindow a été développé par le *MIT* (*Massachusetts Institute of Technology*).

Le concept le plus important lié au système X est le serveur X qui représente l'interface entre le système XWindow et le matériel (carte graphique, écran, souris, ...).

Un serveur XWindow peut être vu comme l'implémentation du système XWindow. Le Serveur libre *XFree86* (sa version récente est la *X11R6*) est le plus connu et le plus utilisé.

1.2.7- L'environnement de bureau

Un environnement de bureau ne possède pas une définition définitive mais nous citons quelques éléments d'explication :

- Un environnement de bureau offre aux utilisateurs une interface homme/machine agréable permettant de réaliser les opérations de base d'une manière intuitive et facile
- Le texte des menus, des boutons et des boîtes de dialogue apparaissent dans un langage prédéfini (Anglais, Français, ...).
- L'environnement de bureau type actuel incorpore un paquetage d'utilitaires de manipulation uniforme pour ce qui est des opérations les plus fréquentes (éditeurs de texte, navigateurs, ...).
- Pour tous les composants ou programmes, une documentation en ligne abondante est disponible en un format facile à lire (*HTML*).

Au cours de l'histoire de l'*UNIX*, plusieurs tentatives ont été visées pour la manipulation des environnements de bureaux. Les plus connues

ont été deux systèmes commerciaux : *Motif* et *CDE* (*Common Desktop Environment*). Cependant, aucun de ces systèmes ne s'est préoccupé d'établir un véritable standard. La multitude des fonctions, longtemps, appréciée par les utilisateurs de *Microsoft Windows* ou de *Apple Mac OS* était ainsi refusée aux utilisateurs d'*UNIX* et aussi de Linux.

KDE (*K Desktop Environment*) et *GNOME* (*GNU Network Object Model Environment*) devraient changer cet état du fait qu'ils sont des environnements bureau Linux, faciles à utiliser, et par-dessus tout, ils sont libre, *KDE* et *GNOME* sont des projets relativement jeunes, débutés respectivement en octobre 1996 et août 1996 [Mic_99].

Les environnements de bureau ont favorisé le développement et l'utilisation des utilitaires Linux graphiques comme *Kuser* ou *UserDrake* pour la configuration des utilisateurs et des groupes.

1.3- Les distributions de Linux

Une distribution Linux est une combinaison du noyau actuel et des programmes additionnels. Elle permet de l'installer facilement.

Nous citons quelques distributions parmi les plus importantes :

- **Mandrake** : cette distribution est libre et est issue des travaux des chercheurs de *MandrakeSoft* qui ont adapté plusieurs programmes pour qu'ils ressemblent à ceux de *Windows*. Cette distribution constitue notre distribution de travail et sa version récente (9.0) est disponible sur le marché algérien.
- **Debian** : cette distribution est développée par beaucoup de volontaires, contrairement à certaines autres distributions qui ont été réalisées pour des objectifs commerciaux.
- **Redhat** : C'est une distribution commerciale de Linux de *Red Hat Software Inc.*

1.4- La licence d'utilisation de Linux

Linux est distribué sous la licence GNU *GPL* (GNU acronyme récursif de : *Gnu Not UNIX*, *GPL* acronyme de : *General Public Licence*) soutenue par la *FSF* (*Free Software Foundation*). Cela entraîne une petite confusion avec l'une des caractéristiques de Linux – Linux est libre –.

Le terme « libre » engendre souvent le terme « gratuit », puisque nous puissions l'obtenir gratuitement, tout du moins via Internet. En réalité, « libre » par-dessus tout, se réfère à la disponibilité de la totalité de son code source. La question qui se pose est :

- Que se passe-t-il si une société exploite le code source de Linux, l'étendue et vend le tout ?

La réponse est que cela est autorisé à condition que le code source du nouveau système soit librement disponible.

Ainsi, la licence *GPL* empêche les utilisateurs de compléter et de vendre des programmes *GNU* en faisant obstacle à toute modification publique. Il existe une variante de la licence *GPL* appelée *LGPL* (*Library GPL*) qui veut qu'une bibliothèque protégée par cette licence puisse aussi être utilisée par des produits commerciaux dont le code n'est pas librement disponible. Sans *LGPL*, les bibliothèques *GPL* ne pourraient être jamais utilisées dans des programmes commerciaux [Mic_99].

1.5- La documentation sur Linux

Il existe plusieurs sources de documentation sur Linux, parmi ces sources on trouve les pages des manuels *man* et *info*, les FAQ (*Frequently Asked Questions*), les documents *howto*, les livres de LDP (*Linux Development Project*) comme le guide d'utilisateur Linux (*Linux user's guide*) et aussi sur de nombreux sites web (<http://www.linux-France.org>, ...) ou sur les groupes de discussion Linux (comp.os.linux.*).

1.6- Conclusion

Le système d'exploitation Linux présente de nombreux avantages à cause de sa conception multitâche et multi utilisateur axée sur une utilisation en réseau, sa diversité d'applications qui sont moins coûteuses que leurs équivalentes sous autres systèmes d'exploitation comme *Windows*, *Mac OS*, *OS2*, *UNIX*, etc. Mais il possède encore des inconvénients propres à rebuter un bon nombre d'utilisateurs, par exemple, pour exploiter un nouveau périphérique, directement reconnu par un autre système d'exploitation comme *Windows*, il faut parfois patienter plusieurs mois à cause de la réticence de certains constructeurs à diffuser toutes les caractéristiques de leur matériel.

Linux apparaît aujourd'hui comme le concurrent le plus sérieux du système d'exploitation dominant (*Windows*) sur tous les segments du marché, des ordinateurs familiaux jusqu'aux énormes serveurs des entreprises. Il permet de maintenir une saine concurrence entre l'économie et l'informatique. C'est un système nécessaire à un équilibre des forces entre une logique de profit et une démarche de partage.

Après avoir brièvement présenter Linux et ses caractéristiques dans ce chapitre, nous allons aborder les notions réseaux de base nécessaires à la réalisation de notre outil dans le chapitre suivant.

Chapitre 2

Notions de base sur les réseaux

2.1- Définitions de base

Il est parfois difficile de parler des réseaux et de leurs protocoles, leur installation et leur configuration et administration sans que les termes utilisés soient bien compris et pour cela il vaut mieux commencer par des brèves définitions de ces termes [www_1] :

Réseau : ensemble d'ordinateurs et de périphériques reliés entre eux par un support de connexion et qui échangent des informations binaires.

Serveur : un ordinateur qui fournit des ressources partagées (fichiers, services, etc.) pour plusieurs clients.

Client : un ordinateur qui accède aux ressources partagées du serveur.

Support de connexion : support physique qui conditionne la façon dont les ordinateurs sont reliés (paires torsadées, fibres optiques, ...).

Carte réseau : carte connectée sur la carte mère de l'ordinateur et permettant de l'interfacer au support physique, c'est à dire aux lignes physiques permettant de transmettre l'information.

Protocole de communication : un ensemble de spécifications, de règles et des conventions à suivre dans un échange d'information entre les différents ordinateurs d'un réseau pour acheminer les données jusqu'au destinataire ou bien pour lui faire comprendre comment les utiliser.

Topologie d'un réseau : on distingue deux types de topologies :

- 1- **La topologie logique** : c'est la façon avec laquelle les données transitent dans le support de communication entre les différents nœuds du réseau. Une topologie logique est réalisée par un protocole d'accès comme *Ethernet*, *Token ring*, *FDDI (Fiber Distributed Data Interface)* et *PPP (Point to Point Protocol)*.
- 2- **La topologie physique** : c'est l'arrangement physique des éléments d'un réseau.

Il existe trois topologies physiques :

- 1- **La topologie en bus** : tous les ordinateurs sont reliés à une même ligne de transmission (bus) par l'intermédiaire des dispositifs matériels appelés Transceivers. Les principales topologies logiques utilisant cette topologie sont *Ethernet* et *Thin Ethernet*.
- 2- **La topologie en anneau** : tous les ordinateurs sont reliés à une ligne de transmission disposée en anneau fermée. Chaque ordinateur a son tour dans une boucle pour avoir le jeton qui lui donne le droit d'émettre des informations qui font le tour d'anneau. *Token ring* et *FDDI* sont deux topologies logiques de la topologie en anneau.
- 3- **La topologie en étoile** : tous les ordinateurs sont reliés aux différents ports d'un dispositif matériel (*hub*, *switch*, ...) qui gère la communication entre ses ports. *Twisted pair Ethernet* et *Fast Ethernet* utilisent cette topologie [Kar_99].

Équipements spéciaux :

- 1- **Un concentrateur (hub)** : permet de relier plusieurs ordinateurs dans un réseau local grâce à ses ports, il répercute toute information reçue dans l'un de ses ports simultanément aux autres ports.
- 2- **Un commutateur (switch)** : il diffère au concentrateur par le fait de ne répercuter l'information reçue qu'aux ports appropriés.
- 3- **Un pont (bridge)** : permet de relier des réseaux locaux fonctionnant sur le même système d'exploitation et utilisant le même protocole, il filtre les données transmises par les nœuds d'un réseau en laissant passer que celles destinées aux nœuds situés à l'opposé de lui.
- 4- **Un routeur (router)** : permet de relier des réseaux locaux utilisant des protocoles différents et d'assurer le routage des messages dans les réseaux étendus, choisir le chemin qu'un message va emprunter et le passage d'un message d'un réseau vers un réseau d'autre type.

2.2- Présentation du TCP/IP

Le protocole **TCP/IP** (*Transport Control Protocol/ Internet Protocol*) est toute une suite de protocoles de communication dans les réseaux recouvrant deux protocoles majeurs, **TCP** et **IP**, d'où vient son nom. Le protocole **TCP/IP** représente une façon avec laquelle les ordinateurs d'un réseau peuvent communiquer entre eux. La suite des protocoles **TCP/IP** était créée à l'origine d'un besoin militaire pour répondre à certains critères tels que le fractionnement des messages en paquets, utilisation d'un système d'adressage des ordinateurs, l'acheminement des données sur le réseau, etc.

2.2.1- l'architecture de TCP/IP

Le système de protocole **TCP/IP** a été décomposé en plusieurs modules effectuant chacun un rôle précis. De plus, ces modules effectuent des tâches les unes après les autres dans un ordre précis pour chaque machine dans le réseau et indépendamment du système d'exploitation, on a donc un système stratifié, c'est la raison pour laquelle on parle de modèle en couches.

Les deux modèles décrivant les couches de protocoles sont les suivants :

- le modèle **DoD** (*Department of Defense*) qui comprend 4 couches.
- le modèle **OSI** (*Open System Interconnection*) qui comprend 7 couches.

Nous allons décrire seulement le modèle **DoD** dans ce qui suit :

Le tableau suivant montre les descriptions des quatre couches composant le modèle **DoD**.

Couche	Description
4- Couche application	Englobe les applications standards du réseau.
3- Couche transport	Assure les connexions entre les différents hôtes sur le réseau.
2- Couche Internet	Fournit l'adresse logique pour l'interface physique, contrôle les messages transmis et assure leur routage.
1- Couche accès réseau	Représente la connexion physique avec les câbles les cartes réseaux, les protocoles d'accès au réseau.

Tableau-1 : Description des couches du modèle DoD.

La figure-2.1 montre le transfert de données selon le modèle DoD.

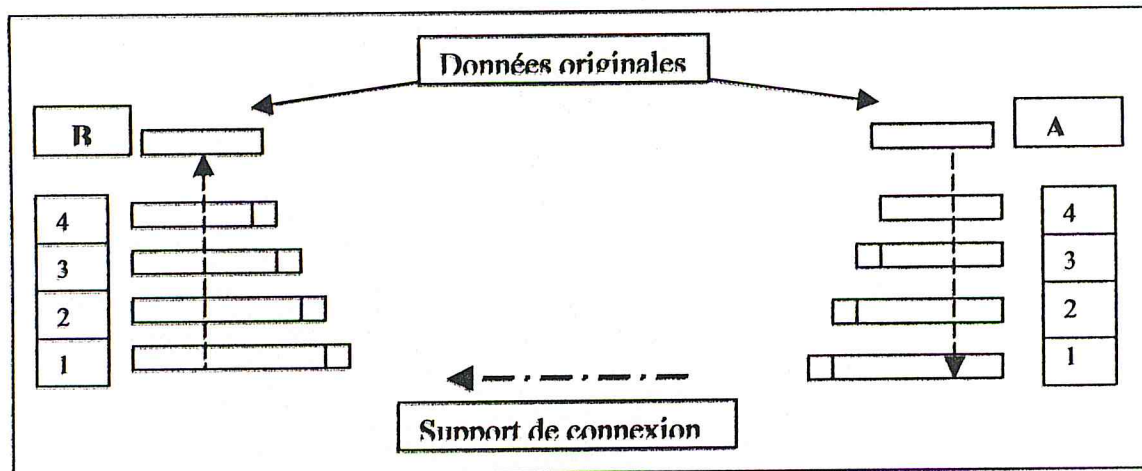


Figure-2.1 : Transfert de données selon le modèle DoD.

Lors d'une émission d'un message par un hôte, il traverse chacune des couches du modèle **DoD**, chaque couche ajoute un en-tête au paquet de données reçu. A la réception, chaque couche lit le paquet de données reçu et élimine l'en-tête approprié, et le message prend son état original.

2.2.2- Hiérarchie de l'implémentation de TCP/IP

La famille des protocoles **TCP/IP** comprend un ensemble performant de services qui peuvent utiliser une variété de technologies réseaux (réseaux étendus ou locaux, les ondes radio, les satellites,...). Les modules du **TCP/IP** et les relations entre ces modules forment la hiérarchie de l'implémentation **TCP/IP**. La figure-2.2 montre cette hiérarchie :

Couche 4	HTTP	FTP	SMTP	DNS	IMAP	POP3	NNTP	...
Couche 3	TCP				UDP			
Couche 2	IP			ICMP		RIP	ARP, RARP, ...	
Couche 1	Ethernet, Token Ring, FDDI, ATM PPP, ...							

Figure-2.2: La hiérarchie d'implémentation de TCP/IP.

Cette figure montre les quatre modules suivant :

- Le module 1 : représenté par la couche 1 et se charge des protocoles d'accès réseaux.
- Le module 2 : représenté par la couche 2 et se charge des protocoles **IP** (pour l'acheminement des paquets depuis l'expéditeur vers le destinataire), **ICMP** (*Internet Control Message Protocol*, pour la gestion des messages des erreurs), **RIP** (*Routing Information Protocol*, pour la détermination des routes des paquets), **ARP** (*Address Resolution Protocol*, pour la translation des adresses **IP** en adresses physiques) et **RARP** (*Reverse ARP*).
- Le module 3 : représenté par la couche 3 et se charge des protocoles **TCP** (qui s'appuie sur **IP** pour le transfert des données

entre l'expéditeur et le destinataire en établissant des connexions) et **UDP** (*User Datagram Protocol*, pour le transfert de données entre l'expéditeur et le destinataire sans connexions).

- Le module 4 : représenté par la couche 4 et se charge des services réseaux (**DNS**, **HTTP**, **FTP**, ...).

2.3- L'adressage IP

Chaque machine connectée à un réseau **TCP/IP** doit être identifiée uniquement d'une façon ou d'une autre. Sans un identificateur unique, on ne saurait pas comment apporter des messages à une machine donnée. De plus, s'il y avait plus d'une machine avec le même identificateur, on ne saurait jamais à laquelle on va envoyer le message.

Dans le cadre du standard actuel – **IP** version 4 – les adresses sont codées sur 32 bits (les adresses du **IPv6**, son déploiement sera en futur proche, sont codées sur 128 bits). Ainsi, tout ordinateur sur Internet ou sur un réseau **TCP/IP** se voit attribuer une adresse de type **a.b.c.d** (a, b, c et d sont des nombres compris entre 0 et 255).

L'adresse **IP** consiste en deux parties : le numéro du réseau (*netid*) et le numéro de l'hôte dans ce réseau (*hostid*). En utilisant deux parties dans les adresses **IP**, les machines sur des réseaux différents peuvent avoir le même numéro d'hôte, mais puisque le numéro du réseau est différent, les machines sont uniquement identifiées.

Les adresses **IP** sont assignées sur la base de la dimension de la compagnie ou de l'organisation pour prévoir le maximum de flexibilité.

Les adresses **IP** appartiennent à des classes bien définies dont les plus importantes sont :

- Classe A : le *netid* varie de 0 à 126.

(un réseau de classe A peut contenir jusqu'à 2^{24} hôtes).

- Classe B : le *netid* varie de 128 à 191.
(un réseau de classe B peut contenir jusqu'à 2^{16} hôtes).
- Classe C : le *netid* varie de 192 à 223.
(un réseau de classe C peut contenir jusqu'à 2^8 hôtes).

En pratique, on utilise le masque réseau (ou le masque du sous réseau s'il s'agit d'un sous réseau) qui succède toujours l'adresse IP pour déterminer rapidement le *hostid* et le *netid* en appliquant un « ET » logique entre l'adresse IP et le masque réseau (le masque réseau de la classe A est 255.0.0.0, de la classe B est 255.255.0.0 et celui de la classe C est 255.255.255.0).

Nous citons quelques adresses IP particulières dans ce qui suit :

- **<netid>.0** : les bits du champ *<hostid>* sont tous à 0.
Elle indique l'adresse du réseau lui-même (tous ses nœuds).
- **<netid>.+1** : les bits du champ *<hostid>* sont tous à 1.
Elle indique l'adresse de diffusion dirigée (*directed broadcast*).
- **+1.+1** : tous les bits de l'adresse IP sont à 1.
Elle indique l'adresse de diffusion limitée (*limited broadcast*).
- **(0.0)** : tous les bits de l'adresse IP sont à 0.
Le netid vaut 0, signifie « le réseau d'ici », le hostid vaut 0, signifie le « le nœud d'ici » sur le réseau, elle est utilisée lorsqu'un nœud essaye de déterminer sa propre adresse IP.
- **(127.x.y.z)** : x, y et z sont compris entre 0 et 255. Cette adresse est appelée l'adresse de bouclage (*loopback*), tout paquet envoyé par une application TCP/IP vers une adresse de ce type a pour conséquence le renvoi de ce paquet à cette application sans que le paquet atteigne le support du réseau.

- **Les adresses IP de la RFC 1918** : ce sont les adresses IP non routables sur Internet, elles sont utilisées pour les réseaux locaux qui sont liés à Internet via un routeur (10.0.0.0→10.255.255.255, 172.16.0.0→172.31.255.255, 192.168.0.0→192.168.255.255).

2.4- La notion de port d'écoute

Lorsque les applications utilisant les services réseaux tels que **HTTP, FTP, ...** doivent communiquer sur Internet ou sur un réseau **TCP/IP** elles utilisent un paramètre supplémentaire en plus de l'adresse **IP** de l'expéditeur et du destinataire : *le numéro de port du service* (chaque service a son propre numéro de port). Chaque paquet transmis sur le réseau contient un champ indiquant le numéro de port du service demandé par l'application.

Certains numéros de port sont usuels (*well-known*, compris entre 0 et 1023 et sont réservés à l'*IANA, Internet Address Network Authority*), d'autres sont enregistrés (certains numéros supérieurs à 1023) et d'autres sont affectés dynamiquement (les numéros de port non enregistrés). Par exemple le port **TCP 80** est un numéro de port usuel et qui est utilisé par le protocole **HTTP** [Kar_99].

2.5- Le domaine de noms

TCP/IP, comme nous avons vu, utilise l'adresse **IP** qui est un numéro sur 32 bits pour acheminer un datagramme à une destination donnée. Il est utile d'oublier ces numéros d'adresses et au lieu d'utiliser des noms communs dans les différentes applications basées sur **TCP/IP** parce que les noms sont plus faciles à se souvenir.

Il y a plusieurs méthodes utilisées pour supporter l'utilisation du domaine de noms en **TCP/IP**. La plus ancienne est d'employer un fichier

texte sur la machine émettrice qui avait des noms et des adresses correspondantes (*/etc/hosts* sur une machine Linux), mais la plus efficace est d'utiliser le service de noms **DNS** (*Domain Name Service*). Nous revenons en détail sur le service **DNS** dans le chapitre suivant.

2.6- Conclusion

Une intervention sur la gestion, l'administration ou la configuration des réseaux sous Linux ou n'importe quel produit *UNIX*, se répercute inévitablement sur les modules **TCP/IP**.

Les modules **TCP/IP** sont construits hiérarchiquement à partir d'une suite de protocoles disposés en couches, ces modules sont connus souvent par les couches de protocoles.

Dans le chapitre suivant nous allons décrire les services réseaux à configurer par notre outil leurs fonctionnements et leurs implémentations et techniques de configuration sous Linux d'une manière un peu détaillée.

Chapitre 3

Etude des services réseaux à configurer

3.1- Introduction

Linux offre une implémentation **TCP/IP** complète et intégrée comme un module du noyau pour donner des fonctions de communication très rapides. Le protocole **TCP/IP** est le plus utilisé sur Internet et le plus connu dans le monde des machines *UNIX*, et tout ce qu'on a besoin pour créer un réseau **TCP/IP** ou ajouter une machine à un réseau **TCP/IP** existant est une carte réseau pour chaque machine dans le réseau et quelques modifications nécessaires de certains fichiers systèmes.

Dans ce chapitre nous allons voir la configuration de base de chaque machine Linux tournant **TCP/IP** et les différents détails des services réseaux à configurer par notre outil.

3.2- Configuration de base de TCP/IP sous Linux

La configuration de **TCP/IP** sous Linux consiste en plusieurs étapes dont certaines sont faites automatiquement durant le processus d'installation et prises en charge par le fichier système « */proc* » qui fait partie du noyau.

L'étape importante est la configuration des interfaces réseaux :

- L'interface de bouclage (*loopback interface*) : elle est configurée automatiquement durant la phase d'installation de Linux et doit exister dans tout machine Linux typique tournant **TCP/IP** dans un réseau ou seule, son adresse **IP** est 127.0.0.1.
- L'interface Ethernet (*Ethernet interface*) : le processus de configuration d'une interface Ethernet est de l'activer par la commande : « *ifconfig eth0 adresseIP* », *eth0* désigne le nom du pilote de la première interface *Ethernet* (on peut avoir des machines à

plusieurs cartes réseaux : eth0, eth1, etc.) et l'ajouter à la table de routage.

De plus des interfaces réseau, il y a plusieurs fichiers à configurer dont la plus part sont semblables dans toutes les dérivés d'UNIX.

Nous examinons brièvement les fichiers les plus importants [Tim_97] :

- Le fichier */etc/hosts* : contient une liste d'adresses IP accompagnées des noms d'hôtes correspondants. C'est en effet, une liste de toutes les machines du réseau les plus utilisées, on les désigne par leurs noms d'hôtes et le système se chargera de rechercher leurs adresses IP associées en évitant d'exécuter un service de noms.
- Le fichier */etc/networks* : contient une liste de noms et adresses IP correspondants du réseau actuel et de ceux qui lui sont connectés.
- Le fichier */etc/protocols* : identifie tous les protocoles de transmission disponibles sur le système avec leurs numéros caractéristiques.
- Le fichier */etc/services* : identifie les différents services réseaux existant.

3.3- Le service de noms

3.3.1-Introduction

Sur un réseau TCP/IP, quand on veut adresser une machine, il est souvent pratique de mémoriser son nom symbolique plutôt que son adresse IP. Comme nous avons vu précédemment, Linux utilise le fichier */etc/hosts* pour résoudre les noms symboliques en adresses IP. Si un nom de machine ne figure pas dans ce fichier nous ne pourrons plus émettre de données à cette machine. Ce fichier doit être téléchargé périodiquement par les différents ordinateurs du réseau

pour la résolution locale des noms. Mais avec la croissance des réseaux et l'occurrence de nombreux changements, l'administration centralisée d'un espace de dénomination en utilisant ce fichier devient très difficile à réaliser et pose vraiment un problème.

La solution au problème posé devra être distribuée et permettant un espace de dénomination hiérarchisé, une extensibilité des types de données, une taille illimitée et des performances acceptables. Cette solution consiste à utiliser le service **DNS**, (*Domain Name Service*), qui est un système de gestion de base de données hiérarchiques et distribuées, basé sur un concept client/serveur. Le but de la base de données **DNS** est de traduire les noms d'ordinateurs en adresses **IP**.

3.3.2- L'organisation du DNS

Les spécifications de base de **DNS** sont définies dans les RFCs (Requests For Comments) 974 et 1034 qui l'organisent en [www_2] :

- L'espace de noms de domaines et les enregistrements de ressource
- Les serveurs de noms
- Les résolveurs de noms

L'espace de noms et les domaines de noms :

L'espace de noms **DNS** est structuré sous la forme d'un arbre logique inversé. Chaque nœud de l'arbre possède un nom pouvant aller jusqu'à 63 caractères. La racine de l'arbre possède le nom réservé "" (null). Chaque nœud, appelé un *domaine*, est un élément de l'espace de noms. Les domaines peuvent être subdivisés au niveau des nœuds à l'intérieur d'un domaine en *sous-domaines* pour des besoins d'administration ou de répartition de la charge. *Le nom de domaine* identifie la position du domaine dans la hiérarchie **DNS**. *Le nom de*

domaine complet FQDN (*Fully Qualified Domain Name*) identifie la machine par rapport à la racine. Les noms de domaine et les noms de domaine complets sont créés en combinant les noms des nœuds depuis le nœud de domaine spécifié jusqu'à la racine et en séparant chaque nom d'un nœud de son successeur par un point (.).

La figure-3.1 présente un exemple extrait d'un espace de noms :

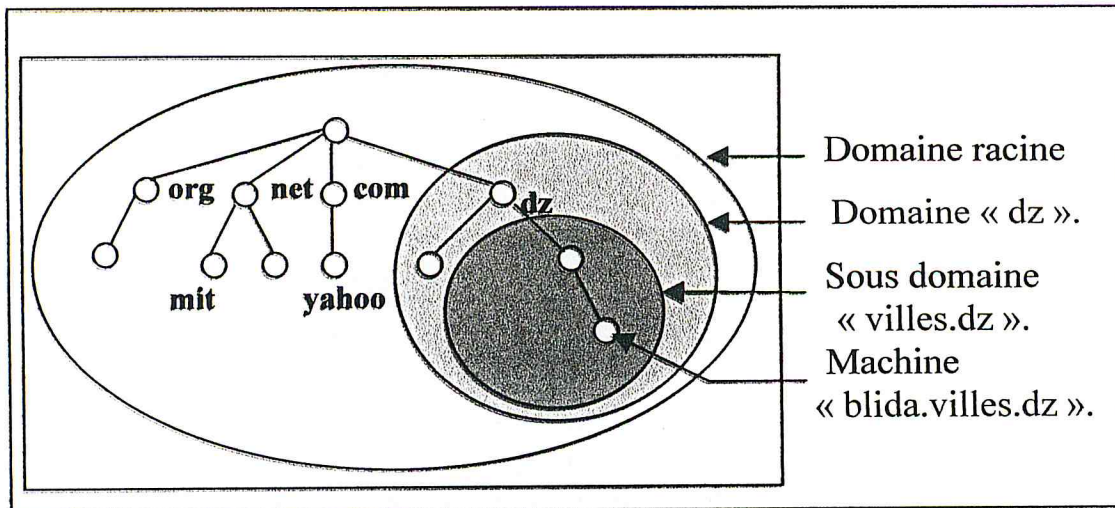


Figure-3.1 : Exemple extrait d'un espace de noms.

Les zones :

Une *zone* est définie comme une portion de l'espace de noms DNS pour lequel existent des enregistrements de la base de données stockés dans un fichier particulier. Un serveur DNS pourrait être configuré pour gérer une ou plusieurs zones. Chaque zone représente un domaine particulier. Une zone parente peut déléguer une zone fille à un ou plusieurs serveurs de noms.

Les enregistrements de ressource (Ressource Records ou RR) :

Un nom de domaine définit un nœud dans l'espace de noms DNS. Chaque nœud est composé d'un jeu d'enregistrements de ressource (Ressource Records ou RR) qui constituent l'information de ressource

associée avec le domaine. Les enregistrements de ressource sont stockés dans des fichiers appelés *fichiers de zones*.

Les enregistrements de ressource contiennent les informations suivantes :

- Propriétaire (*Owner*) : c'est le nom du domaine de l'enregistrement.
- Type : c'est le type de ressource représenté par l'enregistrement.
- Classe : Définit la famille de protocoles en usage (*IN* pour l'Internet).
- TTL (Time to Live) : c'est la durée de vie pendant laquelle un serveur de noms ou un résolveur conserve son entrée dans son cache.
- RDATA : C'est la ressource de données (dépend de Type).

Nous donnons un exemple d'un RR :

« **linux.com A IN 3600 192.120.2.50** »

le propriétaire (linux.com.), le type (A, adresse), la classe (IN), le TTL (3600=1 heure) et le RDATA (adresse IP linux.com). Si le TTL n'est pas précisé, une valeur par défaut est utilisée.

Il existe différents types d'enregistrements de ressource. La RFC 1034 définit sept types d'enregistrements de ressource suivants:

Nom	Description
SOA	Identifie le début d'une zone d'autorité (Start Of Authority)
A	Une adresse d'hôte
NS	Le serveur de nom faisant autorité pour le domaine (Name Server)
PTR	Un pointeur vers une autre partie de l'espace de noms du domaine
CNAME	Le nom autorisé d'un alias
MX	Identifie un échangeur de messagerie pour le domaine (Mail eXchanger)
HINFO	souvent utilisé pour identifier la CPU et le système d'exploitation utilisé par l'hôte (Host Information)

Tableau-2 : Les types d'enregistrements de ressource de la RFC 1034.

En plus de ces enregistrements, plusieurs autres sont définis dans d'autres RFCs (RFC 1035, 1183, 1189, ...) et dont certains sont en phase d'expérimentation.

Les serveurs de noms (name servers) :

Les serveurs de noms ont pour rôle de répondre à des requêtes des clients. Ils contiennent les informations requises dans leurs enregistrements de ressource ou possèdent un pointeur vers d'autres serveurs de noms qui peuvent répondre à la requête. Un serveur de nom pour une zone donnée peut être:

- Primaire : qui obtient ses zones depuis un fichier local.
- Secondaire : obtient les informations concernant ses zones depuis un serveur de noms maître (primaire ou secondaire) à travers le réseau.

Les résolveurs de noms (name resolver) :

Ce sont des programmes clients et qui extraient des informations depuis les serveurs de noms grâce à des requêtes DNS.

3.3.3- Le fonctionnement du DNS

DNS fonctionne selon un mode client/serveur et utilise le port UDP 53. Les résolveurs de noms (clients) envoient d'abord des requêtes UDP vers les serveurs de noms, un serveur de noms questionné par un client moyennant une requête DNS peut ne pas connaître la réponse mais peut savoir quel serveur peut répondre. Il y a alors deux possibilités : soit le serveur renvoie au client la référence du serveur qui sait répondre (mode itératif), soit le serveur poursuit la recherche lui-même auprès de ce serveur et fournit au client sa réponse (mode récursif).

3.3.4- La configuration du serveur DNS

L'implémentation Linux du protocole DNS est le BIND (Berkeley Internet Name Domain) qui fût originellement développée à l'université de Berkeley pour le système d'exploitation UNIX BSD version 4.3.

Mandrake Linux implémente le BIND sous le démon « *named* » qui est généralement configuré pour démarrer avec le système grâce au fichier « *rc* » (la version du serveur BIND que nous travaillons sur elle est la version 9.2.1).

Au plus du fichier */etc/hosts* le DNS utilise [www_3][www_4]:

- Le fichier */etc/host.conf* : qui est utilisé par le résolveur de noms. Il contient en général la ligne « *order hosts, bind* » qui indique au résolveur de noms qu'il doit d'abord vérifier le fichier */etc/hosts*, puis le serveur de noms (*bind*) lorsqu'il tente de résoudre un nom.
- Le fichier */etc/resolv.conf* : qui est aussi utilisé par le résolveur de noms (*name resolver*) pour savoir quel serveur de noms à contacter. Il contient l'adresse IP du serveur de nom primaire et secondaire à contacter lorsqu'il tente de résoudre un nom
- Le fichier */etc/named.conf* : qui est le fichier principal, il contient les propriétés du serveur et la liste des de zones utilisées par le serveur. Un exemple extrait de ce fichier avec une brève description de ses entrées est le suivant:

```
options {
directory "/var/named";           // répertoire des fichiers de configuration
forwarders {212.52.129.34; };      // les adresses IP des serveurs redirecteurs
allow-query { adresse IP ;... ; }; // restreindre utilisation du serveur.
allow-transfer { adresse IP ;... ; }; // restreindre le transfert de zones.
allow-recursion { adresse IP ;... ; }; // restreindre la résolution récursive.

... };

zone "." {                          // la zone du domaine racine d'Internet
type hint;
file "named.ca";
};

zone "algeria.dz" {                 // la zone du domaine « alg.dz »
type master;                       // il s'agit d'une zone primaire
file "zone/alg.dz";
};
```


- Les fichiers de zones : ils contiennent les enregistrements de ressources. Un exemple extrait de l'un de ces fichiers avec une brève description de ce qu'il contient est le suivant :

```
$TTL 3D      // temps de vie par des enregistrements de ressource
@           IN      SOA  algeria.dz      root.algeria.dz
(           1997022700 ; Serial (numéro de série de la zone)
 28800      ; Refresh (temps de rafraichissement de la zone en secondes)
14400      ; Retry  (temps d'essai, après l'échec d'un rafraichissement de la zone)
3600000    ; Expire  (temps d'expiration des données de la zone)
86400 )     ; Minimum (temps de vie minimum)
    NS      IN      alg      ; un enregistrement de nom de serveur (NS)
    MX      IN      10 alg    ; un enregistrement d'échangeur de messagerie (MX)
localhost  A      IN      127.0.0.1 ; enregistrement d'adresse (A)
algeria A      IN      200.200.200.1
www CNAME    IN      alg      ; enregistrement de nom canonique (CNAME)
blida A      IN      200.200.200.9
alger A      IN      200.200.200.16
```

Le redémarrage du serveur de noms (*named*) se fait en cherchant le numéro du processus du serveur de noms dans le fichier « */var/run/named/named.pid* » ou en exécutant la commande « *ps -A* » puis en exécutant la commande « *kill -HUP numéro_processus* ».

Le test du bon fonctionnement se fait en utilisant l'utilitaire « *dig* » avec le nom de domaine à chercher et le type de l'enregistrement associé comme option.

3.4- Le service web

3.4.1- Introduction

Le service web est assuré par le protocole **HTTP** (HyperText Transfer Protocol) qui est le protocole de communication du World Wide Web, service permettant d'accéder à des pages Web qui sont des écrites en langage **HTML** (*HyperText Mark-up Language*) et peuvent contenir

du texte, des images (statiques ou dynamiques), des séquences vidéo et des hyper-liens (des liens vers d'autres pages Web) , ce protocole fonctionne selon un modèle client/serveur permettant ainsi de transférer des pages Web depuis un serveur Web vers un client (un navigateur Web comme Netscape ou Internet Explorer par exemple). Ces pages web sont localisées grâce à des chaînes de caractères *ASCII* appelées URLs (*Uniform Ressource Locator*).

La structure d'un *URL* est la suivante :

protocole://nom_serveur/chemin_d'accès/document :numéro_port

Par exemple, `http://www.linux-France.org/prj/edu/archinet.html` est un URL dont le numéro de port est omis car **HTTP** utilise par défaut le port 80. Les spécifications de la dernière version actuelle du protocole **HTTP** (version 1.1) se trouvent dans la RFC 2616.

3.4.2- Le fonctionnement de HTTP

HTTP fonctionne selon un modèle client/serveur et utilise le port **TCP** numéro 80. le client désirant obtenir une page web par exemple, envoie des requêtes **HTTP** au serveur **HTTP**, ce dernier se charge de leur traitement renvoie les répliquions associées munies des réponses **HTTP** décrivant le déroulement du traitement de ces requêtes.

Une requête se décompose en trois composantes [www_5] :

- La ligne de requête : elle comprend à son tour trois éléments séparés par des espaces, la méthode qui doit être appliquée, l'URL et la version du protocole **HTTP** utilisé par le client (`http/1.1`).
- Les champs d'en-tête de requête : ils contiennent des informations supplémentaires sur la requête et le client comme l'identification du navigateur Web (Web browser), le système d'exploitation,...

- Le corps de la requête : il contient les données transférées par la requête.

Le tableau suivant montre les méthodes de requêtes **HTTP** disponibles :

GET	demande de lire une page située à l'URL spécifié
HEAD	demande de lire l'en-tête d'une page située à l'URL spécifié
PUT	demande de mémoriser une page à l'URL spécifié
POST	envoie une page à l'URL spécifié
DELETE	supprimer une page située à l'URL spécifié
LINK	relie deux ressources existantes
UNLINK	efface un lien entre deux ressources

Tableau 3 : Les méthodes de requêtes HTTP.

Une réponse **HTTP** est toujours envoyée par un serveur Web à un client et elle se décompose en trois composantes :

- La ligne de statut : elle précise trois éléments : la version du protocole utilisé, le code de la réponse et un texte explicatif
- Les champs d'en-tête de réponse : ils contiennent des informations supplémentaires sur la réponse et le serveur.
- Le corps de la réponse : il contient les données demandées.

Les codes de réponse sont affichés par les navigateurs Web lorsque ces derniers n'arrivent pas fournir les pages demandées. Par exemple le code 404 indique une page web non trouvable par le serveur.

3.4.3- La configuration du serveur HTTP

Mandrake Linux utilise le serveur Web Apache qui est un logiciel libre (GNU) produit par une coopération organisée entre les volontaires qui ont continué à améliorer le serveur **httpd** (le premier serveur Web).

Apache reprend tous les paramètres de configuration de **httpd** et son implémentation sous Linux est aussi réalisée par le démon **httpd** qui est généralement configuré pour lancer automatiquement au démarrage du système (la version du serveur Apache que nous travaillons sur elle est la version 1.3.26).

La configuration du serveur Apache s'effectue en modifiant les fichiers « *httpd.conf*, *commonhttp.conf* et *Vhosts.conf* ». Le fichier principal qui englobe tous les autres fichiers est le fichier : *httpd.conf*.

Ces fichiers se trouvent dans le répertoire « */etc/httpd/conf* » et sont constitués de dizaines de lignes contenant des directives de configuration, leurs paramètres, leurs options ainsi que leurs commentaires (ces directives peuvent être imbriquées l'une dans l'autre).

Le tableau ci-dessous montre quelques directives de configuration **HTTP** avec leurs descriptions [www_6] :

Directive	Description
DocumentRoot /var/www/html	Fixe la racine du serveur Web(le répertoire de base où sont cherchées par défaut les pages html lorsque l'URL ne comporte pas de chemin de répertoire).
ServerRoot /etc/httpd	Il s'agit du répertoire où le serveur trouvera son répertoire des fichiers de configuration. On trouve dans <i>/etc/httpd</i> , un lien vers <i>/var/log/httpd/access_log</i> , le fichier-journal des accès aux ressources, réussis ou non.
<Virtualhost adresse_IP> ServerName nom ... </Virtualhost>	Cette directive permet de définir des hôtes virtuels avec toutes ses propriétés (autres directives) qui sont incluses entre les deux délimiteurs (un hôte peut être défini comme une instance du serveur qui s'exécute sur la même machine où est installé le serveur ou une autre machine dans le réseau avec ses propres options.

Listen adresseIP : N°port	Cette directive permet de lancer une instance du serveur web dans la machine désignée par son adresse IP sur le port donné.
<Directory rep> Options ExecCGI, Indexes, ... Allow all </Directory>	Cette directive permet de spécifier les propriétés (<i>ExecCGI</i> pour l'exécution des scripts CGI, <i>Indexes</i> pour l'affichage d'un répertoire si un fichier par défaut n'y est pas trouvé, Autoriser l'accès à tous les utilisateurs, ...) d'un répertoire nommé <i>rep</i> (situé dans le répertoire racine).

Tableau 4 : Description de quelques directives de configuration HTTP.

Un exemple extrait du fichier « *httpd.conf* » est le suivant :

```
# <VirtualHost> definition. These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
# Port: The port to which the standalone server listens. Certain
# firewall products must be configured before Apache can listen to a
# Port 80
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents.
ServerAdmin admin@localhost
# ServerName allows you to set a host name which is sent back to
# clients for your server if it's different than the one the program
# would get (i.e., use "www" instead of the host's real name).
ServerName localhost
```

Le redémarrage d'Apache se fait en lançant l'utilitaire
« *apachectl* » avec l'option « *restart* ».

Le teste de son bon fonctionnement se fait en lançant des requêtes
au serveur web en demandant les hôtes virtuels sur leurs ports configurés
ou les pages web situées dans les répertoires configurés.

3.5- Le service FTP

3.5.1- Introduction

Ce service est assuré par le protocole **FTP** (*File Transfer Protocol*), le protocole de transfert de fichiers entre les machines des réseaux **TCP/IP**. La mise en place du protocole **FTP** date de 1971, date à laquelle un mécanisme de transfert de fichiers (décrit dans le RFC 141) entre les machines du MIT (*Massachusetts Institute of Technology*) avait été mis au point. De nombreux RFCs ont ensuite apporté des améliorations au protocole de base, mais les plus grandes innovations datent de juillet 1973. Le protocole **FTP** est actuellement défini par le RFC 959.

Le protocole **FTP** a pour objectifs de permettre:

- Un transfert de fichiers d'une manière efficace
- Une indépendance des systèmes de fichiers des machines effectuant le transfert des fichiers
- Un partage des fichiers entre des machines distantes

3.5.2- Le fonctionnement de FTP

Le protocole FTP fonctionne selon un modèle client/serveur, fait l'authentification des clients par le serveur et utilise deux ports TCP, le port numéro 20 pour le transfert des données et le port numéro 21 pour le transfert des commandes. Un client doit être un vrai utilisateur avec un nom d'utilisateur et un mot de passe pour la machine où s'exécute le serveur **FTP**. Certains systèmes utilisent **FTP Anonyme** (**Anonymous FTP**) qui permet à tout utilisateur d'accéder à ces systèmes en fournissant *guest* ou *anonymous* comme nom d'utilisateur et l'adresse électronique ou le nom de machine locale comme mot de passe. Lors d'une connexion **FTP**, deux canaux de transmission sont ouverts :

- Un canal pour les commandes (canal de contrôle)
- Un canal pour les données

Nous décrivons la procédure générale d'une session **FTP** :

Le client **FTP** (généralement implémenté comme une fonctionnalité du navigateur Web tel que Netscape ou bien comme un logiciel spécial) envoie une requête à un serveur **FTP** pour obtenir son fichier désiré.

La figure-4 montre la relation entre un client **FTP** et un serveur **FTP** [www_7].

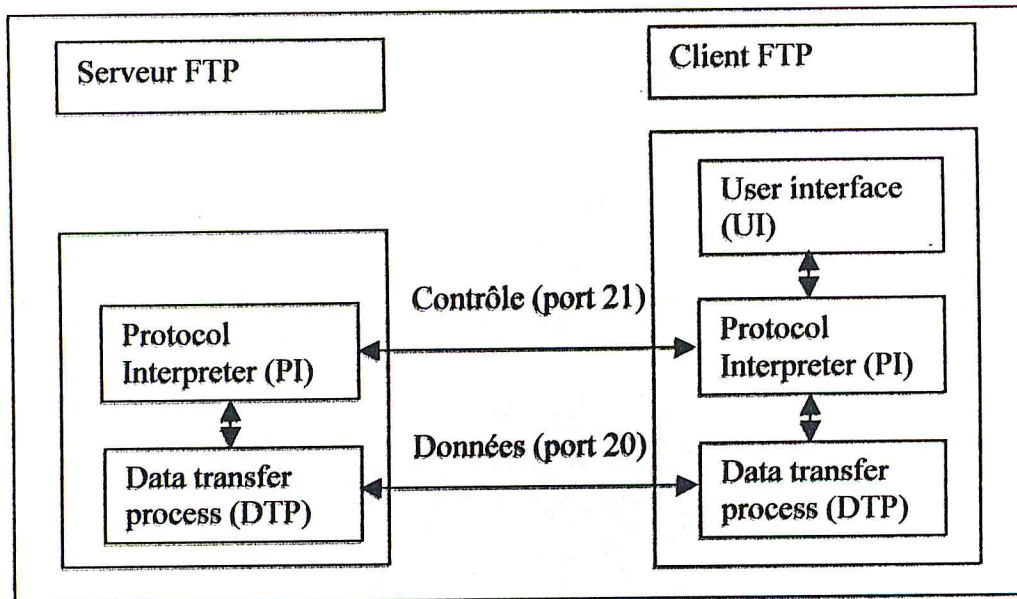


Figure-3.2 : La relation client-serveur **FTP**.

D'après la figure-3.2, le PI client reçoit les demandes en provenance de l'interface utilisateur, les transforme en commande **FTP** et les transmet au PI serveur. Lors de la réception d'une commande **FTP** de transfert de fichier, le PI concerné la transmet au DTP (dont il a la charge) pour ouvrir la connexion avec le DTP de l'autre partie et gérer le canal de données.

Le protocole **FTP** utilise son propre langage de commandes défini par des mots clefs de quatre caractères appelés les commandes **FTP** internes (**FTP** dispose aussi de commandes utilisateur dont nous allons voir plus tard). Les commandes internes servent à établir et à maintenir la connexion **FTP** (alors que les commandes utilisateur permettent d'effectuer des opérations à l'aide de cette connexion).

Il y a trois groupes de commandes [www_7] :

1. les commandes de contrôle d'accès (USER, PASS, ACCT, ...).
2. les commandes de paramétrage de transfert (PORT, TYPE, ...).
3. les commandes de service **FTP** (RETR, STOR, REST, ...).

Les réponses **FTP** sont toujours envoyées par le serveur **FTP** au client pour assurer la synchronisation entre eux et sont représentées par des codes de trois chiffres. Le tableau suivant montre quelques codes de réponses **FTP**, par exemple le code de réponse 200 indique une opération réalisée avec succès.

Le tableau suivant montre quelques commandes utilisateur **FTP** :

Commande	Description
ascii	mettre le mode de transfert en ASCII
binary	mettre le mode de transfert en binaire
cd	Changer le répertoire, dans le coté serveur
close	Terminer la connexion
del	Supprimer un fichier, dans le serveur
dir	Affiche le contenu du répertoire courant, dans le serveur
get	Copier un fichier depuis le serveur sur la machine locale

Tableau 5 : Description de quelques commandes utilisateurs **FTP**.

3.5.3- La configuration du serveur FTP

Mandrake Linux utilise par défaut le serveur FTP *Proftp* et l'implémente sous le démon « *proftpd* » (la version du serveur Proftp que nous travaillons sur elle est la version 1.2.5).

La configuration de *proftpd* s'effectue en modifiant le fichier « *proftpd.conf* » qui se trouve dans le répertoire « */etc* ». Le démon *proftpd* est toujours lancé soit à la demande d'un client par le démon de démarrage *inetd* soit au démarrage du système.

Le fichier de configuration « *proftpd.conf* » contient une série d'instructions de configuration peut être vue comme une série de commandes imbriquées. Certaines commandes s'appliquent à l'ensemble du serveur, mais il est possible de préciser une configuration répertoire par répertoire.

Le fichier *proftpd.conf* se divise en plusieurs parties, qui ne sont pas toutes nécessaires. Nous avons donc plusieurs contextes de configuration qui contiennent les directives de configuration de *proftpd* : *<server config>*, *<Global>*, *<VirtualHost>*, *<Limit>*, *<Directory>*, *<Anonymous>*.

Nous pouvons avoir un fichier « *proftpd.conf* » avec seulement le contexte : *<server config>*. Si le contexte de configuration n'est pas explicité alors il est *<server config>*.

Le tableau ci-dessous montre quelques directives de configuration de *proftpd* [www_8]:

Directive	Description
<Global>	Ce contexte peut être contenu dans un contexte
DefaultRoot ~	<i><server config></i> ou <i><VirtualHost></i> et contient
AllowOverwrite yes	certaines directives (<i>DefaultRoot</i> permet de restreindre
MaxClients 3	l'accès de tout utilisateur à son répertoire <i>/home/user</i> ,

MaxClientsPerHost 1 UseFtpUsers on AllowForeignAddress on ServerIdent on "ProFTP Server" AccessGrantMsg "Salut!" </Global>	AllowOverwrite permet remplacer les anciens fichiers par les nouveaux, MaxClients indique le nombre maximum des utilisateurs qui peuvent connecter simultanément, MaxClientsPerHost indique le nombre max des clients connectant sous le même nom d'utilisateur, FtpUsers permet l'utilisation du fichier <i>/etc/ftpusers</i> pour savoir qui a le droit d'accès au serveur, ...
<VirtualHost ftp.serveur.com> ServerName serveur_virtuel Port 46000 Maxclients 3 MaxClientsPerHost 1 DefaultRoot ~ AccessGrantMsg "Salut !" </VirtualHost>	Un hôte virtuel ftp peut être défini comme une instance du serveur qui s'exécute sur la même machine où est installé le serveur principal, sur un port différent, sous un autre nom de serveur et avec des propriétés différentes. Par exemple cet hôte virtuel nommé <i>serveur_virtuel</i> , utilise le port 46000 avec un maximum de 3 utilisateurs connectant simultanément et un client qui connecte sous le même nom d'utilisateur, un message d'accueil « Salut ! ».

Tableau 6 : Description de quelques directives de configuration FTP.

Un exemple extrait de ce fichier est le suivant :

```

ServerName      "Proftpd Linux Service"
ServerType      standalone
DefaultServer   on
AllowStoreRestart on
# Port 21 is the standard FTP port.
Port            45000
User            nobody
Group           nogroup
# Nombre maximum de clients
#MaxClients     3
# Number of Max Clients per host
#MaxClientsPerHost 1
# Nombre maximums de tentatives de login
MaxLoginAttempts 3
# Message d'accueil après une connexion réussie
AccessGrantMsg  "Bienvenue %u chez moi !"

```


Le redémarrage du serveur **FTP** se fait en cherchant son numéro de processus dans le fichier « /var/run/proftpd.pid » et en exécutant la commande « kill -HUP numéro_processus ».

Le teste de bon fonctionnement du service **FTP** se fait en lançant des requêtes vers le serveur en demandant le serveur principal ou bien les hôtes virtuels configurés sur les différents ports.

3.6- Le Service de messagerie électronique

3.6.1- Introduction

Le service de messagerie électronique englobe deux types de protocoles :

1. le protocole **SMTP** (*Simple Mail Transfer Protocol*) pour le transfert du courrier.
2. les protocoles **POP3** (*Post Office Protocol*) et **IMAP4** (*Internet Mail Access Protocol*) pour la manipulation des boîtes aux lettres.

Le premier service nécessite une configuration bien précise et est celui que nous le allons étudier, tandis que les deux derniers ne nécessitent pas de configuration précise mais juste après l'installation des petits réglages dans le fichier « /etc/inetd.conf » pour leur lancement par le démon *inetd*.

SMTP est le protocole de transfert du courrier électronique, la ressource la plus ancienne et la plus utilisée sur Internet (l'invention du courrier électronique remonte à 1972), d'un serveur expéditeur à un ou plusieurs serveurs destinataires pourvu que leurs adresses soient connues, les adresses du courrier électronique utilisées par le protocole **SMTP** sont de la forme suivante :

nom_destinataire@nom_domaine_machine_destination.

Le protocole **SMTP** lors de son invention ne peut transférer que des messages sous formes de texte utilisant seulement des caractères ASCII. Avec la croissance des besoins d'échanger des documents de plusieurs formats (images, son, ...) on a alors développé des programmes permettant d'encoder des fichiers quelconques et de les inclure dans des fichiers texte : le programme **UUENCODE** s'imposa rapidement comme un standard. Mais a fait place à un autre protocole plus puissant : **MIME** (*Multipurpose Internet Mail Extension*, c'est le plus courant) [www_9].

Les spécifications de **SMTP** sont disponibles dans la RFC 2821.

3.6.2- Le fonctionnement de SMTP

Le protocole **SMTP** est un protocole orienté connexion, utilise le port **TCP** numéro 25 et fonctionne selon un mode client/serveur, le client désirant envoyer un message le passe à son serveur **SMTP**, celui-ci le place dans une file d'attente, puis tente de le livrer à la machine de destination. Si cette machine n'est pas accessible, le serveur **SMTP** tentera selon sa configuration de le transmettre ultérieurement (le message peut passer sur plusieurs serveurs **SMTP** pour arriver à sa destination). La **figure-3.3** illustre bien le fonctionnement de **SMTP** :

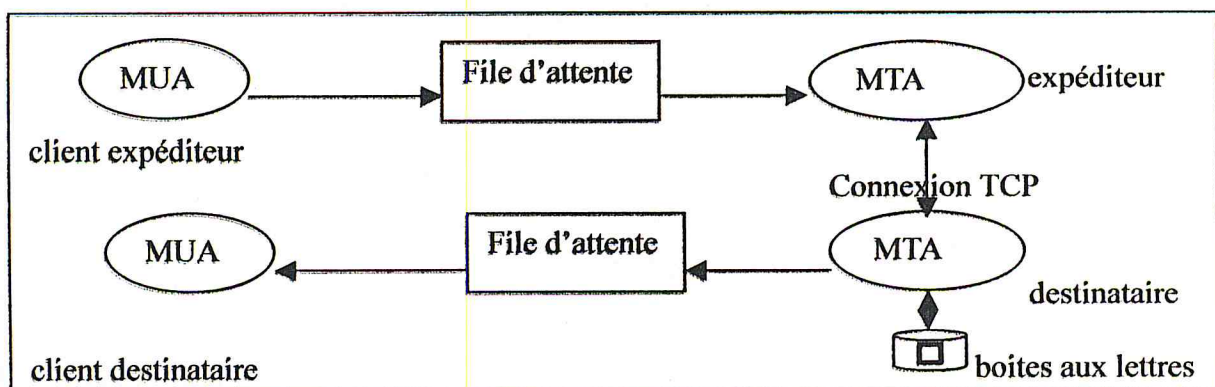


Figure-3.3 : Schéma d'une messagerie SMTP.

Le client expéditeur grâce à son logiciel de courrier électronique MUA (Mail User Agent) qui est implémenté sur le navigateur web ou sur n'importe quelle autre application (Elm, Pine ou mail) peut envoyer un ou plusieurs messages à un ou plusieurs destinataire, ces messages sont groupés dans une file d'attente généralement gérée par le MTA (Mail Transfer Agent) expéditeur qui implémente le serveur SMTP expéditeur, celui ci assure leurs acheminement au serveur destination en établissant une connexion TCP, ainsi, le client destinataire peut consulter sa boîte aux lettres qui contient les messages reçus sur son MTA destinataire (serveur SMTP destinataire).

Le serveur SMTP, contrairement au serveur FTP, est généralement présent sur le système en tant que démon (daemon), et n'est donc pas lancé à la demande et joue les deux rôles, celui d'un émetteur et celui d'un récepteur.

Le protocole SMTP fonctionne grâce à des commandes textuelles lancées par le client SMTP (MUA) au serveur SMTP, ce dernier renvoie pour chacune de ces commandes une réponse composée d'un numéro et d'un message descriptif [www_10]. Le tableau suivant montre quelques commandes SMTP :

Commande	Exemple	Description
HELO (EHLO)	EHLO 194.66.55.127	Identification à l'aide de l'adresse IP ou du nom de domaine de l'ordinateur expéditeur
MAIL FROM	MAIL FROM expéditeur@domaine.com	Identification de l'adresse de l'expéditeur
RCPT TO	RCPT TO destinataire@domaine.com	Identification de l'adresse du destinataire
DATA	DATA message	Corps du mail
QUIT	QUIT	Sortie du serveur SMTP

Tableau 7 : Description de quelques commandes SMTP

3.6.3- Configuration du serveur SMTP

Il existe actuellement trois serveurs **SMTP** connus sous Linux :

- *Sendmail* : le plus connu et le plus utilisé mais celui qui a eu de nombreuses failles de sécurité connues.
- *Postfix* : protège la plus part des problèmes de sécurité, il présente l'avantage d'utiliser un fichier de configuration humainement lisible (contrairement avec Sendmail) ce qui le rend moins susceptible d'être mal configuré, il est utilisé par la distribution Linux Mandrake.
- *QMail* : développé spécifiquement pour éviter les problèmes de sécurité, son principal problème est sa licence d'utilisation qui n'est pas réellement OpenSource et donc n'est pas toujours présent dans toutes les distributions Linux.

Nous avons choisi Postfix car il est installé par défaut dans la distribution Mandrake qui constitue notre distribution de travail, il est protégé contre certaines attaques, il est trois fois rapide que *QMail* et moins susceptible d'être mal configuré comme son équivalent Sendmail. Mandrake Linux implémente le serveur mail *Postfix* sous le démon *postfix* qui est lancé au démarrage du système grâce au fichier *rc* (la version du serveur Postfix que nous travaillons sur elle est la version 1.1.11). Postfix est basé sur des programmes (processus) semi-résidents, coopérants et assurant chacun d'eux une tâche bien définie pour rendre tout service offert par un programme disponible pour tous les autres programmes. Un démon maître (*postfix*) se charge de lancer ces programmes en fonction des besoins, de les dupliquer un nombre configurable de fois, de les réutiliser un nombre configurables de fois et de les arrêter au bout d'un temps configurable.

La configuration de Postfix s'effectue en modifiant plusieurs fichiers de configurations qui se trouvent dans le répertoire

« */etc/postfix* ». Nous citons les fichiers les plus importants :

- Le fichier *aliases* : qui contient la liste des aliases aux comptes utilisateurs réels du système, les aliases sont utilisés localement pour délivrer le courrier. Ce fichier est utilisé pour générer le fichier de la base de données des aliases « *aliases.db* » qui est utilisé par le serveur.
- Le fichier *canonical* : qui contient les noms canoniques et leurs correspondants des utilisateurs réels. Les noms canoniques sont utilisés pour remplacer les adresses locales (réelles) des utilisateurs système (utilisateur-n@localhost.dom) par des adresses lisibles et valides pour être repérées depuis Internet (par exemple de la forme nom.prénom@fournisseur.com). Ce fichier est utilisé pour générer le fichier de la base de données des noms canoniques *canonical.db* qui est utilisé par le serveur.
- Le fichier *transport* : qui contient les chemins à prendre pour délivrer le courrier à certaines utilisateurs, noms de domaines ou domaines spécifiques pour éviter à relayer tout le courrier à un serveur unique spécifié dans le fichier *main.cf*. . Ce fichier est utilisé pour générer le fichier de la base de données de transport de courrier *transport.db* qui est utilisé par le serveur.
- Le fichier *virtual* : qui contient les adresses des domaines virtuels et leurs correspondantes des utilisateurs réels du système afin de permettre de délivrer le courrier aux vrais utilisateurs dont les adresses utilisées sont de la forme : adresse@domainevirtuel. Et contient aussi une adresse spécifique appelée « catch-call » qui reçoit les messages destinés à des adresses non spécifiées dans ce fichier. Ce fichier est utilisé pour générer le fichier de la base de

données des domaines virtuels *virtual.db* qui est utilisé par le serveur.

- Le fichier *relocated* : qui contient les adresses électroniques des utilisateurs déplacés (ses adresses ne sont plus valables) et leurs adresses actuelles à les contacter (adresses personnelles, numéro de téléphone, adresses électronique, ...). . Ce fichier est utilisé pour générer le fichier de la base des données des utilisateurs déplacés *relocated.db* qui est utilisé par le serveur.
- Le fichier *main.cf* : qui est le fichier principal de configuration de postfix et qui est constitué dizaines de lignes de type :
attribut = valeur où valeur peut référencer une valeur numérique, une ou plusieurs valeurs alphabétiques ou alphanumériques séparées par des espaces ou des virgules plus des espaces, des fichiers ou des programmes. Ces attributs constituent les directives de configuration de Postfix.

Le tableau suivant montre quelques directives de configuration

Postfix avec leurs descriptions [www_10]:

Directive	Description
queue_directory	spécifie le répertoire de la file d'attente
daemon_directory	spécifie le répertoire contenant les démons de Postfix
myhostname	spécifie le nom de la machine
myorigin	spécifie la machine qui a posté le courrier
alias_map	spécifie le fichier des aliases des destinataires
alias_database	spécifie le fichier de la base de données des alaises qui est généré à partir du fichier d'aliases précédent
Mydestination	spécifie le ou les domaines dont ce serveur reçoit son courrier.
mynetworks	spécifie les adresses des réseaux de confiance suivies de leurs masques de réseaux ou de sous réseaux dont ce serveur peut relayer ses courriers vers leur destination.

Tableau 8 : Description de quelques directives de configuration SMTP.

Un exemple extrait de ce fichier est le suivant :

```
# INFORMATIONS SUR LES REPERTOIRES LOCAUX
queue_directory = /var/spool/postfix
command_directory = /usr/local/sbin
daemon_directory = /usr/local/libexec/postfix
# POSSESSION DES FILES D'ATTENTE ET DES PROCESSUS
mail_owner = postfix
# NOMS DE LA MACHINE ET DU DOMAINE
myhostname = blida.alg
# POUR L'ENVOI DU COURRIER
myorigin = $myhostname
# GESTION DES ALIAS
alias_maps = hash:/etc/postfix/aliases
alias_database = hash:/etc/postfix/aliases
```

La génération des fichiers de base de données des alias, des noms canoniques, etc. se fait en se plaçant dans le répertoire « /etc/postfix » et en lançant la commande « *postmap* nom_fichier ».

Le redémarrage du serveur **Postfix** se fait en lançant la commande « *postfix reload* ».

Le test de la bonne configuration de Postfix se fait en lançant la commande « *postfix -t* ».

3.7- Le service de news

3.7.1- Introduction

Le service de news est assuré généralement par le protocole NNTP (*NetNews Transfer Protocol*) qui est le protocole de transfert de *News* (nouvelles) sur Internet et sur n'importe quel réseau TCP/IP.

Netnews ou les *news Usenet* est aujourd'hui l'un des services les plus important sur le Web, il maintient de nombreux groupes de discussion (un groupe de discussion appelé aussi *newsgroup* est un forum d'articles ayant un sujet commun) sur de beaucoup de sujets dans des milliards de pages Web.

Le *Usenet* peut se définir comme un réseau ou une collection de sites éloignés qui s'échangent des news sur différents sujets. L'unité de base de news est l'article qui est un message écrit et posté sur Internet par un utilisateur. On lui ajoute un en-tête spécifique ayant un format similaire à celui du courrier électronique décrit dans la RFC-822.

L'article est soumis à un ou plusieurs groupes de discussion qui sont organisés dans une hiérarchie, le nom d'un de chaque groupe indique sa place dans la hiérarchie et le sujet du groupe, par exemple « **comp.os.linux-announce** » correspond à un groupe de discussion dont l'intérêt est de communiquer à propos de l'informatique (**computers**), plus particulièrement des systèmes d'exploitation (**os**, operating system), précisément les annonces de Linux (**linux-announce**). Il existe plusieurs hiérarchies principales comme comp (informatique), bionet (biologie), alt (libre) et aussi des hiérarchies propres à de nombreux pays fr (France), de (Allemagne), ...

Le protocole **NNTP** est développé en 1986, ses spécifications se trouvent dans les RFC 977 et 2980.

3.7.2- Le fonctionnement de NNTP

NNTP fonctionne selon un modèle client/serveur et utilise le port **TCP** numéro 119. L'utilisateur lit, crée et poste des messages (articles) à l'aide d'une application représentant le client **NNTP**, appelé le lecteur de news (newsreader), qui envoie une commande au serveur **NNTP**, celui-ci l'exécute et retourne le résultat avec le code de réponse correspondant.

Les messages **NNTP** ont un format similaire à celui des messages **SMTP**, ils contiennent les champs obligatoires : From (adresse de l'envoyeur), Date, Newsgroups (les noms de groupes où le message doit être distribué), Subject (sujet), MessageID (identificateur), Path(chemin

utilisé par le message) et des champs optionnels (Replay to, Sender, Followupto, Expires, references, Control, Distribution, ...) [www_11].

Le tableau suivant montre quelques commandes NNTP :

Commande	Description
article	Elle permet de retrouver le message courant, si elle est suivie par un identificateur ou un numéro elle retrouve le message correspondant.
group nom_groupe	Elle permet de sélectionner le groupe de discussion nom_groupe comme groupe courant.
post	Elle permet de poster un message
Quit	Elle indique la fin de session NNTP

Tableau 9 : Description de quelques commandes NNTP.

Les messages NNTP envoyées depuis ou vers le serveur NNTP se terminent toujours par un point (.) sur une ligne séparée. Ils sont toujours accompagnés des codes de réponse indiquant la réussite ou l'échec de la commande.

NNTP possède deux méthodes de transfert de news : l'une appelée « *pushing* » (pousser) et l'autre « *pulling* » (tirer). La première revient au protocole « *I have/ Send me* » où le client soumet un article au serveur grâce à la commande « *Ihave message-id* », le serveur renvoi un code de réponse indiquant s'il l'a déjà ou s'il le veut. S'il veut l'article, le client lui l'envoi. Cette méthode a une lourde charge causée par la recherche de chaque article dans la base de données. La deuxième méthode est réaliser à l'aide de la commande « *Newnews* » qui force le client de fournir un nom du newsgroup et une certaine date pour avoir la liste de tous les articles arrivés après cette date.

3.7.3- La configuration du serveur NNTP

Mandrake Linux utilise le serveur NNTP INN (*InterNet News*) qui est le serveur le plus utilisé comme serveur de news sur de nombreux

sites Internet et l'implémente sous le démon *innd* (la version su serveur INN que nous travaillons sur elle est la version 2.3.3). Ce démon a pour tâche le traitement de tous les articles entrants, la sauvegarde de ses articles dans le *spool* de *news*, leur relais à toutes les liaisons sortantes correspondantes ainsi que la gestion de leur historique et leur expiration. Le démon *innd* est lancé toujours au démarrage du système et continue son exécution en arrière plan et on ne peut avoir qu'un seul démon *innd* à tout moment pour ne pas avoir des besoins de synchronisation des états interne des autres copies *innd* utilisant le *spool* de *news*.

La configuration du serveur INN s'effectue en modifiant plusieurs fichiers de configuration qui se trouvent dans des répertoires différents, les plus importants de ces fichiers sont les suivants [www_12] :

- Le fichier « */var/lib/news/active* » : détermine les groupes de discussion supportés par le serveur, il contient aussi pour chaque groupe de discussion le plus grand numéro de ses articles, le plus petit numéro de ses articles et les options du groupe comme l'autorisation des lecteurs de news à poster les articles directement vers ce serveur « y », est ce que le groupe est modéré « m », etc.
- Le fichier « */var/lib/news/newsgroup* » : il contient les descriptions des groupes de discussion disponibles, ainsi, chaque ligne de ce fichier contient le nom du groupe et une description sur ce groupe.
- Le fichier « */etc/news/newsfeed* » : détermine pour ces groupes de discussion les sites (serveurs de news) qui vont recevoir ses articles, il détermine aussi les options des articles destinés à un site donné comme la taille maximale d'un article, passage de test sur le champs d'entête (Distribution), limites du nombre de sites contenus dans le champs d'entête Path, etc.
- Le fichier « */etc/news/nntpsend.ctl* » : associe les sites utilisés dans le fichier précédent qui ne sont pas des vrais noms de domaines

complets à des noms de domaines complets sur lesquels résident les serveurs de news avec la taille maximale à fournir pour chaque transfert vers ces serveurs et quelques arguments passés à la commande « *innxmit* » qui relance le processus « *nntpsend* ».

- Le fichier « */etc/news/incoming.conf* » : détermine les hôtes qui ont l'autorisation de fournir en news ce serveur via le protocole NNTP et les paramètres de contrôle (nombre maximal de connexions autorisées, mots de passe, ...). La syntaxe de ce fichier est un peu différente des autres fichiers, elle est constituée de trois types d'entrées les paires « clé : valeur », les pairs « peers » et les groupes.
- Le fichier « *readers.conf* » : détermine les clients qui ont accès à ce serveur et est utilisé précisément par le programme « *nnrpd* ». Ce fichier est constitué d'un ensemble d'entrées composées de deux types de directives de configuration : « *auth* » et « *access* », la première assure l'authentification du client par son domaine de noms ou son adresse IP et la deuxième indique les permissions d'accès (lecture ou postage).
- Le fichier « */etc/news/moderators* » : détermine les adresses électroniques des modérateurs des groupes de discussion modérés (un groupe de discussion modéré ne permet le postage des articles que par le modérateur du groupe), ainsi, il faut lui envoyer l'article à poster par courrier électronique puis, attendre s'il va accepter de le poster ou pas.
- Le fichier « *inn.conf* » : c'est le fichier principal de configuration de INN, il est composé de ligne de type attribut : valeur, ce fichier contient les paramètres globaux du serveur INN qui sont spécifiés à l'aide des attributs. Un exemple extrait de ce fichier est le suivant :

```
mta:                /usr/sbin/sendmail -oi -oem %s
organization:       A poorly-installed InterNetNews site
#pathhost:          localhost
pathnews:           /usr
# General Settings
domain:             domaine
server:             news.serveur.domaine
maxartsize:         1000000
maxconnections:     50
port:               119
addnntppostingdate: true
addnntppostinghost: true
localmaxartsize:    1000000
moderatormailer:    %s@moderators.isc.org
...
```

Le redémarrage de INN se fait en lançant la commande
« *ctlinnd reloadfile ** » et son test de la bonne configuration se fait en
lançant la commande « *ctlinnd checkfile* ».

3.8- Le service de partage de ressources Samba

3.8.1- Introduction

Les réseaux Microsoft Windows (actuellement très répandus dans les réseaux locaux) offrent une manière totalement transparente de partager des fichiers, en permettant notamment la copie par un simple glisser-déposer. Cependant, ce type de réseau ne permet à la base qu'un partage de fichiers entre machines fonctionnant avec un système Microsoft Windows ou OS/2. Ainsi, pour une machine fonctionnant avec Linux, il existe une solution: c'est d'utiliser le protocole Samba. En effet, **Samba** permet de partager les différentes ressources de la machine Linux comme les fichiers, répertoires, imprimantes, partitions(disques durs),

lecteurs (disquettes, CD-ROM, DVD), etc. avec les machines sous Windows, et inversement, et d'une manière totalement transparente.

3.8.2- Le fonctionnement de SAMBA

Le service **Samba** utilise les commande du protocole **SMB** (Server Message Bloc, a été conçu conjointement par Intel, IBM et Microsoft, il permet la connexion Linux / Windows) qui s'appuie sur le protocole **NetBIOS** (Network Basic Input Output System, a été conçu par IBM et utilisé pour « Microsoft Windows for Worksgroup ») qui fonctionne à son tour en mode client/serveur.

Le schéma suivant montre cette hiérarchie [www_13]:

Applications	
SMB	
NetBIOS	
TCP/IP	NetBUL
Pilote Réseau	

Le protocole **Samba** fonctionne selon un mode client/serveur, le serveur offre ses différentes ressources aux clients Windows qui s'y connecteront sous un compte créé par root, après une authentification par mot de passe.

Le serveur Samba est constitué de deux démons :

- **smbd** : fournit les services de connexion des clients accédant aux ressources et leur authentification.
- **nmbd** : fournit les services de résolution des noms NetBIOS en affichant la liste des différentes ressources disponibles.

Le client **Samba** est implémenté sous Windows par des routines systèmes incorporées dans la pile **TCP/IP** et **NetBIOS** et sous Linux par un programme appelé *smbclient* qui fournit une interface permettant

d'accéder aux différentes ressources partagées des stations Windows qui agissent comme serveurs **SMB**, il est lancé par la commande suivante : *smbclient \\nom_serveur_smb\\ressource -U nom_utilisateur*. Les utilitaires *smbmount* et *smbumount* sont utilisés aussi pour le montage et le démontage des ressources comme des systèmes de fichiers Linux par le protocole **Samba**, généralement on utilise le service **Samba** pour partager des ressources d'un serveur Linux entre des clients Windows. Chaque demande de connexion par **Samba**, d'une station Windows au serveur Linux, laisse une trace stockée dans un fichier *log.%m*, situé dans le répertoire */var/log/samba* (%m désigne le nom de la station). Toute connexion pourra donc être identifiée de manière précise puis examinée sur une ligne de ce fichier :

nom d'utilisateur, nom de la machine, date, heure de début, ...

Samba peut implémenter la sécurité au niveau de l'utilisateur, (ce qui est recommandé) et non au niveau des ressources comme c'est le cas dans les réseaux de type WorkGroup.

3.8.3- Configuration du serveur Samba

La configuration de **Samba** doit être effectuée sur la machine Linux (le serveur) et aussi sur les stations Windows (les clients).

Sous Windows elle est faite tout simplement par l'activation de NetBIOS et l'affectation d'une adresse IP (la pile TCP/IP supposée installée), d'un nom de station et d'un nom de groupe (qui doit être présent dans le fichier « *smb.conf* » dont nous allons l'examiner par la suite) pour chaque poste.

Sous Linux (ce qui nous intéresse) elle est faite par l'intermédiaire du fichier « *smb.conf* » qui se situe généralement dans le répertoire « */etc/samba* ».

Ce fichier décrit les ressources partageables, ainsi que leurs permissions et restrictions associées. Il se découpe en des sections, référencées par des lignes contenant ses noms entre crochets, comprenant chacune d'elles un ensemble de lignes de paramètres de type attribut = valeur.

Il existe trois sections principales [www_14] :

- La section [global] : définit les paramètres généraux sur le serveur
- La section [home] : définit le partage du répertoire personnel
- La section [printers] : définit le partage des imprimantes

On peut trouver d'autres sections comme par exemple :

- La section [tmp] : permet d'exporter le répertoire *tmp* pour effectuer les échanges de fichiers
- La section [profiles] : définit les profils utilisateurs
- La section [name_ressource] : définit name_ressource (qui peut être un répertoire, un lecteur disquette, un lecteur CD-ROM, ...) comme une ressource partagée

Le tableau ci-dessous montre quelques paramètres généraux de la section [global] :

Paramètre	Description
netbios name	Spécifie le nom netbios du serveur samba. Le nom netbios est visible dans le voisinage réseau des ordinateurs sous windows.
invalid users	Liste les utilisateurs interdits d'accès à Samba.
interfaces	Si le serveur linux dispose de plusieurs cartes réseaux et on veut restreindre son activité à un seul réseau.
security	Choix du mode de sécurité que l'on veut utiliser. Si on utilise security=user ça oblige que chaque utilisateur ait un compte sur le serveur GNU/Linux.

Tableau 10 : Description de quelques paramètres de configuration de Samba.

Le tableau suivant montre quelques variables **Samba** utilisées par le fichier « *smb.conf* » :

Variable	Définition
%a	Architecture du client Exemple: Win95, Win98, WinNT, ...
%I	Adresse IP du client
%m	Nom NetBIOS du client
%g	Groupe primaire de l'utilisateur %u
%H	Répertoire home de l'utilisateur %u
%u	Nom de l'utilisateur Unix courant
%S	Nom du partage actuel

Tableau 11 : Description de quelques variables **Samba**.

Un exemple extrait du fichier *smb.conf* est le suivant:

```
[global]
workgroup = MYGROUP
server string = Samba Server
printcap name = /etc/printcap
load printers = yes
log file = /var/log/samba/log.%m
security = user
[printers]
comment = All Printers
path = /var/spool/samba
browseable = no
guest ok = no
writeable = no
printable = yes
```

Le redémarrage du serveur Samba se fait en lançant la commande « *samba restart* » et son test de la bonne configuration se fait en lançant l'utilitaire **Samba** « *smbtestparm* ».

3.9- Le service de routage et Fire-wall

3.9.1- Introduction

Tout ordinateur connecté à n'importe quel réseau est susceptible d'être victime d'une intrusion pouvant compromettre l'intégrité du système ou bien y altérer les données.

Les pirates informatiques ayant l'intention de s'introduire dans les systèmes informatiques, en premier temps recherchent des failles dans les protocoles réseaux, les systèmes d'exploitations, les applications, voire même le personnel d'une organisation ! . Ils scrutent donc le réseau, en envoyant des paquets de données de manière aléatoire, à la recherche d'une machine connectée (cible), puis ils cherchent une faille de sécurité afin de l'exploiter et d'accéder aux données s'y trouvant. Cette menace est d'autant plus grande que la machine est connectée en permanence à Internet. Ainsi, il est nécessaire de se protéger de ces intrusions en adoptant une méthode de sécurité assurant l'intégrité et la sûreté de l'information dans le réseau. La sécurité peut être appliquée à plusieurs niveaux et assurée par plusieurs méthodes, parmi ces méthodes, il y a une connue sous le nom de Fire-wall [www_1].

Le fire-wall peut se définir comme un système physique ou logique (matériel ou logiciel) renforçant la sécurité entre un réseau interne et un réseau externe ou bien servant d'interface entre un réseau interne et les réseaux qui lui sont connectés afin de contrôler et éventuellement bloquer la circulation des paquets de données, en analysant les informations contenues dans les couches 2,3 et 4 du modèle DoD.

Le fire-wall constitue un point de centralisation, tout le monde passe par lui pour sortir du réseau interne et consulter l'Internet, c'est le seul élément donnant cet accès et c'est lui qui détermine :

- A quels services internes peut-on accéder de l'extérieur ?
- Quels éléments externes peuvent accéder aux services internes autorisés ?
- A quels services externes peut-on accéder au moyen des éléments internes ?

Lorsque certaines machines du réseau interne ont besoin d'être accessibles de l'extérieur (comme c'est le cas par exemple pour un serveur web, un serveur de messagerie,...) il est souvent nécessaire de créer une nouvelle interface vers un réseau à part, accessible aussi bien du réseau interne que de l'extérieur, sans pour autant risquer de compromettre la sécurité de l'entreprise. On parle ici de la zone démilitarisée (DMZ, *DeMilitarized Zone*) [www_1].

La figure suivante montre une représentation d'un Fire-wall.

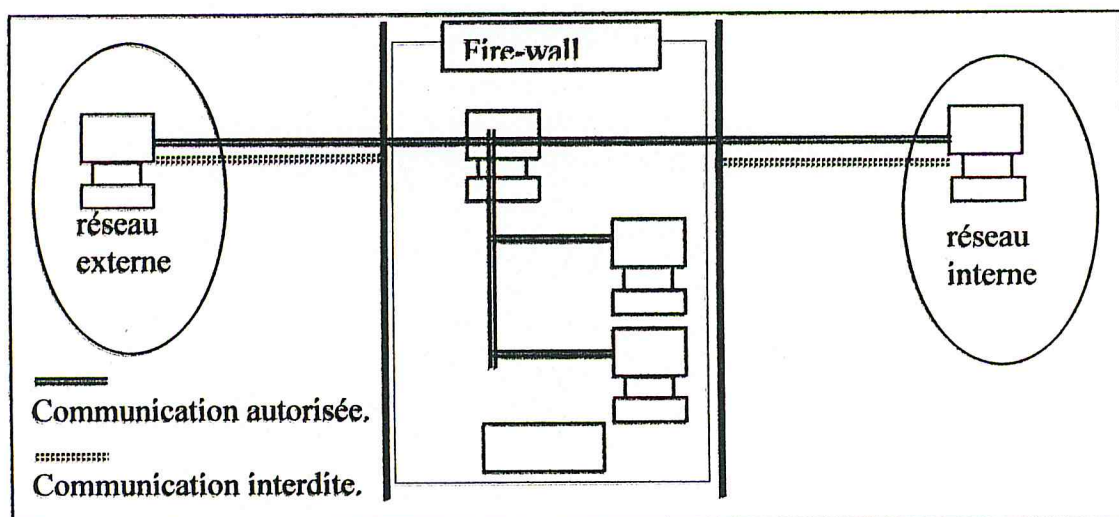


Figure-3.4 : Représentation d'un système Fire-wall [www_1].

Dans le cas où la zone protégée se limiterait à l'ordinateur sur lequel le fire-wall est installé on parle de *fire-wall* personnel [www_15].

De plus, un système Fire-wall peut être utilisé comme un routeur complet permettant d'assurer l'acheminement des paquets, la translation des adresses IP, la redirection des paquets, ...

3.9.2- Le fonctionnement d'un Fire-wall

Un fire-wall fonctionne selon ses règles prédéfinies qui lui permettent :

- Soit d'autoriser uniquement les communications ayant été explicitement autorisées (tout ce qui n'est pas explicitement autorisé est interdit).
- Soit d'interdire les communications ayant été explicitement interdits (tout ce qui n'est pas explicitement interdit est autorisé).

Les deux approches dépendent de la politique de sécurité adoptée, la première est la plus sûre, mais elle impose toutefois une définition précise et contraignante des besoins en terme de communication.

Il y a deux types de Fire-wall [how_99] :

- Le Fire-wall IP ou filtrant : qui est basé sur la technique de filtrage des paquets IP (*datagrammes*) échangés entre les deux réseaux. Le filtrage peut être appliqué soit sur les adresses IP d'origine ou de destination (*adress filtering*), soit sur le type de protocole ou le numéro de port (*protocol filtering*). C'est le type le plus utilisé dans les réseaux TCP/IP et qui sera aussi utilisé pour être configuré par notre outil.

- Le Fire-wall comme serveur mandataire : qui est chargé de réaliser toutes les connexions entre les deux réseaux. Tout utilisateur désirant établir une connexion avec l'autre réseau doit être authentifié par le serveur mandataire qui lui établit cette connexion et lui renvoi le résultat.

3.9.3- Le routage derrière un Fire-wall

Pour une machine Linux dotée de deux interfaces réseaux (une pour le réseau local et l'autre pour Internet) ou bien d'une seule interface réseau pour le réseau interne et une connexion Internet passant par un modem, il existe deux configurations possibles selon que l'ISP (Internet Service Provider) gère le trafic vers les machines du réseau local [www_15]:

- *Routed* : appelée aussi passerelle simple, elle consiste à diviser le réseau (ou le sous réseau) local attribué par l'ISP en deux sous réseaux locaux le premier regroupe les machines de la zone DMZ et le deuxième pour le reste des machines, toutes les machines devront être reconfigurées pour utiliser la machine Fire-wall et non plus le retour comme destination par défaut, toutes les machines du réseau locales sont atteintes depuis Internet par le biais du Fire-wall.
- *Non Routed* : elle est moins coûteuse que la précédente, elle ne nécessite pas l'obtention d'un réseau complet ou un sous réseau auprès de l'ISP mais quelques adresses IP ou bien une seule adresse IP. elle consiste à reconfigurer les machines du réseau local qui possèdent des adresses IP de la RFC 1918 ou des adresses IP publiques(attribuées par un ISP) pour utiliser la

machine Fire-wall et non plus le routeur comme destination par défaut dont son adresse IP doit être publique. Cette configuration est réalisée en utilisant plusieurs méthodes :

1. SNAT (Source Network Address Translation) : elle est basée sur la translation des adresses des machines locales en l'adresse de la machine Fire-wall pour toute connexion initiée depuis le réseau local à Internet.
2. DNAT (Destination Network Address Translation): elle est basée sur la translation de l'adresse du Fire-wall en une adresse d'une machine locale. Cette méthode permet donc une connexion initiée depuis Internet à une machine locale ce qui impossible avec SNAT tant que cette machine n'a pas une adresse IP publique.
3. Proxy-ARP : elle est basée sur le principe de rendre le trafic parallèle au Fire-wall (directement de l'ISP vers les machines ayant des adresses IP publiques) vers un trafic derrière un Fire-wall (le trafic passe par le Fire-wall pour arriver à ces machines). Ces machines ont le même masque du réseau que celui de la machine fire-wall et utilisent cette dernière comme destination par défaut.
4. Static NAT (Network Address Translation) : elle est basée sur le principe de réaliser un mapping (une translation) entre les adresses IP de la RFC 1918 et les adresses IP publiques attribuées (elle applique le SNAT en cas d'une connexion depuis le réseau local vers Internet et le DNAT en cas d'une connexion depuis Internet vers le réseau local).

3.9.4- La configuration du serveur routeur/fire-wall

Une machine Linux peut être utilisée comme un routeur et Fire-wall filtrant grâce à des serveurs Linux spécifiques réalisant le routage et le filtrage de paquets transmis entre les deux réseaux.

Parmi ces serveurs, nous avons choisi le serveur routeur/fire-wall « *Shorline Fire-wall* » connu aussi par *Shorewall* (disponible sur notre machine Mandrake Linux et sa version que nous travaillons sur elle est la version 1.3.7). *Shorewall* est basé sur les utilitaires de routage et filtrage classique « *iptables* » et « *iproute* ». Shorewall est un logiciel libre et peut être modifié et redistribué sous les contraintes de la licence GPL.

Le serveur *Shorewall* démarre généralement avec le système et utilise plusieurs fichiers de configuration qui se trouvent dans le répertoire « */etc/shorewall* ». Nous décrivons les fichiers les plus importants dans la configuration de shorewall dans ce qui suit [www_15]:

- Le fichier « */etc/shorewall/interfaces* » : contient les détails des interfaces réseaux disponibles sur la machine Fire-wall. On peut avoir des zones qui sont assignées à des hôtes ou à des sous réseaux en les introduisant dans le fichier « */etc/shorewall/hosts* ».
- Le fichier « */etc/shorewall/zones* » : contient les noms de zones, shorewall voit le réseau où il tourne comme un ensemble de zones (*fw* la zone décrivant le système où est installé shorewall, *net* la zone décrivant Internet, *loc* la zone décrivant le réseau local, *dmz* la zone décrivant la zone démilitarisée, ...).
- Le fichier « */etc/shorewall/policy* » : contient les politiques (ACCEPT : autoriser, DROP : ignorer et REJECT : interdire) par défaut pour les connexions d'une zone à une autre.

- Le fichier «*/etc/shorewall/rules*» : contient les règles qui définissent les exceptions aux politiques par défaut. Dans l'exemple suivant sont définies trois règles : la première permet d'accepter les connexions depuis Internet vers la machine Fire-wall sur le port TCP 25 (le cas d'un serveur de messagerie installé sur cette machine), la deuxième permet de rediriger les connexions depuis un réseau spécifique d'Internet sur le port 80 vers une machine du réseau interne (le cas d'un serveur web installé sur cette machine) et la troisième permet de rediriger toutes les connexions depuis Internet sur le port 119 vers la machine Fire-wall sur le port 2555 (le cas où est installé un serveur spécifique pour l'asservissement des requêtes de ces connexions).

#Action	Source	Destination	Protocol	D Port	S Port	Original Address
ACCEPT	net	fw	tcp	25		
DNAT	net:200.200.200.0/24	192.168.1.2	tcp	80	-	178.166.12.35
REDIRECT	net	2555	tcp	119	-	178.166.12.39

On peut spécifier d'autres règles comme REJECT et DROP .

- Le fichier «*/etc/shorewall/nat*» : contient les règles définissant le mode NAT statique pour le Fire-wall.
- Le fichier «*/etc/shorewall/proxyarp*» : contient les règles définissant le mode ProxyARP pour le Fire-wall.
- Le fichier «*/etc/shorewall/blacklist*» : définit la liste noire, il contient les adresses IP ou les adresses des réseaux dont les paquets ayant provenance d'eux sont ignorés ou rejetés (on peut spécifier de plus le protocole de transport et le numéro de port).
- Le fichier «*/etc/shorewall/shorewall.conf*» : est le fichier principal de configuration de Shorewall. Il permet de configurer les divers paramètres Shorewall (le répertoire de travail de shorewall, l'utilisation des listes noires, niveau de logging, ...).

Un exemple extrait de ce fichier est le suivant :

```
SHARED_DIR=/usr/share/shorewall
LOGFILE=/var/log/firewall
LOGUNCLEAN=info
BLACKLIST_LOGLEVEL=info
TCP_FLAGS_LOG_LEVEL=info
RFC1918_LOG_LEVEL=info
PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/bin:/usr/local/sbin
STATEDIR=/var/state/shorewall
FW=fw
NAT_ENABLED=Yes
MANGLE_ENABLED=Yes
IP_FORWARDING=On
ADD_IP_ALIASES=Yes
ADD_SNAT_ALIASES=Yes
ROUTE_FILTER=No
NEWNOTSYN=Yes
BLACKLIST_DISPOSITION=DROP
...
```

Le redémarrage du serveur routeur/fire-wall Shorewall se fait en lançant la commande « *shorewall restart* » et son test de la bonne configuration se fait en lançant la commande « *shorewall check* ».

3.10- Conclusion

Dans ce chapitre, nous avons vu les détails de configuration d'une machine Linux pour faire fonctionner **TCP/IP**, nous avons abordé les différents services réseaux à configurer par notre outil, leurs descriptions, leurs fonctionnements ainsi que les détails de leurs fichiers de configuration.

Dans le chapitre suivant nous allons voir les détails de la conception de notre outil et son interface utilisateur.

Chapitre 4

Conception

4.1- Introduction

Notre outil a pour rôle de configurer les services réseaux étudiés dans le chapitre précédent. Pour concevoir cet outil nous avons choisi une approche modulaire permettant de le diviser en un ensemble de modules configurant chacun d'eux un service réseau étudié précédemment. Ainsi, nous aurons sept modules à concevoir comme le montre la figure suivante :

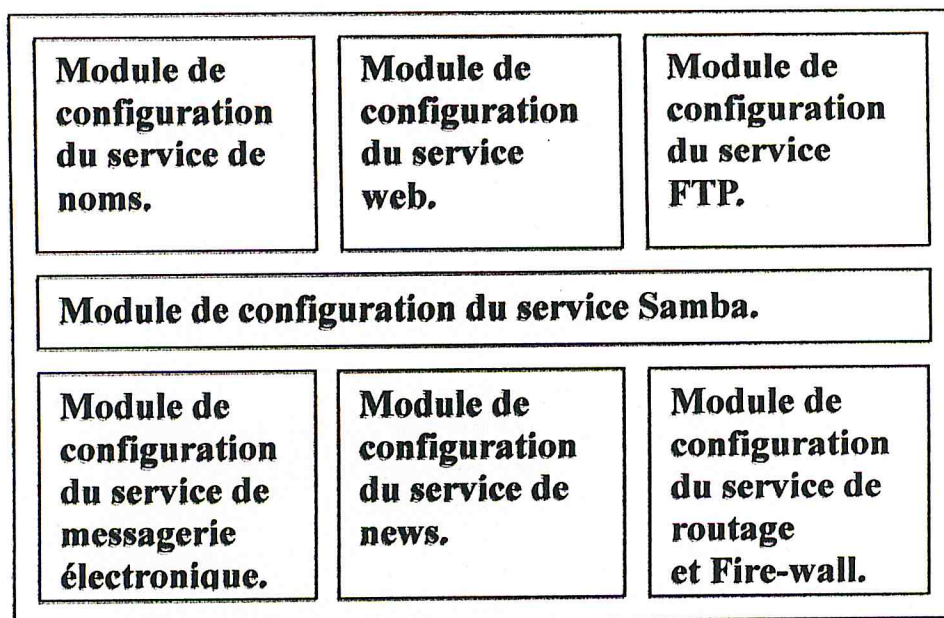


Figure-4.1 : Schéma de conception de l'outil de configuration des divers services réseaux.

Sans oublier la conception de l'interface utilisateur qui sera le point de transition entre les actions de l'administrateur et les réactions de l'outil et qui va présenter en plus des fonctionnalités de la configuration des services réseaux vus précédemment, les fonctionnalités des tests de bon fonctionnement ou de la bonne configuration de ces services et aussi de consultation des informations sur les états d'exécution des serveurs et les statistiques des cartes réseaux disponibles.

4.2- Conception des modules de configuration des services

4.2.1- Module de configuration du service de noms

Ce que doit fournir ce module est :

- La visualisation du nom du serveur de noms et des différentes zones existantes et la possibilité de les modifier, les supprimer et l'ajout d'autres zones (primaires ou secondaires) et les modifier.
- La consultation des propriétés du serveur de noms (répertoire de travail du serveur, le fichier d'identificateur de processus, ...) et la possibilité de les modifier.
- La visualisation des propriétés de la zone ainsi que la possibilité de les modifier (nom, fichier, notification des serveurs esclaves, restreindre l'autorisation d'utilisation, ...), la possibilité d'ajouter des enregistrements de ressource (enregistrement A, NS, ...), la possibilité de les supprimer et de les modifier.

Nous avons vu ce module comme un ensemble de composantes comme le montre la figure suivante :

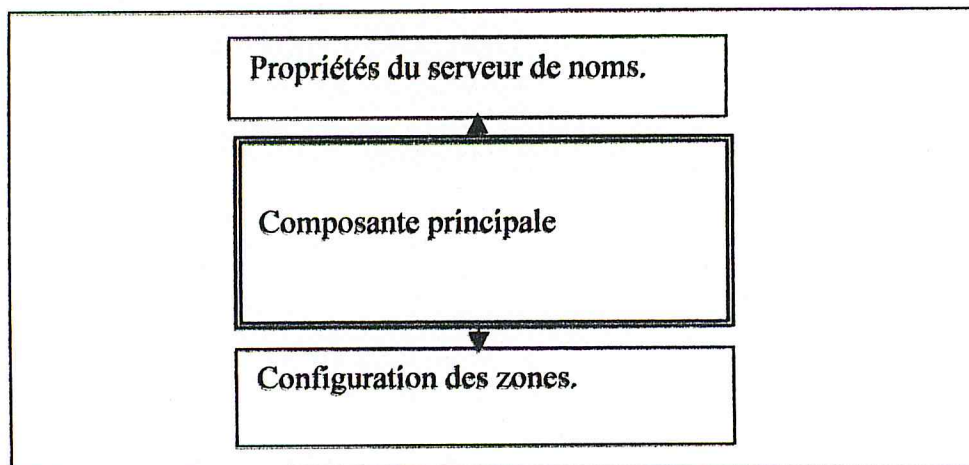


Figure-4.2 : schéma de conception du module de configuration du service de noms.

Nous donnons l'explication des rôles de ces composantes :

La composante principale :

Elle contient le nom de serveur et la liste des zones. Ainsi, au lancement du module de configuration de **DNS**, cette composante retire le nom de la machine et examine le fichier de configuration

« */etc/named.conf* » en retirant toutes les zones disponibles et en les affichant à l'administrateur qui à son tour peut appliquer des changements sur les propriétés du serveur ou sur les zones en lançant la composante appropriée. Cette composante permet aussi de redémarrer le serveur pour prendre en compte les changements affectés par les autres composantes.

La composante des propriétés du serveur de noms:

Cette composante permet d'examiner le fichier de configuration « */etc/named.conf* » et retirer toutes les propriétés du serveur (répertoire de travail, fichier d'identificateur du processus, l'utilisation des redirecteurs et la liste des redirecteurs à contacter, mettre esclave le serveur, restriction d'utilisation du serveur à certaines machines, ...) en les affichant à l'administrateur qui à son tour peut les modifier ou pas et retourner à la composante principale. Toute modification appliquée par l'administrateur se répercute sur le fichier « */etc/named.conf* » sans violation des règles syntaxiques suivies.

La composante de configuration des zones :

Cette composante permet d'ajouter des zones (primaires ou secondaire) avec toutes ses propriétés (nom, fichier, type, notification des serveurs esclaves en cas de changement, numéro de série, temps de rafraîchissement, ...). Elle regroupe les informations de la nouvelle zone passées par l'administrateur et les classe en deux catégories : une pour la mettre dans le fichier de configuration « */etc/named.conf* » et l'autre dans un nouveau fichier qui constitue le fichier de cette zone. Cette composante permet aussi après une confirmation de l'administrateur de

supprimer une zone sélectionnée en supprimant toutes ses entrées dans le fichier de configuration « */etc/named.conf* » et en supprimant aussi son fichier de zone. Elle permet aussi de visualiser les propriétés d'une zone sélectionnée en examinant le fichier de configuration « */etc/named.conf* » et le fichier de cette zone, retirer les propriétés de cette zone (type de zone, fichier de la zone, numéro de série, liste des enregistrements de ressources, ...). Elle permet aussi d'ajouter des enregistrements de ressource en choisissant son type (enregistrement d'adresse A, enregistrement du pointeur PTR, enregistrement du nom de serveur NS, enregistrement d'échangeur de messagerie MX, ...) puis en spécifiant ses paramètres qui dépendent du type choisi (par exemple pour un enregistrement A on spécifie le nom de domaine de l'hôte et son adresse IP), de supprimer un enregistrement de ressource et de modifier ses paramètres. Elle permet de plus de valider les modifications apportées à une zone donnée et de retourner à la composante principale.

4.2.2- Module de configuration du service web

Ce module doit fournir :

- La visualisation du nom de serveur web et la possibilité de consulter et de modifier les propriétés globales de ce serveur (adresse de l'administrateur, répertoire de travail, répertoire racine, fichier d'identificateur de processus, temps d'attente des connexions, paramètres de site, paramètres de logging,...).
- la visualisation des adresses **IP** et les numéros de ports des machines du réseau où le serveur peut écouter et accepter les requêtes clients et la possibilité d'ajouter, de supprimer ou de modifier ces adresses **IP** et numéros de ports
- La visualisation des hôtes virtuels (un hôte virtuel peut être défini comme une instance du serveur créée par le serveur lui-même avec un nom et des propriétés différentes et qui s'exécute sur la même machine où est installé le serveur et sur un port différent.) disponibles dans ce serveur, la possibilité d'ajouter de nouveaux hôtes virtuels, de supprimer des hôtes virtuels existants ou bien la modification des hôtes virtuels existants ainsi que leurs propriétés(nom du serveur de l'hôte, les ports à écouter, paramètres de site, paramètres de logging, paramètres SSL-Secure Socket Layer-, paramètres des répertoires, paramètres de variables d'environnement, ...).
- La possibilité de configurer les modules Apache.
- La possibilité de redémarrer le serveur de noms pour prendre en comptes les changements affectés par les différentes actions de l'administrateur.

Nous avons vu ce module comme un ensemble de composantes comme le montre la figure 4.3.

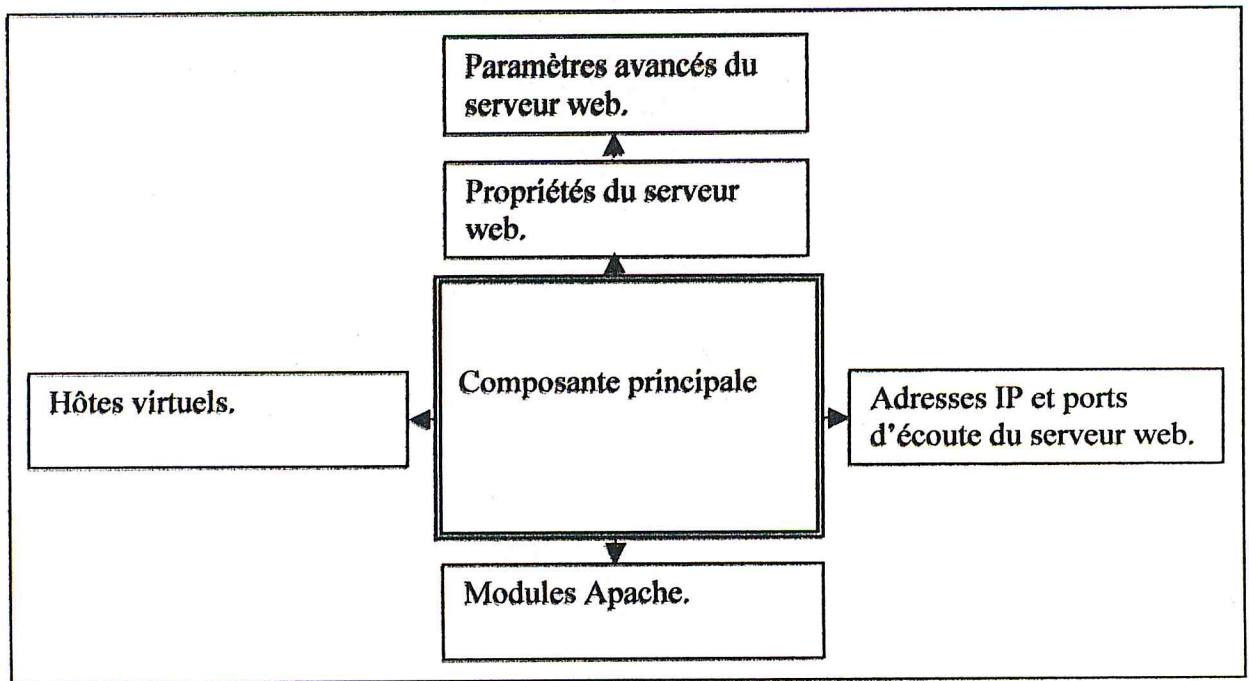


Figure-4.3 : schéma de conception du module de configuration du service web.

Nous expliquons le rôle des différentes composantes de ce schéma :

La composante principale :

Elle contient le nom du serveur, les adresses IP et les ports d'écoute du serveur ainsi que la liste des hôtes virtuels disponibles sur le serveur. Au lancement de cette composante elle examine le fichier de configuration globale « */etc/httpd/httpd.conf* » et retire toutes les informations nécessaires des différentes directives de configuration concernant le nom de serveur, les adresses IP et les ports d'écoute du serveur et les hôtes virtuels disponibles. Elle les visualise ensuite en permettant à l'administrateur de consulter les propriétés du serveur, d'ajouter, de supprimer ou de modifier les adresses IP et les ports d'écoute du serveur, d'ajouter, de supprimer ou de modifier les hôtes virtuels disponibles ainsi que leurs propriétés, ou bien de configurer les modules Apache en lançant la composante appropriée.

Elle permet aussi de redémarrer le serveur web pour prendre en compte les changements effectués par l'administrateur.

La composante des propriétés du serveur web :

Cette composante permet de consulter les propriétés globales du serveur et la possibilité de les modifier (nom du serveur web, répertoire racine, fichier d'identificateur de processus du serveur, nombre maximal des processus du serveur, nombre minimal des processus du serveur, temps maximal des connexions, autorisation de nombre illimité de requêtes par connexion ou limitation de ce nombre, autorisation des connexions persistantes, l'autorisation d'un nombre illimité de requêtes par connexion persistante ou limitation de ce nombre, ...). Elle permet aussi de consulter les paramètres avancés du serveur (paramètres du site, de logging, des variables d'environnement et des répertoires) en lançant la composante appropriée. Elle examine le fichier de configuration « */etc/httpd/httpd.conf* » et le fichier « */etc/httpd/commonhttp.conf* » et retire toutes les informations nécessaires pour la visualisation des propriétés du serveur et modifie le contenu de ces deux fichiers en cas de changement de propriétés sans violation des règles syntaxiques suivies pour ces fichiers. Elle permet aussi de revenir à la composante principale.

La composante des paramètres avancés du serveur web :

Cette composante permet de consulter les paramètres avancés du serveur (Paramètres du site : liste de recherche des pages et la liste des pages d'erreurs. Paramètres de logging : fichier de logging, fichier de logging des erreurs, niveau de logging et la recherche DNS. Paramètres des variables d'environnement : ajout, suppression et modifications des variables pour les scripts CGI, des variables passées à ces scripts et la désactivation de certaines variables. Paramètres des répertoires : ajout, suppression et modification des répertoires, leurs propriétés et leurs options.), de les modifier et de revenir à la composante précédente.

La composante des adresses et ports d'écoute du serveur web :

Cette composante permet d'ajouter des adresses IP et des numéros de ports de la machine sur lesquels le serveur écoute les requêtes clients, de supprimer des adresses IP et des numéros de ports utilisés par le serveur après la confirmation de l'administrateur et de les modifier en maintenant toutes les entrées correspondantes dans le fichier de configuration «*/etc/httpd/httpd.conf*». Elle permet aussi de valider les modifications apportées et de retourner à la composante principale.

La composante des hôtes virtuels :

Cette composante permet d'ajouter des hôtes virtuels avec ses propres noms, propriétés (adresse IP, port d'écoute, répertoire racine, adresse de l'administrateur, ...) et paramètres avancés (paramètres de site, de logging, des variables d'environnement, des répertoires et de plus de ces paramètres qui ressemblent à ceux du serveur il y a les paramètres de sécurité SSL –*Secure Socket Layer*– : fichier de certificat, fichier de la clé, fichier de logging SSL, type d'authentification, nombre de vérifications dans la chaîne et les options SSL), de supprimer des hôtes virtuel après confirmation et de modifier ses propriétés et ses paramètres avancés en maintenant le fichier «*/etc/httpd/Vhosts.conf*».

Elle permet aussi de valider les modifications apportées et de retourner à la composante principale.

La composante des modules Apache:

Elle permet de visualiser la liste des modules Apache présent sur le serveur qui sont chargés automatiquement au démarrage, d'ajouter des nouveaux modules ou de supprimer des modules existant en maintenant le fichier «*commonhttpd.conf*».

4.2.3- Module de configuration du service FTP

Ce module doit fournir :

- La visualisation du nom de serveur **FTP** et la possibilité de consulter et de modifier ses propriétés (numéro de port, le mode de transfert de données,...).
- La visualisation des Limites d'accès aux services offerts par le serveur et la possibilité d'ajout, de suppression ou de modification d'une limite.
- La visualisation des répertoires, la possibilité d'ajouter, de supprimer de modifier un répertoire (en modifiant ses propriétés comme les limites d'accès sur la consultation de ses fichiers, le transfert de ses fichiers, la mise à jours de ses fichiers, ...).
- La possibilité de configurer l'accès anonyme (un accès anonyme permet de définir les modalités d'accès au serveur **FTP** anonyme),
- La visualisation de la liste des utilisateurs interdits, la possibilité d'ajouter et de supprimer des utilisateurs interdits.
- La visualisation des hôtes virtuels (qui peut être défini comme une instance du serveur qui s'exécute sur la même machine mais avec des propriétés différentes.), la possibilité d'ajouter, de supprimer ou de modifier des hôtes virtuels (en modifiant ses propriétés comme le nom de l'hôte virtuel, le numéro de port, les limites d'accès, les répertoires, ...).

Nous avons vu ce module comme un ensemble de composantes comme le montre la figure 4.4.

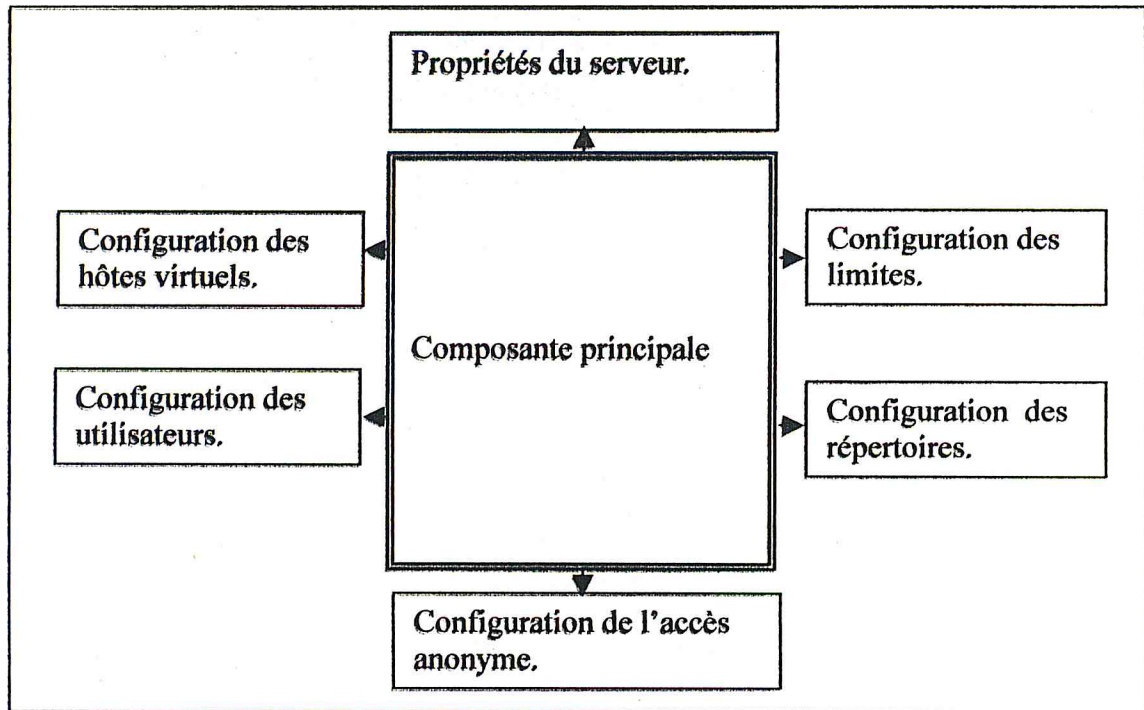


Figure-4.4 : schéma de conception du module de configuration du service FTP.

Nous expliquons les différentes composantes de ce module :

La composante principale :

C'est la composante principale, elle contient le nom du serveur, la liste des limites, la liste des répertoires disponibles sur le serveur, le nom de l'accès anonyme, la liste des utilisateurs qui n'ont pas accès au serveur, et la liste des hôtes virtuels. Toutes ces informations sont obtenues en examinant le fichier de configuration globale « */etc/proftpd.conf* ». Cette composante permet donc de retirer puis de visualiser toutes ces informations nécessaires concernant le nom de serveur, les limites, les répertoires, l'accès anonyme, les utilisateurs et les hôtes virtuels. Elle permet de modifier ces informations en lançant les composantes appropriées. Elle permet aussi de lancer la composante des propriétés du serveur FTP et de le redémarrer.

La composante des propriétés du serveur FTP :

Cette composante examine le fichier de configuration « */etc/proftpd.conf* » et retire toutes les informations nécessaires pour la visualisation des propriétés du serveur (nom et type du serveur, numéro de port, le nom de l'utilisateur, le nom du groupe, le mode de transfert de données, le nombre de clients connectés simultanément, ...). Cette composante permet aussi de revenir à la composante principale.

La composante de configuration des limites d'accès :

Cette composante permet d'ajouter des limites qui vont s'appliquer soit sur la partie générale du serveur, soit sur la partie générale des hôtes virtuels, ou soit sur les répertoires. Elle permet aussi de modifier les limites existantes ou de supprimer des limites après la confirmation de l'administrateur et de retourner à la composante principale.

La composante de configuration des répertoires:

Cette composante permet d'ajouter des répertoires, de modifier les propriétés des répertoires, de supprimer des répertoires existants après la confirmation de l'administrateur et de retourner à la composante principale. Elle permet aussi la configuration des limites d'accès pour les répertoires.

La composante de configuration de l'accès anonyme:

Cette composante permet de configurer un accès anonyme, de modifier les propriétés de cet accès anonyme (le chemin d'accès, le nombre maximum de clients atteints, le nom de l'utilisateur et du groupe) ou de supprimer l'accès anonyme configuré après la confirmation de l'administrateur. Elle permet aussi de retourner à la composante principale.

La composante des utilisateurs interdits :

Cette composante permet d'ajouter des utilisateurs qui n'ont pas accès au serveur en maintenant le fichier « */etc/ftpusers* », de supprimer

des utilisateurs déjà existés dans ce fichier après la confirmation de l'administrateur et de retourner à la composante principale.

La composante des hôtes virtuels :

Cette composante permet d'ajouter des hôtes virtuels avec ses propres propriétés (nom de l'hôte virtuel, le numéro de port, le nombre de clients connectés en même temps, le nombre de clients pour la même personne, le message d'accueil, le message affiché lors d'une connexion réussite ou perdue), de modifier les hôtes virtuels existants ou de supprimer des hôtes virtuels après la confirmation de l'administrateur et de retourner à la composante principale. Elle permet aussi la configuration des limites d'accès des hôtes virtuels.

4.2.4- Module de configuration du service de messagerie électronique

Ce que ce module doit fournir c'est :

- La visualisation du nom de serveur de messagerie et la possibilité de consulter et modifier les propriétés globales du serveur (le nom de domaine du courrier sortant, les domaines de noms dont le serveur reçoit leur courrier, les interfaces réseaux du serveur, ...) ainsi que les paramètres **SMTP** (temps des transactions **SMTP**, nombre d'erreurs pour ignorer temporairement les requêtes d'un client, nombre d'erreurs pour clore une session **SMTP**, etc.
- La possibilité de configurer le support des aliases.
- La possibilité d'activer ou de désactiver le support pour les noms canoniques ainsi que sa configuration.
- La possibilité d'activer ou de désactiver le support pour les domaines virtuels ainsi que la possibilité de sa configuration.

- La possibilité d'activer ou de désactiver le support pour le transport de courrier ainsi que la possibilité de sa configuration.
- La possibilité d'activer ou de désactiver le support pour les utilisateurs déplacés ainsi que la possibilité de sa configuration.

Nous avons vu ce module comme l'ensemble des composantes montrées dans la figure suivante :

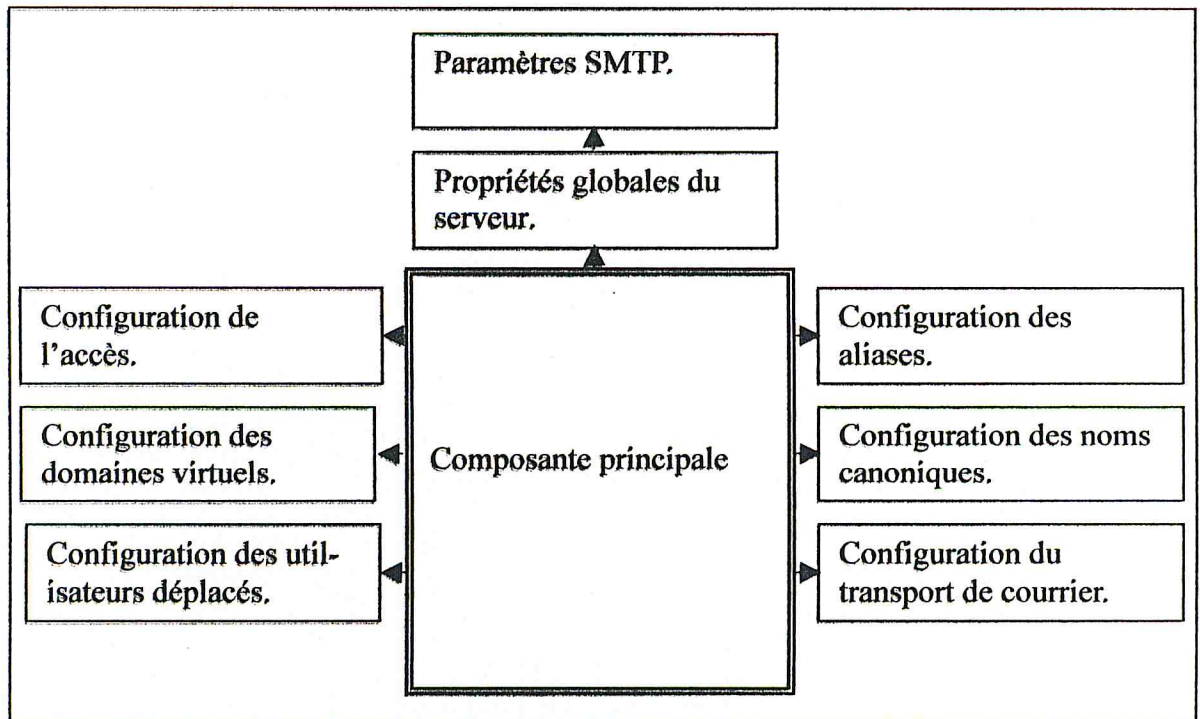


Figure-4.5 : Schéma de conception de sous module de configuration du service de messagerie électronique.

Nous donnons l'explication des rôles de ces composantes :

La composante principale :

Elle permet de visualiser le nom du serveur **SMTP** et de consulter les propriétés globales du serveur **SMTP**, de visualiser les fichiers de la table des alias et la base données des alias et la possibilité de configurer les alias, d'activer, désactiver ou configurer le support des : noms canoniques, domaines virtuels, transport de courrier et les utilisateurs déplacés en lançant les composantes appropriées. Tout en

maintenant le fichier de configuration global de postfix « */etc/postfix/main.cf* ». Et de plus elle est chargée de régénérer les fichiers des bases de données pour les différents supports (aliases, noms canoniques, ...) utilisés par le serveur et de redémarrer ce dernier pour prendre en compte les différentes modifications apportées.

La composante des propriétés du serveur SMTP :

Elle permet de consulter et modifier les propriétés générales du serveur (le domaine de noms du courrier sortant, domaines de noms dont le serveur reçoit leur courrier, la classe des erreurs à rapporter à l'administrateur, les réseaux de confiance, le masquage des adresses, le relais vers d'autres serveurs de messagerie, les paramètres de temps, les répertoires utilisées par le serveur, paramètres des processus, les limites sur les tailles des messages, des boîte aux lettres, ...) en maintenant le fichier de configuration globale « */etc/postfix/main.cf* » et les diverses directives de configuration appropriées sans violation des règles syntaxiques suivies. Elle permet aussi de consulter les paramètres SMTP en lançant la composante appropriée et d'effectuer les différentes modifications apportées et retourne à la composante principale.

La composante des paramètres SMTP :

Elle permet de consulter et modifier les paramètres SMTP (temps d'une transaction SMTP, temps avant émettre une réponse SMTP d'erreur, nombre des erreurs pour lesquelles le client sera ignorer temporairement ou bien sa session sera terminée automatiquement, les codes des réponses d'erreurs, l'activation ou la désactivation de certaines commandes, etc. Elle permet aussi d'effectuer les modifications apportées et de retourner à la composante précédente en maintenant le fichier de configuration globale « */etc/postfix/main.cf* ».

La composante des aliases :

Elle permet de visualiser la liste des aliases disponibles sur le serveur, donner la possibilité d'ajouter et de supprimer des aliases en maintenant le fichier de configuration des aliases « */etc/postfix/aliases* » et de retourner à la composante principale.

La composante de configuration de l'accès :

Elle permet de visualiser la liste des utilisateurs autorisés ou interdits d'accéder au serveur. Elle permet aussi d'ajouter ou de supprimer un utilisateur (autorisé ou interdit d'accéder) à la liste en maintenant le fichier « */etc/postfix/access* ».

La composante de configuration des noms canoniques :

Elle permet de visualiser la liste des noms canoniques (pour les émetteurs et les receveurs de courrier) utilisés par le serveur, donner la possibilité d'ajouter et de supprimer un nom canonique en maintenant le fichier de configuration des noms canoniques « */etc/postfix/canonical* ». et de retourner à la composante principale.

La composante de configuration des domaines virtuels :

Elle permet de visualiser la liste des domaines virtuels gérés par le serveur et la possibilité d'ajouter, de supprimer ou de modifier ces domaines virtuels, une modification d'un domaine virtuel entraîne un ajout ou une suppression de ses adresses. Tout en maintenant le fichier de configuration des domaines virtuels « */etc/postfix/virtual* ». Elle permet aussi d'effectuer les modifications apportées et retourner à la composante principale.

La composante de configuration de transport de courrier :

Elle permet de visualiser les différents chemins à suivre par les adresses ou les domaines spécifier pour acheminer le courrier. Elle donne la possibilité d'ajouter ou de supprimer des chemins de transport constitué d'un domaine virtuel, d'un serveur de messagerie à transférer le

courrier de ce domaine vers lui et d'un numéro de port. Elle permet d'effectuer les changements apportés en maintenant le fichier de configuration de transport de courrier « */etc/postfix/transport* » et de retourner à la composante principale.

La composante de configuration des utilisateurs déplacés :

Elle permet de visualiser la liste des utilisateurs déplacés ainsi que leurs adresses actuelles à contacter, elle permet d'ajouter un utilisateur à la liste ou de le supprimer de cette liste en maintenant le fichier des utilisateurs déplacés « */etc/postfix/relocated* ». Elle permet aussi d'effectuer les changements apportés et de retourner à la composante principale.

4.2.5- Module de configuration du service de news

Ce module doit fournir les fonctionnalités suivantes :

- La visualisation du nom du serveur de news, la possibilité de consulter ses propriétés (le domaine de noms du serveur, le nom d'hôte qui doit apparaître dans le « From » des articles postés, le numéro de port, ...) et la possibilité de les modifier.
- La visualisation des groupes de news supportés par le serveur, la possibilité d'ajouter, de supprimer ou de modifier ses propriétés (adresse électronique du créateur, le numéro du plus petit article actif, postage local des articles autorisé ou non, ...).
- La visualisation des liaisons de news qui permettent de contrôler quels sont les groupes de news à relayer aux autres serveurs, la possibilité d'ajouter, de supprimer ou de modifier les propriétés d'une liaison de news (le nom du site, les groupes de news concernés, options d'expédition comme la taille maximale d'un article expédié, type de liaison, ...).

- La visualisation des fournisseurs de ce serveur en news, les fournisseurs peuvent être des serveurs de news (pairs au sens INN) ou bien des groupes de serveurs (groupes de pairs), la possibilité d'ajouter, de supprimer ou de modifier un fournisseur (en modifiant ses propriétés comme son nom d'hôte ou adresse IP, nombre maximal de connexion, contrôle de flux, ...).
- La visualisation des lecteurs de news des hôtes autorisés à utiliser le serveur, la possibilité d'ajouter, de supprimer ou de modifier un lecteur de news (en modifiant ses propriétés les groupes de discussions autorisés en lecture ou en postage, ...).

Nous avons vu ce module comme un ensemble de composantes comme le montre la figure suivante :

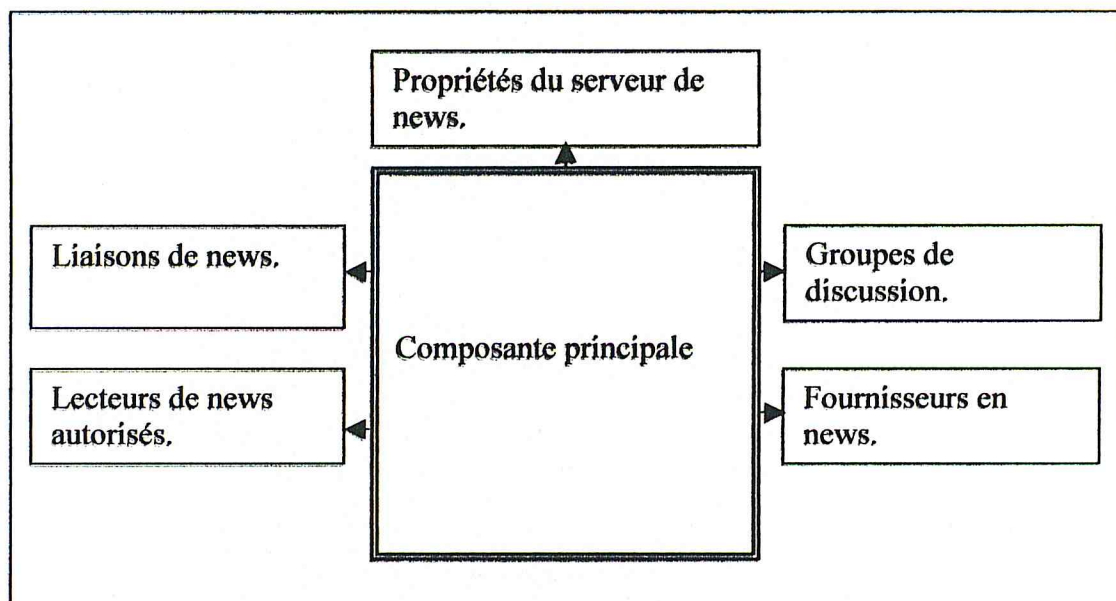


Figure-4.6 : schéma de conception du module de configuration du service de news.

Nous expliquons les rôles des différentes composantes de ce schéma :

La composante principale:

Elle permet de lancer toutes les autres composantes. Elle permet de visualiser le nom du serveur et les différentes listes des groupes de discussion, liaisons de news, fournisseurs en news et lecteurs de news autorisés. Elle permet aussi de redémarrer le serveur de news pour la prise en compte des modifications apportées.

La composante propriétés du serveur de news:

Elle permet d'examiner le fichier « */etc/news/inn.conf* » et visualiser les propriétés du serveur (domaine de noms du serveur, nom d'hôte et chemin apparaissant dans les articles postés, adresse électronique par défaut des modérateurs, taille maximale des articles acceptés, configuration comme serveur esclave et assignation du serveur maître, répertoire de configuration, liste des adresses électroniques des modérateurs de groupes de discussion, ...), elle permet la mise à jours de la liste des modérateurs des groupes de discussion et donne la possibilité de modifier ces propriétés et retourner à la composante principale.

La composante des groupes de discussion :

Elle permet d'ajouter des groupes de discussion en spécifiant toutes leurs propriétés (nom du groupe, adresse électronique du créateur, possibilité de postage local ou pas, groupe modéré, interdiction du postage de tout, description, numéro du plus petit article, ...), de supprimer des groupes de discussion ou de les modifier. Tout en maintenant les fichiers « */var/lib/news/newsgroups* », « */var/lib/news/active* » et « */var/lib/news/active.times* ». Elle permet aussi de valider les modifications apportées et retourner à la composante principale.

La composante des liaisons de news :

Elle permet d'ajouter des liaisons de news en spécifiant ses propriétés (nom du site et son nom de domaine, les groupes de discussion

concernés, taille maximale des news à fournir dans un seul transfert, taille maximale des articles expédiés, type de liaison, ...), de supprimer des liaisons de news ou de modifier ses propriétés en maintenant les fichiers « */etc/news/newsfeeds* » et « */etc/news/nntpsend.ctl* ». Elle permet aussi de valider les modifications apportées et retourner à la composante principale.

La composante des fournisseurs en news :

Elle permet d'ajouter des serveurs fournisseurs en news (pairs) ou bien des groupes fournisseurs qui contiennent plusieurs serveurs fournisseurs en news de ce serveur en spécifiant ses propriétés (nom d'hôte ou adresse IP du serveur, nombre maximal des connexions, groupes concernés, ...), de les supprimer et de les modifier en maintenant le fichier « */etc/news/incoming.conf* ». Elle permet aussi de valider les modifications apportées et retourner à la composante principale.

La composante des lecteurs de news autorisés :

Elle permet d'ajouter des lecteurs de news des hôtes autorisés en spécifiant ses propriétés (nom d'hôte ou adresse IP, groupes de discussion en lecture seulement, en postage seulement, ...), de les supprimer de la liste et de modifier ses propriétés en maintenant le fichier « */etc/news/nnrp.access* ». Elle permet aussi de valider les modifications apportées et retourner à la composante principale.

4.2.6- Module de configuration du service de partage des ressources Samba

Ce module doit fournir les fonctionnalités suivantes :

- La visualisation du nom de serveur **Samba** et la possibilité de consulter et de modifier ses propriétés globales.
- La visualisation de la liste des ressources partagées existantes et la possibilité d'ajouter, de supprimer ou de modifier une ressource.
- La visualisation de la liste des adresses **IP** des machines autorisées.
- La possibilité de redémarrer le serveur samba pour prendre en comptes les changements effectués par l'administrateur.

Nous avons vu ce module comme un ensemble de composantes comme le montre la figure suivante :

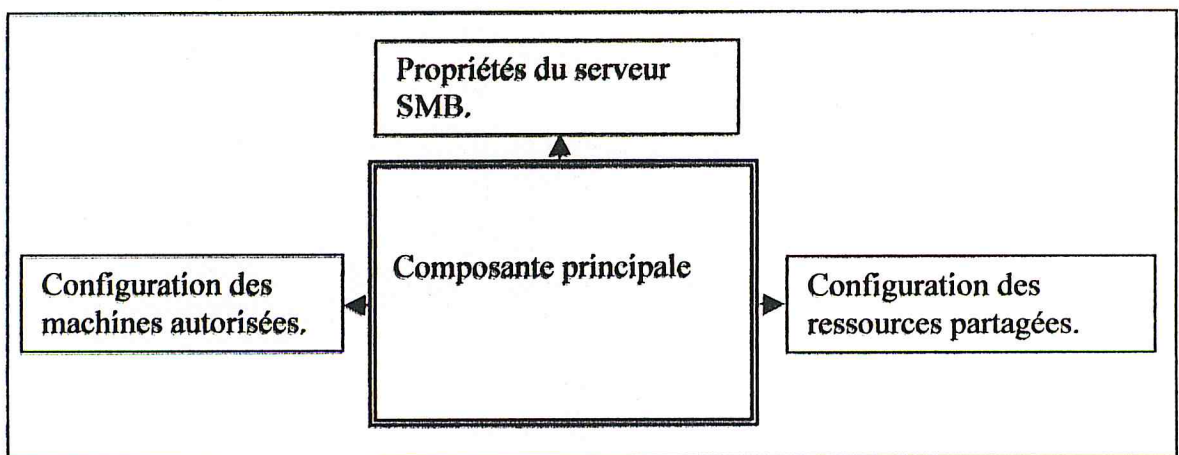


Figure-4.7: schéma de conception du module de configuration du service de partage de ressources **Samba**.

Nous expliquons les rôles des différentes composantes de ce schéma :

La composante principale :

Elle contient le nom du serveur, la liste des ressources partagées disponibles et la liste des adresses IP des machines ou des réseaux qui ont le droit de se connecter au serveur. Ces informations sont obtenues en examinant le fichier de configuration « */etc/samba/smb.conf* ». Cette composante permet à l'administrateur de consulter les propriétés globales du serveur en lançant la composante appropriée et de configurer (ajouter, supprimer ou modifier) des ressources partagées et d'ajouter ou de supprimer des adresses IP en lançant respectivement les composantes appropriées. Elle permet aussi de redémarrer le serveur pour prendre en compte les changements apportés par l'administrateur.

La composante des propriétés du serveur SMB :

Cette composante permet de consulter les propriétés globales du serveur SMB (nom du serveur, nom du groupe de travail, le mode d'impression, la taille du fichier journal, le niveau de sécurité, le mot de passe du serveur, l'emplacement du fichier des mots de passe, mot de passe crypté, ...) et la possibilité de les modifier et de revenir à la composante appropriée.

La composante de configuration des ressources :

Cette composante permet d'ajouter, de supprimer des ressources partagées ou de modifier ses propriétés (nom de la ressource, le commentaire sur la ressource, le chemin d'accès à la ressource, affichage de la ressource pour tous, ressource accessible à tous, ...). Elle permet aussi de retourner à la composante principale.

La composante des machines autorisées :

Cette composante permet d'ajouter ou de supprimer des adresses IP des machines ou des réseaux autorisés à utiliser le serveur. Cette composante permet aussi de retourner à la composante principale.

4.2.7- Module de configuration du service de routage et Fire-wall

Ce module doit fournir les fonctionnalités suivantes :

- La possibilité de visualiser le fichier de configuration global, de consulter les propriétés globales du Fire-wall, la possibilité de modifier ces propriétés et de configurer les paramètres de routage.
- La possibilité de visualiser toutes les entrées des divers fichiers de configuration du Fire-wall (zones, interfaces, hôtes, politiques de sécurité, des règles, ...), d'ajouter ou de supprimer ses entrées.
- La possibilité de redémarrer le système Fire-wall pour prendre en compte tous les changements effectués.

Ainsi, nous avons vu ce module comme le montre la figure suivante :

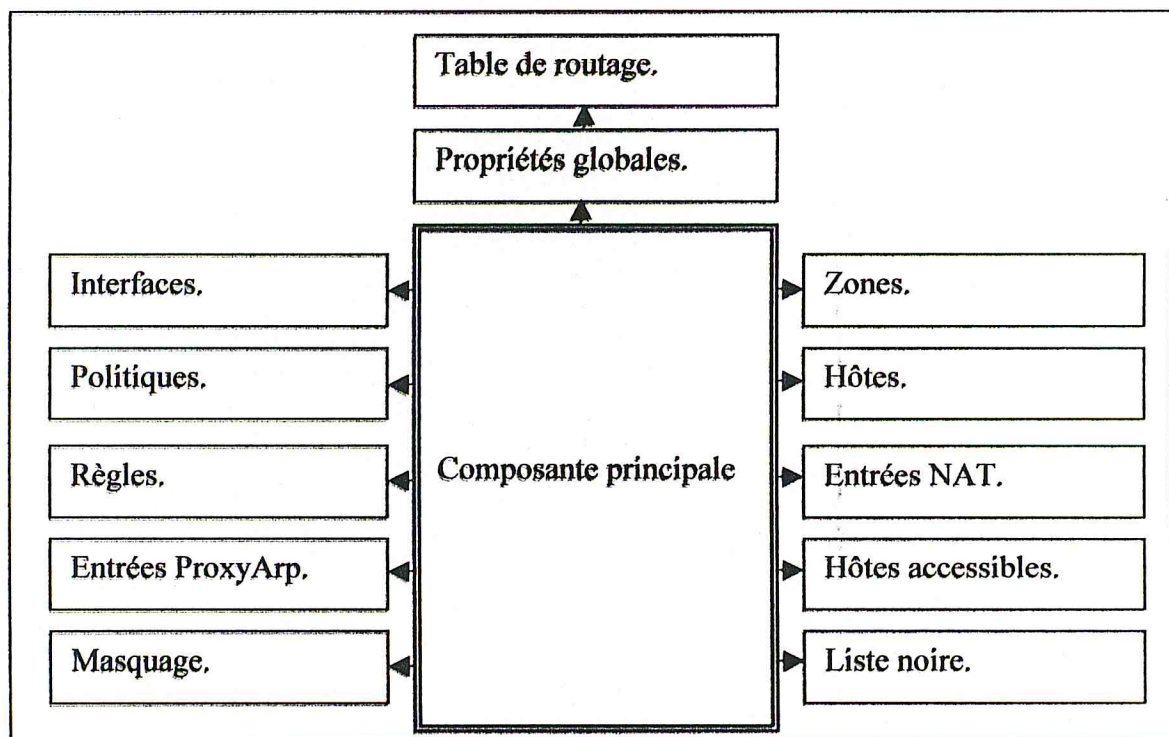


Figure-4.8 : Schéma de conception du module de configuration du service de routage et Fire-wall.

Nous expliquons les différentes composantes de ce schéma :

La composante principale :

Elle permet de visualiser les noms des divers fichiers de configuration du Fire-wall (fichier de configuration global « */etc/shorewall/shorewall.conf* », fichier des zones « */etc/shorewall/zones* », fichier des interfaces « */etc/shorewall/interfaces*, fichiers des hôtes « */etc/shorewall/hosts* », fichier des règles « */etc/shorewall/rules* », fichier des politiques de sécurité « */etc/shorewall/policy* », ...) et de configurer ses entrées en lançant les composantes appropriées. Elle permet aussi de redémarrer le serveur routeur/fire-wall pour prendre en considération toutes les modifications apportées aux différents fichiers de configuration.

La composante des propriétés globales :

Elle permet d'examiner le fichier de configuration global « */etc/shorewall/shorewall.conf* » et de visualiser toutes ses propriétés en donnant la possibilité à l'administrateur de les modifier. Elle permet aussi de valider les modifications à ce fichier et de retourner à la composante principale. Elle permet en plus, de lancer la composante de configuration de la table de routage.

La composante de configuration de la table de routage :

Elle permet d'examiner la table du routage du noyau, de retirer toutes ses entrées et de les visualiser à l'administrateur en lui donnant la possibilité d'ajouter ou de supprimer ces entrées (qui sont composées de plusieurs colonnes comme l'adresse IP du destination, l'adresse IP du passerelle, le masque du réseau, le nom de l'interface correspondante, ...). Elle permet aussi de valider les modifications apportées à cette table et de retourner à la composante précédente.

La composante des zones:

Elle permet de visualiser la liste des zones qui est contenue dans le fichier des zones « */etc/shorewall/zones* », d'ajouter ou de supprimer ses zones (ces zones sont composées du nom court de la zone qui est utilisé par le système, le nom complet de la zone et une description sur cette zone). Elle permet aussi de valider les différentes modifications apportées à ce fichier et retourner à la composante principale.

La composante interfaces :

Elle permet de visualiser les entrées de la table des interfaces qui est contenue dans le fichier « */etc/shorewall/interfaces* », d'ajouter ou de supprimer ses entrées (ces entrées sont composées de plusieurs entrées comme le nom de la zone correspondante à cette interface, le nom de l'interface et les options de l'interface comme le fonctionnement en **DHCP**, la vérification dans la liste noire de ses adresses **IP** correspondantes, l'emploi des adresses de la famille RFC 1918, ...). Elle permet aussi de valider les différentes modifications apportées à ce fichier et de retourner à la composante principale.

La composante des hôtes :

Elle permet de visualiser les entrées de la table des hôtes qui est contenue dans le fichier « */etc/shorewall/hosts* » (qui sont composées du nom de la zone correspondante, nom de l'interface correspondante suivi de l'adresse **IP**) l'emploi de ce fichier constitue une fonctionnalité supplémentaire pour le Fire-wall en terme des zones, il permet de spécifier plusieurs zones constituées chacune d'un ou de plusieurs hôtes mais toutes ces zones correspondent à une seule interface (ce qui est n'est pas le cas pour le fichier des interfaces qui nécessite une seule zone pour une interface donnée). Elle permet aussi d'ajouter et supprimer ses entrées. Elle permet en plus, de valider les modifications apportées à ce fichier et de retourner à la composante principale.

La composante des politiques de sécurité :

Elle permet de visualiser la liste des politiques de sécurité de la table des politiques de sécurité qui est contenue dans le fichier « */etc/shorewall/policy* », d'ajouter ou de supprimer des politiques de sécurité (une politique de sécurité constitue une entrée dans ce fichier et est composée du nom de la zone source, le nom de la zone destination, l'action à appliquer qui peut être Accepter, ignorer ou rejeter et le niveau de logging). Elle permet aussi de valider les différentes modifications apportées à ce fichier et de retourner à la composante principale.

La composante règles de sécurité :

Elle permet de visualiser la liste des règles de sécurité de la table des règles qui est contenue dans le fichier « */etc/shorewall/rules* », d'ajouter ou de supprimer des règles de sécurité (une règle de sécurité est toujours prioritaire à une politique de sécurité donnée et composée de plusieurs colonnes comme l'action à appliquer « Accepter, Ignorer, Rejeter, Rediriger ou DNAT », la zone source, l'adresse IP de l'hôte source ou du sous réseau particulier, ...). Elle permet aussi de valider les différentes modifications apportées à ce fichier et de retourner à la composante principale.

La composante de configuration du NAT :

Elle permet de visualiser les entrées de la table NAT qui est contenue dans le fichier « */etc/shorewall/nat* », d'ajouter ou de supprimer ses entrées (une entrée de ce fichier est composée de plusieurs colonnes comme l'adresse IP publique de l'hôte utilisant la méthode NAT, le nom de l'interface correspondante, l'adresse IP interne, autoriser toutes les interfaces du système Fire-wall, ...). Elle permet aussi de valider les différentes modifications apportées à ce fichier et de retourner à la composante principale.

La composante de configuration du ProxyArp :

Elle permet de visualiser la liste des entrées de la table ProxyArp contenue dans le fichier de configuration « */etc/shorewall/proxyarp* », d'ajouter ou de supprimer ses entrées (une entrée de ce fichier est composée de plusieurs colonnes comme l'adresse IP publique de l'hôte utilisant la méthode ProxyArp, l'interface interne, l'interface externe, ...).

Elle permet aussi de valider les différentes modifications apportées à ce fichier et de retourner à la composante principale.

La composante des hôtes accessibles :

Elle permet de visualiser la liste des entrées de la table des hôtes accessibles quand le Fire-wall est arrêté contenue dans le fichier « */etc/shorewall/routestopped* », d'ajouter ou de supprimer ses entrées (l'interface d'accès, l'adresse IP de l'hôte accessible). Elle permet aussi de valider les différentes modifications apportées à ce fichier et de retourner à la composante principale.

La composante de configuration du masquage (masquerading):

Elle permet de visualiser la liste des entrées de la table de masquage, (qui est généralement utilisée pour le partage de connexion Internet), contenue dans le fichier « */etc/shorewall/masq* », d'ajouter ou de supprimer ses entrées (interface de masquage, l'adresse IP locale, ...). Elle permet aussi de valider les différentes modifications apportées à ce fichier et de retourner à la composante principale.

La composante de configuration de la liste noire :

Elle permet de visualiser la liste des entrées de la liste noire contenue dans le fichier « */etc/shorewall/blacklist* ». Ce fichier contient les adresses IP des machines à interdire ou à ignorer leurs paquets passés par le Fire-wall. Elle permet aussi de valider les différentes modifications apportées à ce fichier et de retourner à la composante principale.

- la deuxième permettant de lancer les différents tests du bon fonctionnement ou de la bonne configuration de ces services réseaux grâce à des utilitaires Linux spécifiques de test. Elle permet aussi de donner les informations sur l'état d'exécution des serveurs, la possibilité de les démarrer ou les arrêter, et les statistiques des interfaces réseaux disponibles.

4.4- Conclusion

Dans ce chapitre nous avons abordé les étapes nécessaires à la conception de notre outil de configuration des divers services réseaux.

Ainsi, nous avons vu les détails de la conception des différents modules de configuration des services réseaux et nous avons vu aussi une description de la conception de l'interface utilisateur.

Dans le chapitre suivant nous allons aborder les différentes étapes de la réalisation de cet outil.

Chapitre 5

Réalisation

5.1- Introduction

Notre logiciel SERCONF est un logiciel de configuration des divers réseaux vus dans les chapitres précédents. Il permet en plus la configuration de ces services, le test de vérification du bon fonctionnement ou de la bonne configuration de ces services.

5.2- L'environnement de développement matériel

La réalisation de notre logiciel nécessite un PC doté d'une puissance suffisante pour l'installation de Linux et les divers serveurs fournissant les services réseaux vus dans les chapitres précédents. Ce PC doit disposer aussi de deux cartes réseaux reliées chacune d'elles à un autre PC Linux pour le test du fonctionnement du service de routage et Fire-wall

5.3- L'environnement de développement logiciel

▪ Le système d'exploitation

Nous avons choisi la distribution Linux Mandrake 9.0 qui est disponible sur le marché, elle utilise le noyau Linux 2.4.19. Cette distribution implémente tous les serveurs assurant les services réseaux à configurer par notre logiciel.

▪ Le langage de programmation

Notre choix s'est porté sur le langage **PHP**, acronyme récursif signifie « *PHP : Hypertext Processor* », qui est développé en 1994 par Rasmus Lerdorf (**PHP4** est la dernière version officielle). Il y a trois cas d'utilisation de PHP :

1. Scripts web (l'interpréteur **PHP** est un module du serveur web).

2. Scripts en ligne de commande (comme les scripts du *Shell*).
3. Application à interface graphique (comme *Delphi*).

PHP se voit plus adapté pour la réalisation de notre logiciel dans une architecture qui s'articule autour d'un serveur web (cas 1).

PHP est un langage de script HTML, son code est interprété et exécuté du côté serveur, ce qui le distingue des autres langages de script comme *Java script* dont son code est interprété et exécuté par le navigateur web. **PHP** diffère aussi des langages de scripts CGI (Common Gateway Interface) comme Perl qui impose d'écrire tout un programme pour générer une seule page web, avec **PHP**, il suffit d'inclure le code **PHP** entre une balise de début et une balise de fin dans une Page HTML pour permettre au navigateur web de passer en mode **PHP**.

PHP est indépendant du système d'exploitation car son code est exécuté par le serveur web, il a une syntaxe très proche à celle du langage C et s'inscrit directement dans des pages HTML.

L'inconvénient majeur de **PHP** est l'absence d'un débogueur.

PHP occupe une place importante dans le monde de l'Internet, plus de 500 000 sites web conçus avec **PHP** ont été recensés en 1999, malgré la concurrence des autres langages de script comme ASP (*Active Server Page*) et JSP (*Java Server Page*).

5.4- Problème rencontré et solution proposée

Au cours de la phase de programmation nous avons rencontré le problème suivant :

Les scripts PHP s'exécutent sous l'utilisateur système « *apache* » et non plus sous « *root* », ce qui mit la modification des fichiers de configurations et l'exécution de certaines commandes de redémarrage

et de teste des différents services impossible.

Pour pallier ce problème, il faut réaliser un serveur qui s'exécute en arrière plan au moment d'exécution de l'outil pour effectuer les modifications des fichiers de configurations et les commandes de test et de redémarrage des différents serveurs grâce à un protocole spécifique, l'outil lance une requête et le serveur s'assure de son identification, exécute la tâche demandée et lui envoie la réponse appropriée.

La réalisation de ce serveur peut être effectuée en utilisant le langage **PHP** comme script de ligne de commande (cas 2), mais, Actuellement, **PHP** ne dispose pas de module stable pour les *Sockets* (les versions les plus récentes de **PHP** comme la version 4.2.3 ou 4.3.1 utilisent aussi des modules expérimentaux pour les *Sockets*). Et pour ne pas avoir des problèmes au futur (changements de noms de fonctions, de nombre de paramètres, ...), nous avons choisi de réaliser ce serveur en utilisant le langage *Java* qui est connu par sa stabilité en ce qui concerne le domaine des *Sockets* grâce à son package « *java.lang.net* » d'une part et de l'autre part pour montrer l'interopérabilité de **PHP** avec d'autres langages de programmation.

5.5- Présentation de l'interface utilisateur

Notre outil permet de configurer les différents services réseaux vus précédemment, la configuration de chacun de ces services est assurée par un module spécifique.

Chaque module présente plusieurs interfaces de configuration et en plus des interfaces des modules de configuration, notre outil comporte aussi des interfaces pour les informations sur les états des serveurs et les statistiques des cartes réseaux et aussi pour le test du bon fonctionnement ou la bonne configuration de ces services.

Nous présentons dans ce qui suit les interfaces les plus importantes des modules de configurations de notre logiciel.

5.5.1- L'interface principale

Elle représente la page d'accueil de notre logiciel. Elle nous permet de lancer les différents modules de configuration des services réseaux vus dans les chapitres précédents ainsi que les différents tests sur ces services réseaux par des simples clics sur les icônes ou les liens appropriés. Elle permet aussi de consulter les informations sur les différents serveurs et les statistiques des cartes réseaux disponibles.

La figure-5.1 montre la page d'accueil de notre logiciel.

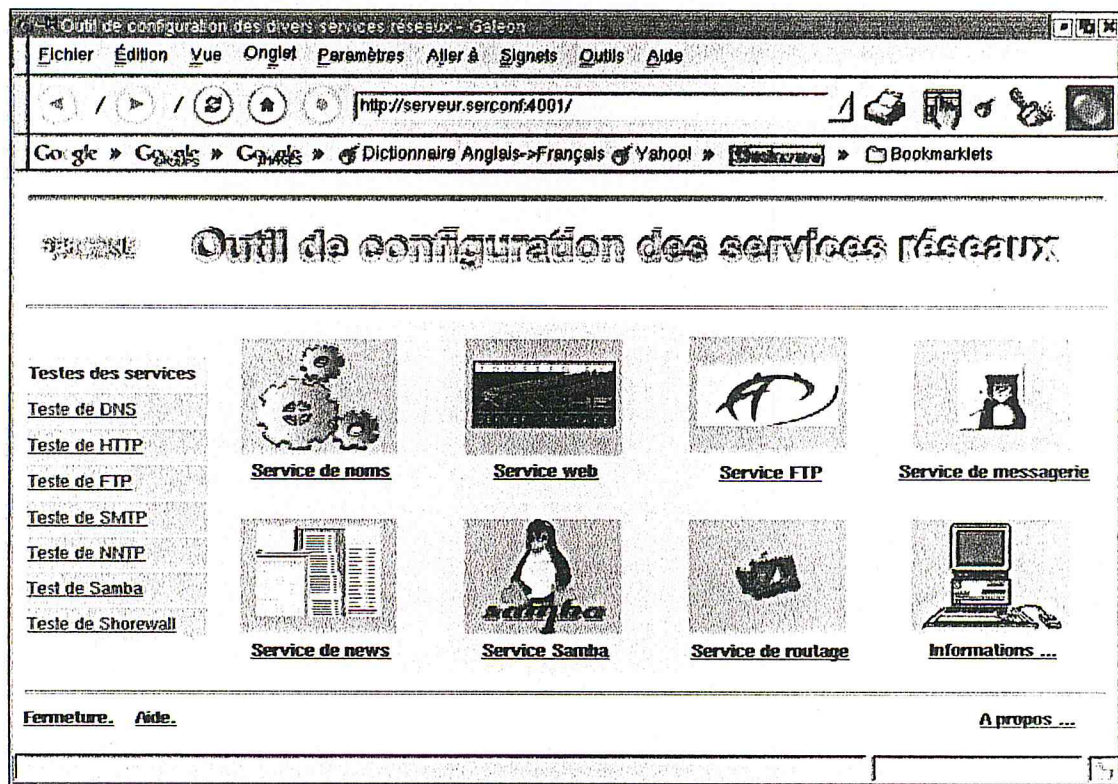


Figure-5.1 : Interface principale de l'outil de configuration des divers services réseaux sous Linux.

5.5.2- Interface du module de configuration du service de noms

La figure-5.2 montre l'interface de configuration du service de noms. L'utilisateur peut consulter les propriétés de son serveur en cliquant sur le bouton « Propriétés », peut voir la liste des zones disponibles sur son serveur et peut ajouter une zone en cliquant sur le bouton « Ajouter », supprimer une zone en cliquant sur le bouton « Supprimer » ou modifier une zone en cliquant sur le bouton « Modifier » pour modifier ses propriétés et sa liste des enregistrements de ressource. Il peut aussi consulter la page d'aide sur la configuration de ce service en cliquant sur le bouton « Aide » ou redémarrer le serveur de noms pour prendre en compte les changements apportés en cliquant sur le bouton « OK ».

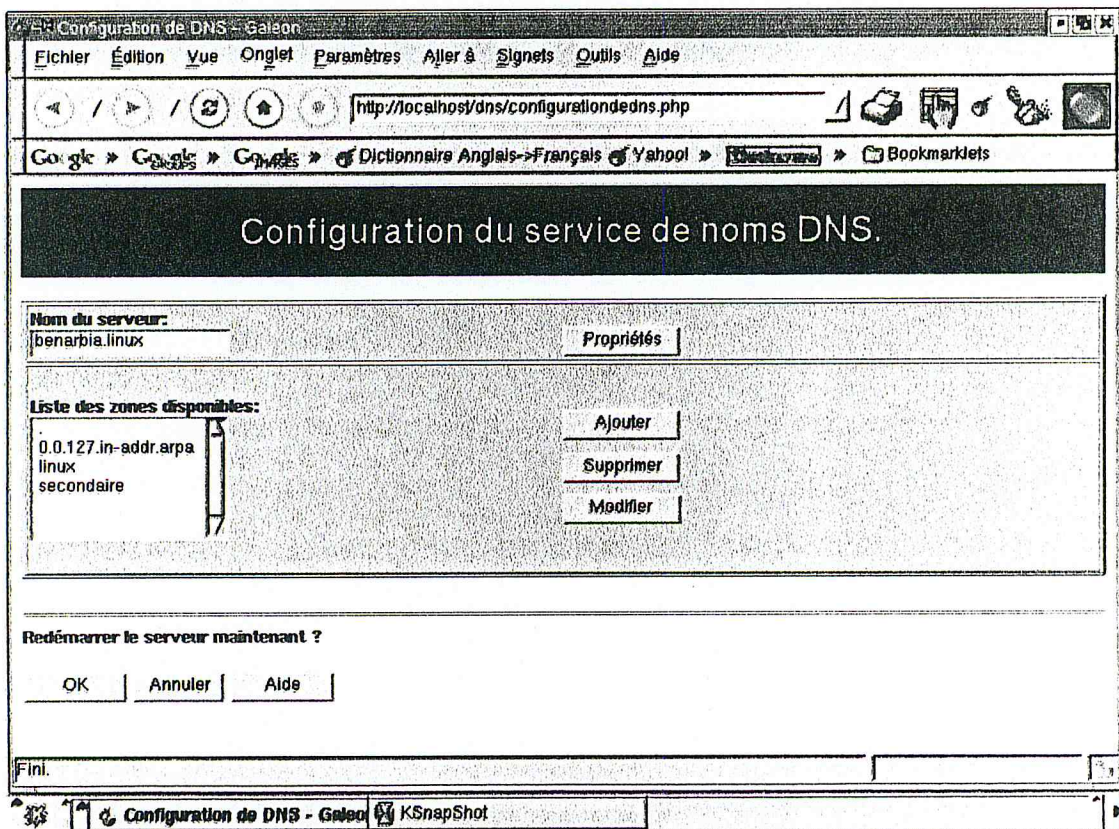


Figure-5.2 : Interface du module de configuration du service de noms.

La figure-5.3 montre la page des propriétés du serveur de noms, l'utilisateur peut modifier les différentes propriétés du serveur (port, l'utilisation du Fire-wall, des redirecteurs, ...) et cliquer sur le bouton « OK » pour valider ses modifications et peut aussi consulter la page d'aide sur les propriétés du serveur en cliquant sur le bouton « Aide ».

Figure-5.3 : La page des propriétés du serveur de noms.

Les figures 5.4, 5.5, 5.6, et 5.7 montrent respectivement l'interface d'ajout d'une zone primaire, de modification d'une zone primaire, d'ajout d'un enregistrement de ressource et de test de bon fonctionnement du service de noms.

Ajout d'une zone primaire - Gafson

Fichier Édition Vue Onglet Paramètres Aller à Signets Outils Aide

http://localhost/dns/ajouterzone.php

Google » Google » Google » Dictionnaire Anglais-Français » Yahoo! » Bookmarks

Ajout d'une zone primaire.

Nom de la zone: Le nom de la zone correspond au domaine de nom à créer. Pour une zone inverse, mettre le nom de la zone sous la forme x.in-addr.arpa où x est le numéro de réseau inversé.

Fichier de la zone:

Notifier les serveurs esclaves: ☒ Yes Temps de vie (TTL): jours

Adresse électronique de l'administrateur: Numéro de série:

Temps de rafraîchissement: secondes ☒ Temps d'essai: secondes ☒

Temps d'expiration: secondes ☒ Temps minimum: secondes ☒

☐ Autoriser seulement les machines suivantes: Entrer les adresses IP des machines autorisées à utiliser cette zone, chaque adresse IP doit être terminée par un point virgule (;). Par défaut toute les machines du réseau sont autorisées.

☐ Entrer les adresses IP des serveurs autorisés à transférer cette zone.

Fin.

Figure-5.4 : La page d'ajout d'une zone primaire.

Propriétés de la zone - Gafson

Fichier Édition Vue Onglet Paramètres Aller à Signets Outils Aide

http://localhost/dns/propdezone.php

Google » Google » Google » Dictionnaire Anglais-Français » Yahoo! » Bookmarks

Temps d'expiration: secondes ☒ Temps minimum: secondes ☒

Temps d'expiration: secondes ☒ Temps minimum: secondes ☒

☐ Autoriser seulement les machines suivantes: Entrer les adresses IP des machines autorisées à utiliser cette zone, chaque adresse IP doit être terminée par un point virgule (;). Par défaut toutes les machines du réseau sont autorisées.

☐ Autoriser le transfert seulement pour: Entrer les adresses IP des serveurs autorisés à transférer cette zone, chaque adresse IP doit être terminée par un point virgule (;). Par défaut tous les serveurs sont autorisés.

Élu	Nom de domaine	Casse	Type	Information
☒	linux.	IN	NS	benarbia
☒	benarbia	IN	A	192.168.1.1
☒	linux.	IN	MX	10 mail
☒	ben	IN	CNAME	benarbia
☒	linux.	IN	MX	20 smtp

Ajouter Supprimer Modifier

OK Annuler Aide

Fin.

Figure-5.5 : La page de modification d'une zone primaire.

The screenshot shows a web browser window titled 'Enregistrements - Galseon'. The address bar displays 'http://localhost/dns/enregistrements.php'. The page has a menu bar with 'Fichier', 'Édition', 'Vue', 'Onglet', 'Paramètres', 'Aller à', 'Signets', 'Outils', and 'Aide'. Below the menu is a search bar with 'Google' and 'Dictionnaire Anglais->Français' suggestions. The main content area has a dark header with the text 'Enregistrement d'échangeur de messagerie.' Below this is a form with three input fields: 'Nom de domaine:' with the value 'linux', 'Nom du serveur de messagerie électronique:' with the value 'smtp', and 'Priorité:' with the value '20'. At the bottom of the form are three buttons: 'OK', 'Rétablir', and 'Annuler'. The status bar at the bottom shows 'Fin.'.

Figure-5.6 : La page d'ajout d'un enregistrement de ressource MX.

The screenshot shows a web browser window titled 'Teste de bon fonctionnement du service de noms - Galseon'. The address bar displays 'http://localhost/dns/teste-dns.php'. The page has the same menu bar as Figure 5.6. The main content area has a dark header with the text 'Teste de bon fonctionnement du service de noms DNS.' Below this is a form with two input fields: 'Nom de domaine:' with the value 'benarbia.linux' and 'Paramètre de teste:' with a dropdown menu showing 'Enregistrement A'. Below the form is a section titled 'Résultats du teste:' containing the following text: 'Requête N° : 49125', 'Statu de la requête : NOERROR', 'Question : benarbia.linux. IN A', 'Réponse : benarbia.linux. 259200 IN A 192.168.1.1', 'Temps de la requête : 1146 msec', 'Date de lancement de la requête : Thu Jul 31 17:52:28 2003', and 'Taille du message reçu : 62 octets'. At the bottom of the results section are two buttons: 'OK' and 'Annuler'. The status bar at the bottom shows 'Fin.'.

Figure-5.7 : La page de test de bon fonctionnement du service de noms.

5.5.3- Interface du module de configuration du service web

La figure-5.8 montre l'interface du module de configuration du service web. Ainsi, l'utilisateur peut voir le nom de son serveur web, la liste des adresses IP et ports d'écoute et la liste des hôtes virtuels disponibles. Il peut consulter les propriétés de son serveur en cliquant sur le bouton « Propriétés », peut ajouter, supprimer ou modifier une adresse IP ou un port d'écoute en cliquant sur les boutons appropriés et il peut aussi ajouter, supprimer ou modifier un hôte virtuel en cliquant sur les boutons appropriés.

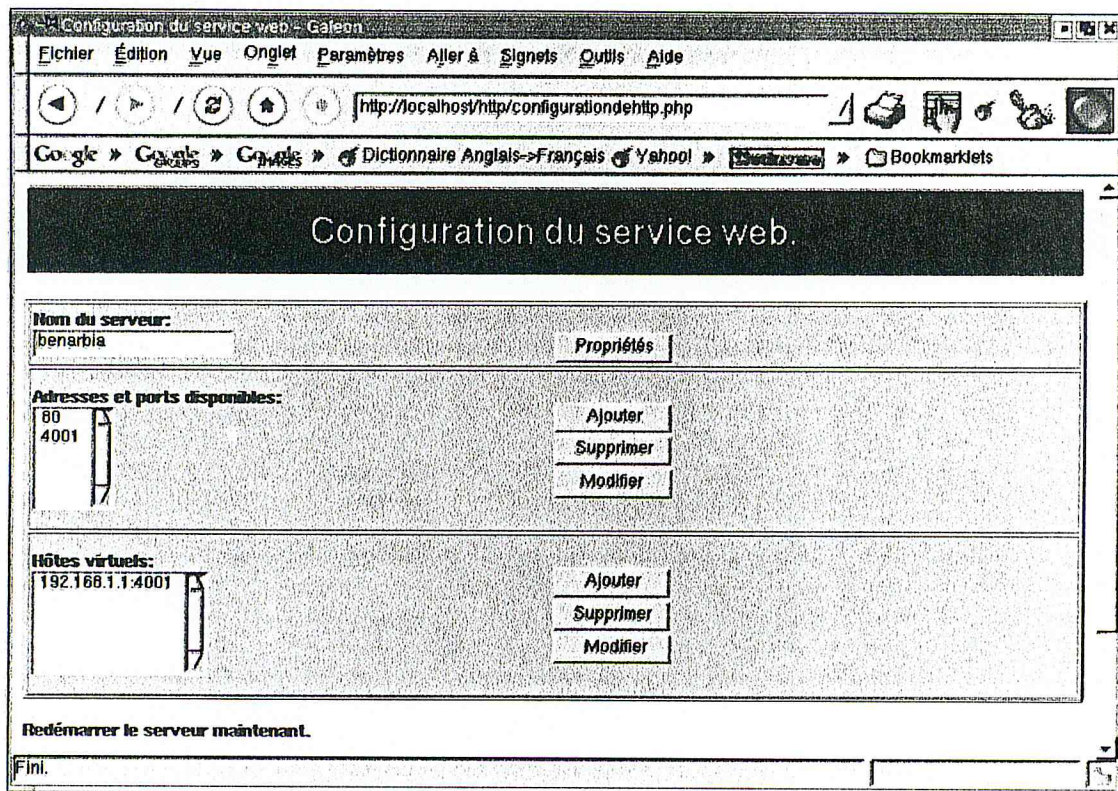


Figure-5.8 : Interface du module de configuration du service web.

La figure-5.9 montre la page des propriétés (adresse de l'administrateur, répertoire racine, nombre de processus, temps d'attente des connexions, autorisation des connexions persistantes, ...) du serveur web où l'utilisateur peut modifier ces propriétés en cliquant sur le bouton « OK », il peut aussi consulter les paramètres avancés en cliquant sur le bouton « Paramètres... » et consulter la page d'aide sur les propriétés du serveur en cliquant sur le bouton « Aide ».

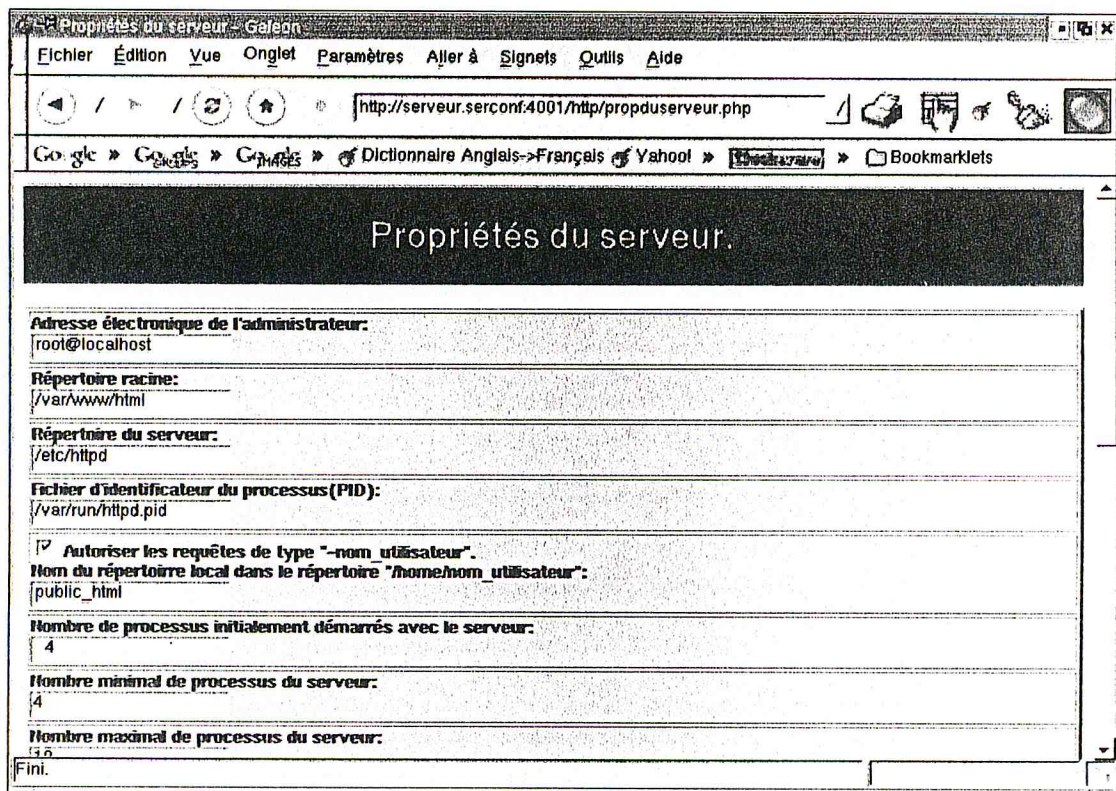


Figure-5.9 : La page des propriétés du serveur web.

Les figures 5.10, 5.11, 5.12 et 5.13 montre respectivement la page de sélection des classes des paramètres avancés du serveur, la page d'une classe de paramètres avancés qui est la classe de configuration du site, la page d'ajout d'un hôte virtuel et enfin la page d'une classe de paramètres avancés des hôtes virtuels qui est la classe des paramètres SSL.

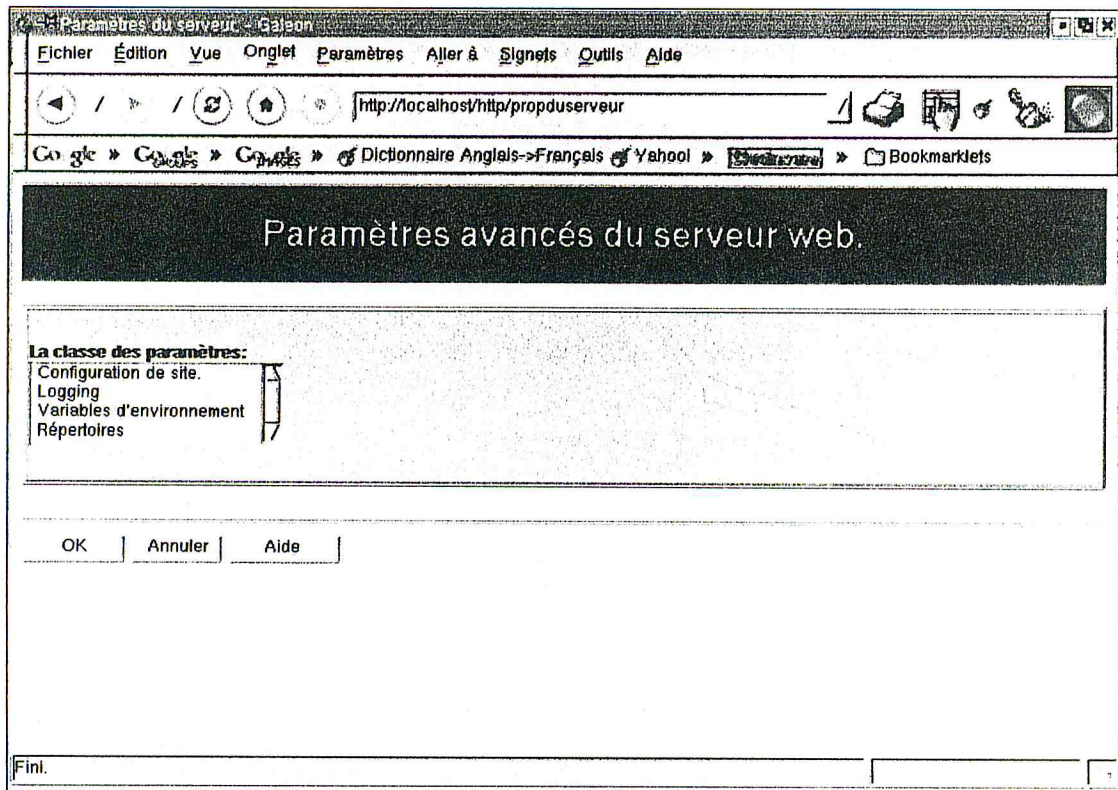


Figure-5.10 : La page de sélection des classes de paramètres du serveur.

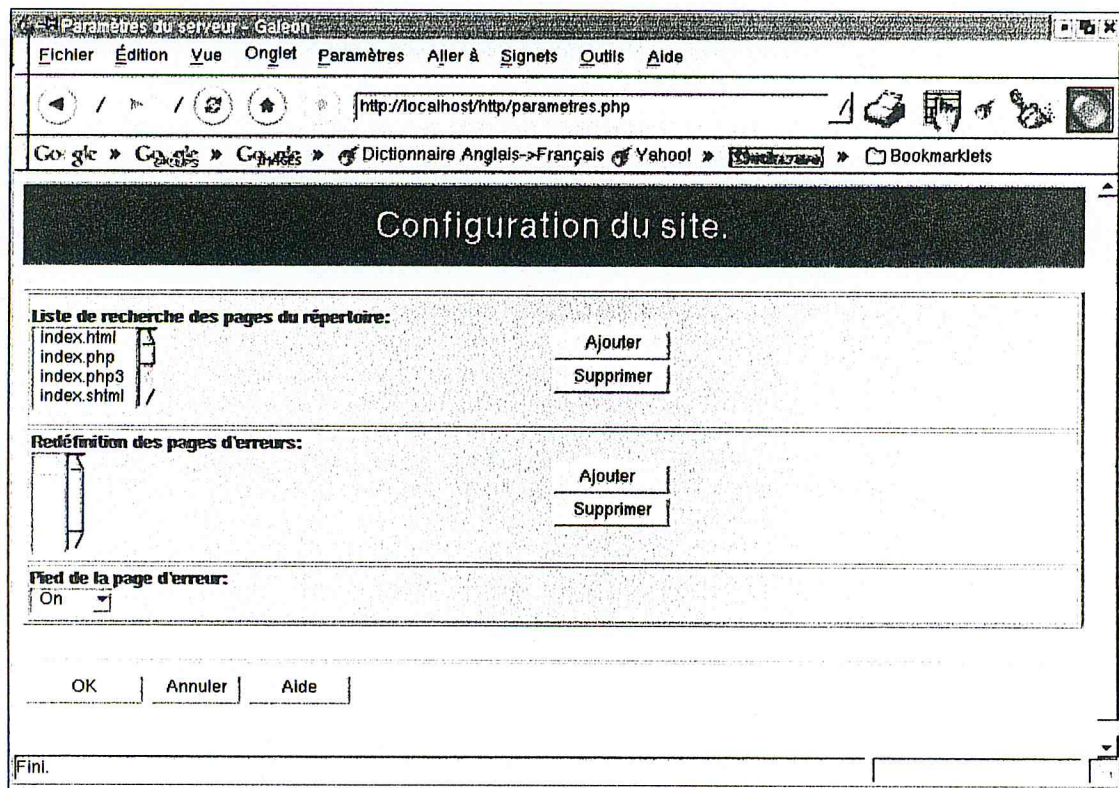


Figure-5.11 : La page de configuration des paramètres de site.

Ajout d'un hôte virtuel.

Type de l'hôte virtuel:

Adresse IP de l'hôte virtuel:

Numéro de port de l'hôte virtuel:

Nom du serveur de l'hôte virtuel:

Répertoire racine:

Adresse électronique de l'administrateur:

OK Rétablir Annuler Aide

Fin.

Figure-5.12 : La page d'ajout d'un hôte virtuel.

Support SSL.

☒ Utiliser le support SSL.

Fichier du certificat SSL:

Fichier de la clé SSL:

Transfer de logging SSL au fichier:

Type de l'authentification SSL:

Nombre de vérifications:

Options SSL:

- ☐ Autoriser l'authentification basique.
- ☐ Exporter les certificats.
- ☐ Exporter les variables d'environnement compatibles.
- ☐ Optimisation des négociations SSL.

Fin.

Figure-5.13 : La page de configuration des paramètres SSL.

5.5.4- Interface du module de configuration du service FTP

La figure-5.14 montre l'interface du module de configuration du service FTP. Ainsi, l'utilisateur peut voir le nom du serveur FTP, la liste des limites d'autorisation, la liste des répertoires d'accès et aussi la liste des hôtes virtuels disponibles et peut ajouter, supprimer ou modifier ses entrées en cliquant sur les boutons appropriés, il peut aussi consulter les propriétés du serveur FTP en cliquant sur le bouton « Propriétés », il peut aussi activer et configurer l'accès anonyme en cochant la case appropriée et en cliquant sur le bouton « Configurer ». Enfin, il peut redémarrer le serveur FTP en cliquant sur le bouton « OK » ou consulter la page d'aide de configuration du service FTP en cliquant sur le bouton « Aide ».

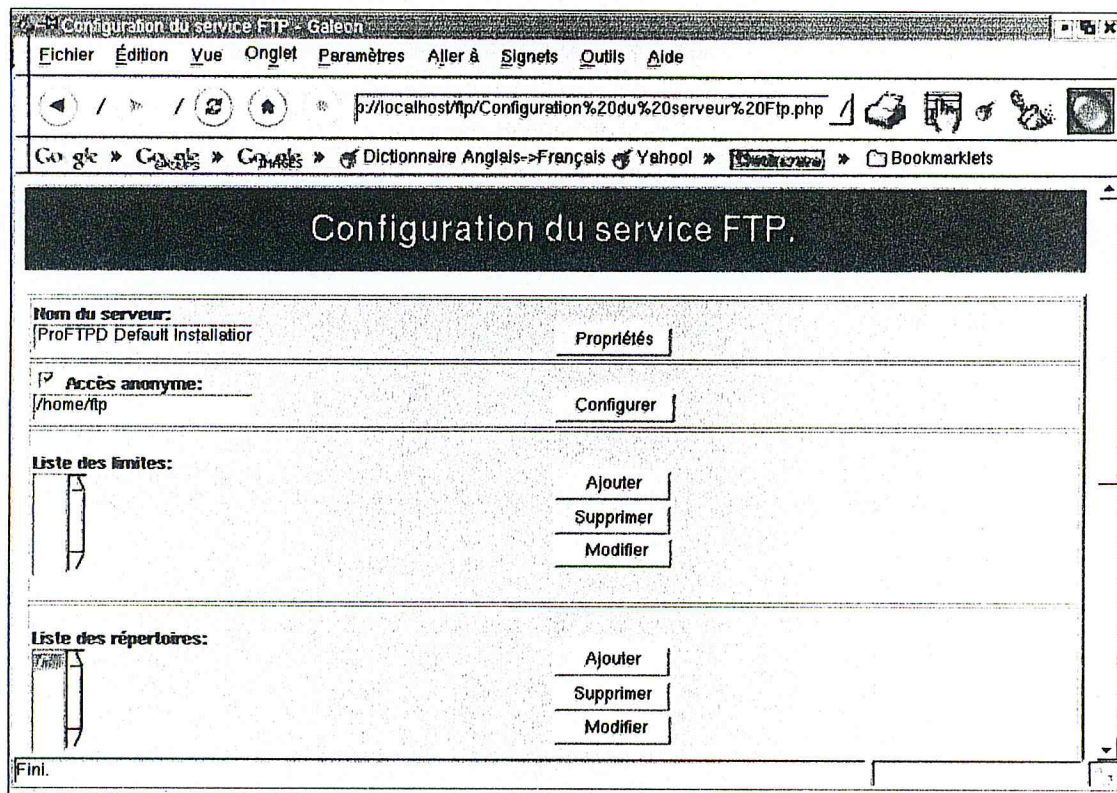


Figure-5.14 : Interface du module de configuration du service FTP.

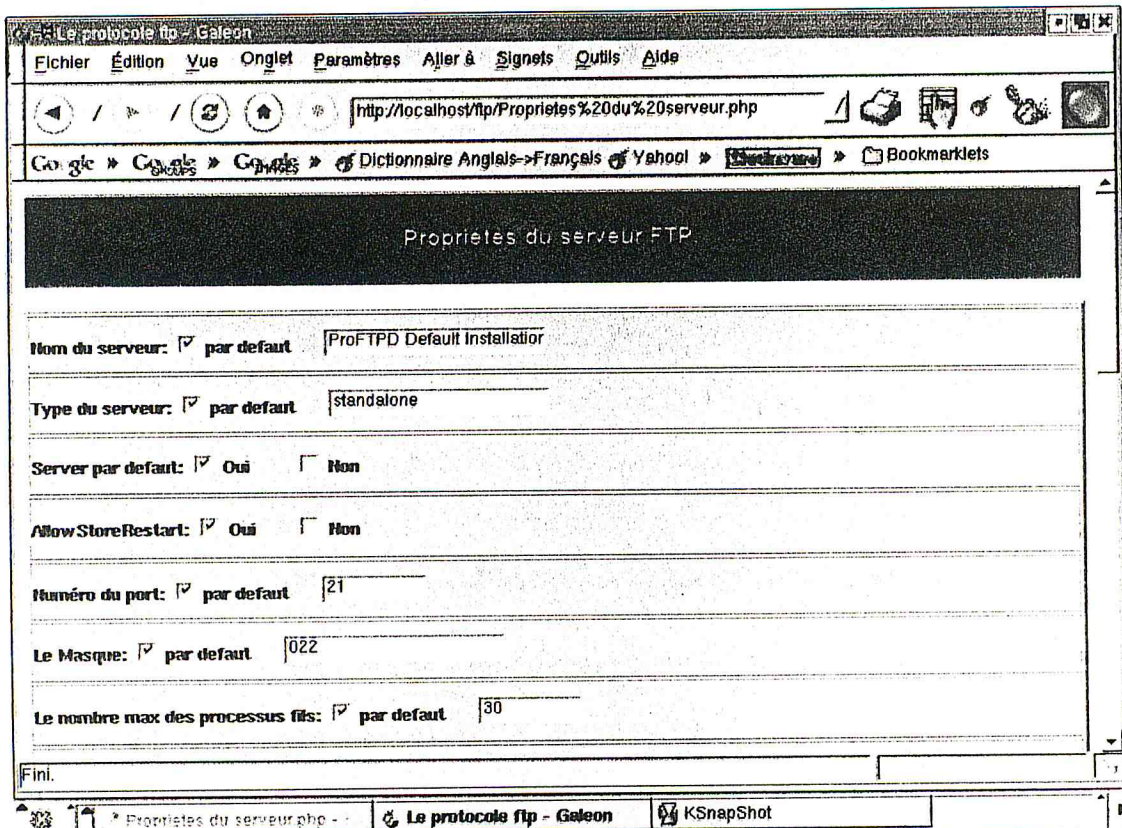


Figure-5.15 : La page des propriétés du serveur FTP.

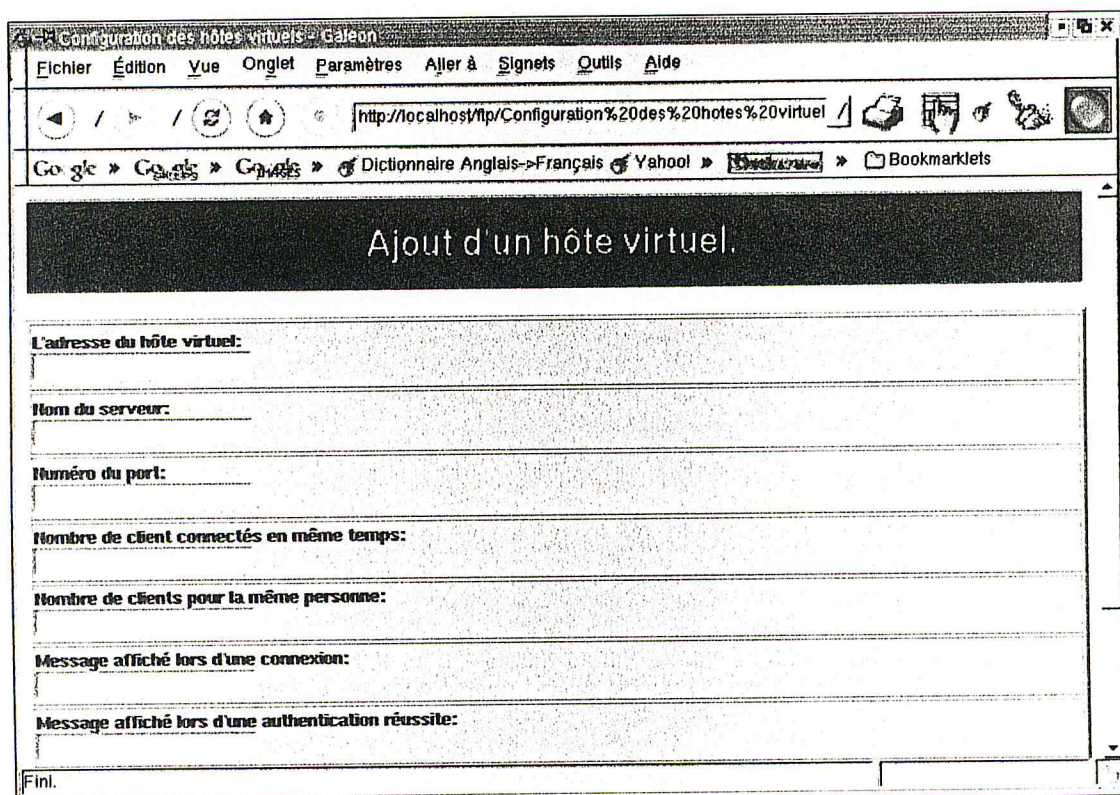


Figure-5.16 : La page d'ajout d'un hôte virtuel FTP.

5.5.5- Interface du module de configuration du service de messagerie électronique

La figure-5.17 montre l'interface de configuration du service de messagerie électronique. Ainsi, l'utilisateur peut voir le nom de son serveur SMTP, consulter ses propriétés globales en cliquant sur le bouton « Propriétés » et à partir de la page des propriétés peut consulter les paramètres SMTP du serveur. Il peut aussi activer (ou désactiver) et configurer les supports des aliases, de noms canoniques, des domaines virtuels et le support de gestion des utilisateurs déplacés en cochant (ou en décochant) les case appropriées et en cliquant sur les boutons appropriés. Il peut de plus, redémarrer le serveur ou consulter la page d'aide en cliquant sur les boutons « OK » ou « Aide » respectivement.

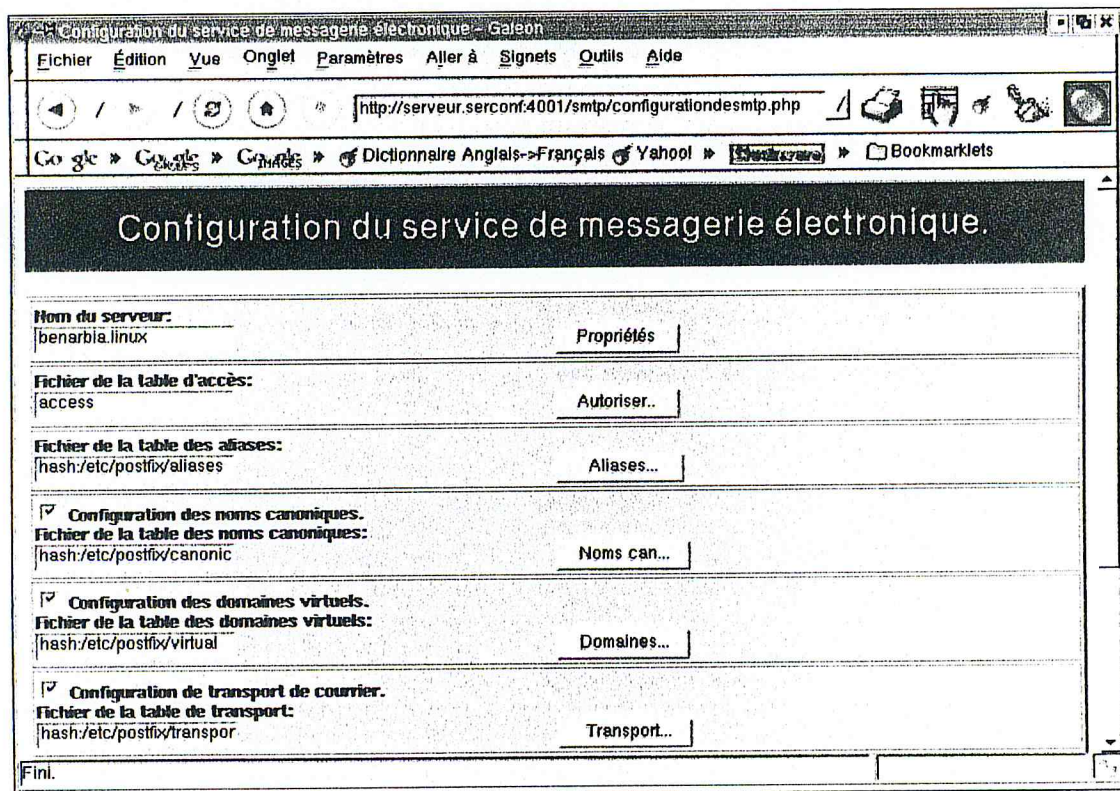


Figure-5.17 : Interface du module de configuration du service de messagerie électronique.

Les figures 5.18 et 5.19 montrent la page de propriétés du serveur SMTP et la page de configuration des aliases respectivement.

Propriétés du serveur - Galeon

Fichier Édition Vue Onglet Paramètres Aller à Signets Outils Aide

http://localhost/smtp/propduserveur.php

Google » Google » Google » Dictionnaire Anglais->Français » Yahoo! » Bookmarks » Bookmarks

Propriétés du serveur.

Domaine de noms du courrier sortant: \$myhostname	Les classes des erreurs à rapporter au postmaster: resource, software
Domaine de noms du serveur: benarbia.linux	Les interfaces réseaux du serveur: locale
Domaines de noms dont le serveur reçoit leur courrier: \$myhostname, localhost Spécifier les domaines de noms séparés par des virgules et/ou des espaces.	Les réseaux de confiance: 127.0.0.0/8 Spécifier les adresses IP des réseaux dont le serveur délivre leur courrier à son destination séparées par des virgules et/ou des espaces.
☒ Délivrance directe du courrier à son destination. Délivrance du courrier via:	☐ Utiliser le masquage des adresses. Masquer les noms d'hotes par le domaine:
Répertoire de la file: /var/spool/postfix	Temps maximal d'asservissement d'une requête: 18000s (secondes).
Répertoire des commandes: /usr/sbin	Temps d'attente après une déconnexion IPC: 100s (secondes).
Répertoire de spool: /var/spool/mail	Temps d'une communication I/O interne: 3600s (secondes).

Fin.

Figure-5.18 : La page des propriétés du serveur SMTP.

Configuration des alias - Galeon

Fichier Édition Vue Onglet Paramètres Aller à Signets Outils Aide

http://localhost/smtp/alias.php

Google » Google » Google » Dictionnaire Anglais->Français » Yahoo! » Bookmarks » Bookmarks

Configuration des alias.

Liste des alias disponibles:

postmaster: root	Ajouter Supprimer
daemon: root	
games: root	
ingres: root	
nobody: root	
system: root	
toor: root	

OK Annuler Aide

Fin.

Figure-5.19 : La page de configuration des alias.

5.5.6- Interface de configuration du service de news

La figure-5.20 montre l'interface de configuration du service de news. Ainsi, l'utilisateur peut voir le nom de son serveur de news et consulter ses propriétés en cliquant sur le bouton « Propriétés ». Il peut aussi ajouter, supprimer ou modifier les entrées des listes des groupes de discussion, des liaisons des news, des fournisseurs en news et des lecteurs autorisés en cliquant sur les boutons appropriés et en maintenant toutes les colonnes de ces entrées. Il peut de plus, redémarrer le serveur de news ou consulter la page d'aide sur la configuration de ce service en cliquant sur les boutons « OK » ou « Aide » respectivement.

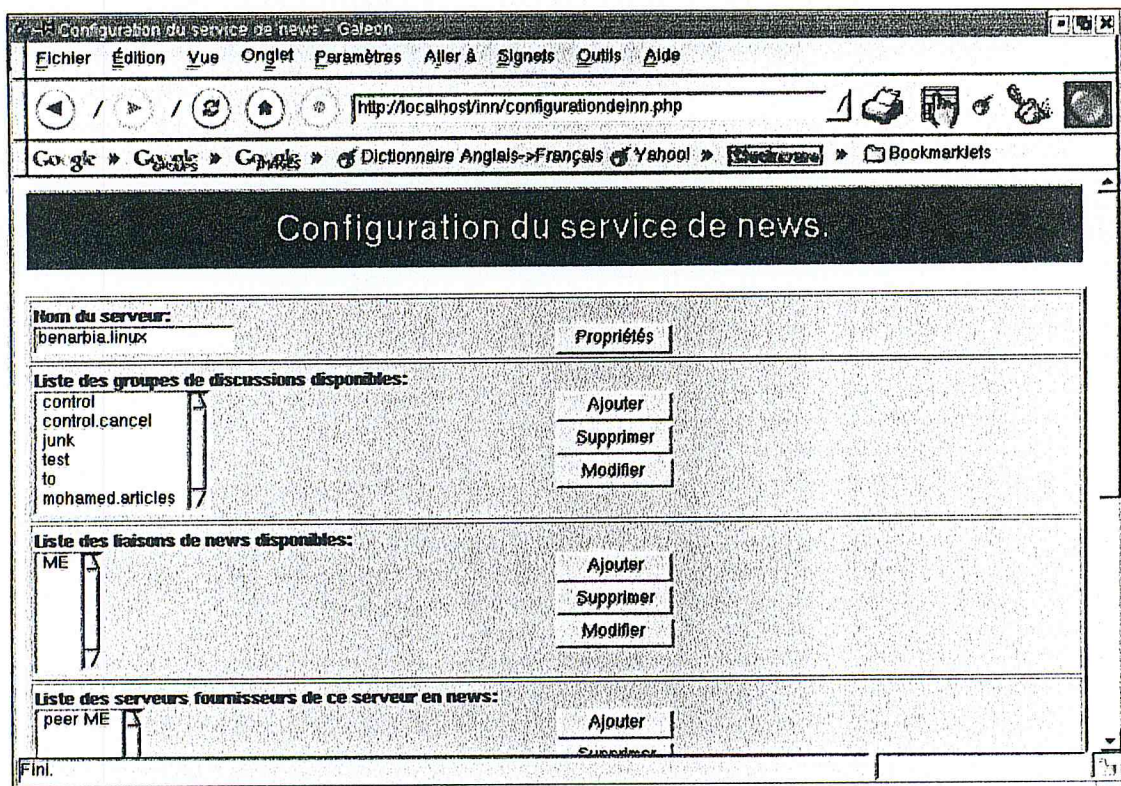


Figure-5.20 : l'interface du module de configuration du service de news.

Les figures 5.21 et 5.22 montre la page des propriétés du serveur de news et la page d'ajout d'un groupe de discussion.

Domaine de noms du serveur: linux	Nom d'hôte apparaissant dans le champ "From": benarbia.linux
Organisation: A poorly-installed Internet	Le chemin apparaissant dans le champ "Path": linux
Adresse électronique par défaut des modérateurs: %s@moderators	Agent de transport de courrier (mta): /usr/sbin/sendmail -oi -ot
Numéro de port: 119	Nombre maximal de connexions: 50
Taille maximale d'un article accepté par le serveur: 1000000	Taille maximale d'un article local posté localement: 1000000
Répertoire de configuration: /etc/news	Répertoire de spool: /var/spool/news
Répertoire de logging: /var/log/news	Répertoire de la vue d'ensemble: /var/spool/news/overview
Répertoire des articles entrants: /var/spool/news/incoming	Répertoire des articles sortants: /var/spool/news/outgoing

Fin.

Figure-5.21 : La page des propriétés du serveur de news.

Nom du groupe de discussion:

Le plus petit numéro de ses articles actifs:

Le plus grand numéro de ses articles actifs:

Adresse électronique du créateur:

Option du groupe:

- ☐ Autoriser le postage local des articles
- ☐ Interdire le postage local des articles
- ☐ Interdire le postage des articles
- ☐ Groupe modéré
- ☐ Ne pas conserver les articles

Description:

Fin.

Figure-5.22 : La page d'ajout d'un groupe de discussion.

5.5.7- Interface du module de configuration du service Samba

La figure-5.23 montre l'interface du module de configuration de service **Samba**. Ainsi, l'utilisateur peut voir le nom de son serveur et peut consulter ses propriétés en cliquant sur le bouton « Propriétés ». Il peut aussi voir la liste des ressources partagées et ajouter, supprimer ou modifier une ressource en cliquant sur le bouton approprié. Il peut aussi voir la liste des adresses IP des machines autorisées à utiliser le serveur et ajouter, supprimer ou modifier une entrée de cette liste. Il peut de plus, redémarrer le serveur en cliquant sur le bouton « OK » ou consulter la page d'aide sur la configuration de ce service en cliquant sur le bouton « Aide ».

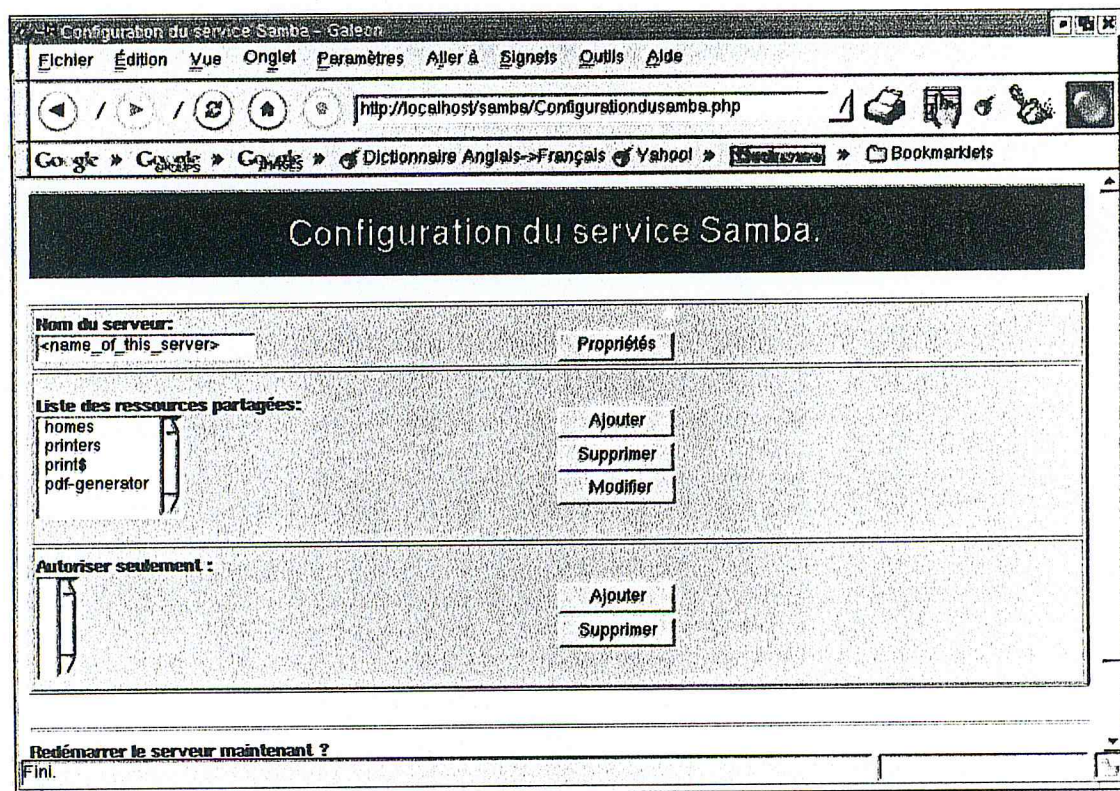


Figure-5.23 : Interface module de configuration du service Samba.

Les figure 5.24 et 5.25 montrent la page des propriétés du serveur **SMB** et la page d'ajout d'une ressource à partager respectivement.

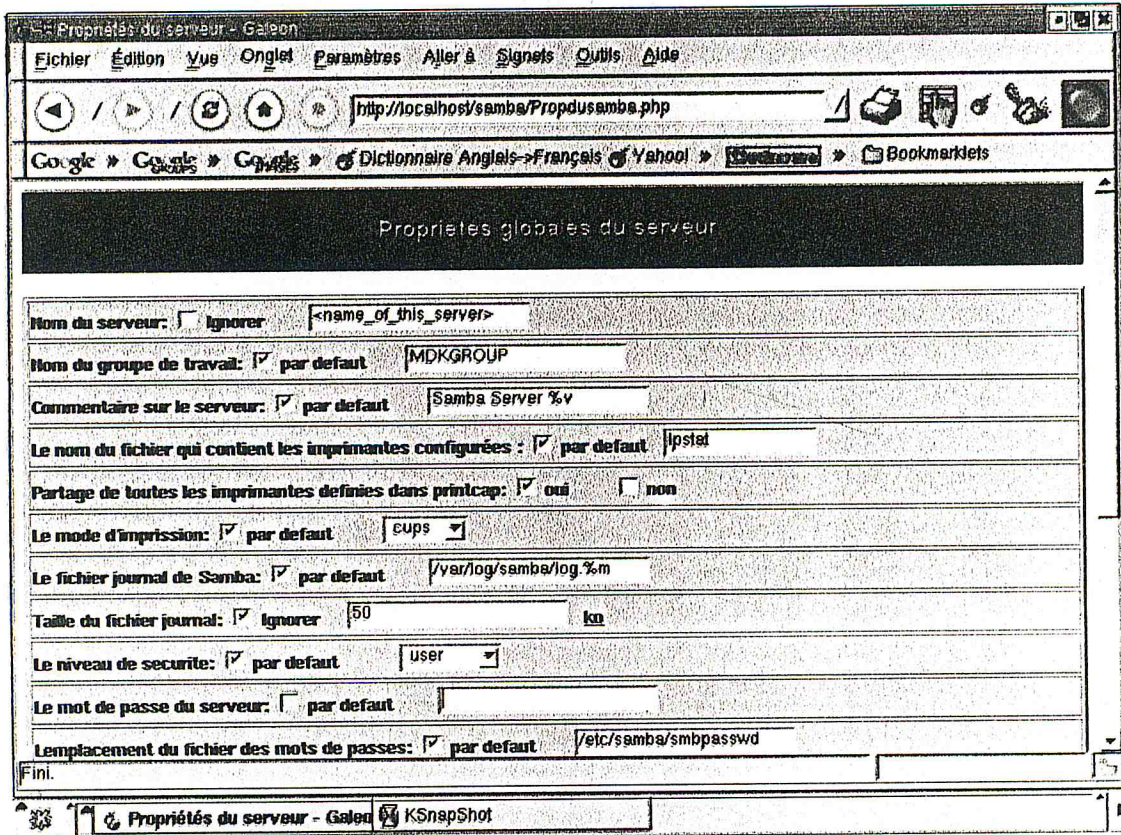


Figure-5.24 : La page des propriétés du serveur SMB.

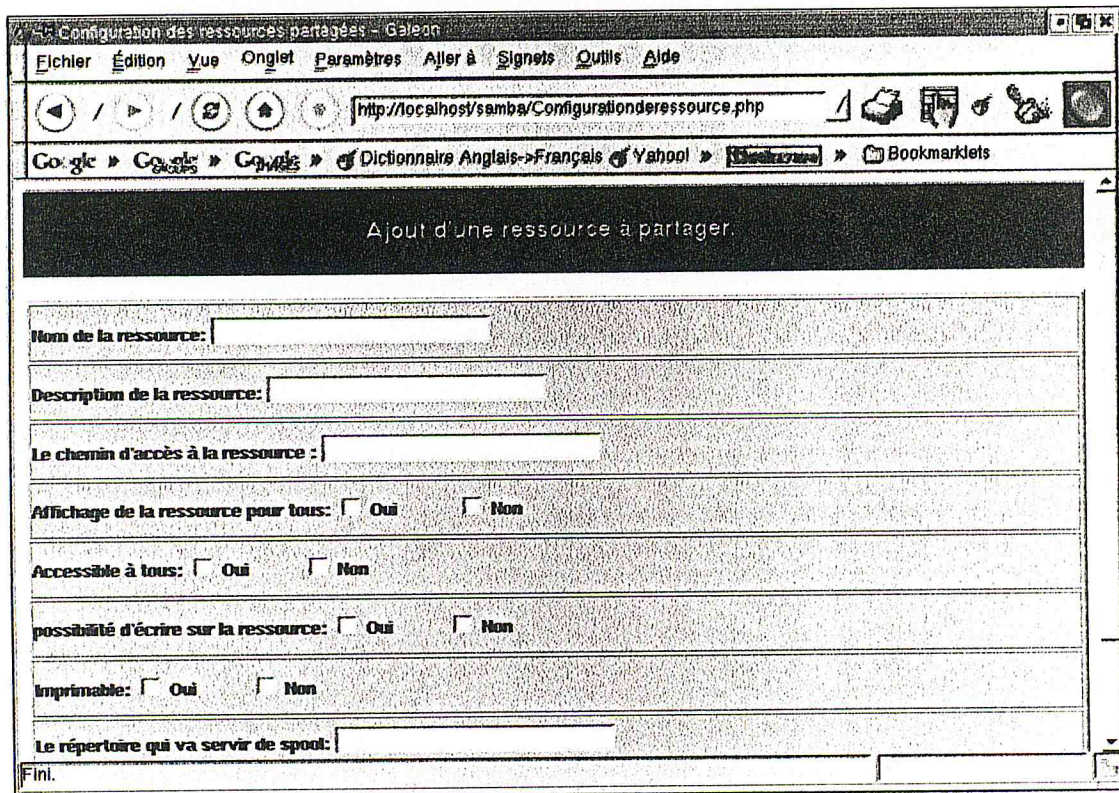


Figure-5.25 : La page d'ajout d'une ressource à partager.

5.5.8- Interface de configuration du service de routage et Fire-wall

La figure-5.26 montre l'interface du module de configuration du service de routage et Fire-wall. Ainsi, l'utilisateur peut consulter les propriétés du serveur routeur/Fire-wall en cliquant sur le bouton « Propriétés ». Il peut aussi voir tous les fichiers contenant les différentes tables des informations (table des zones, des politiques de sécurité, des règles de sécurité, ...) utilisées par ce service et configurer ces tables en cliquant sur les boutons appropriés. Il peut de plus, redémarrer le serveur en cliquant sur le bouton « OK » et consulter la page d'aide de ce service en cliquant sur le bouton « Aide ».

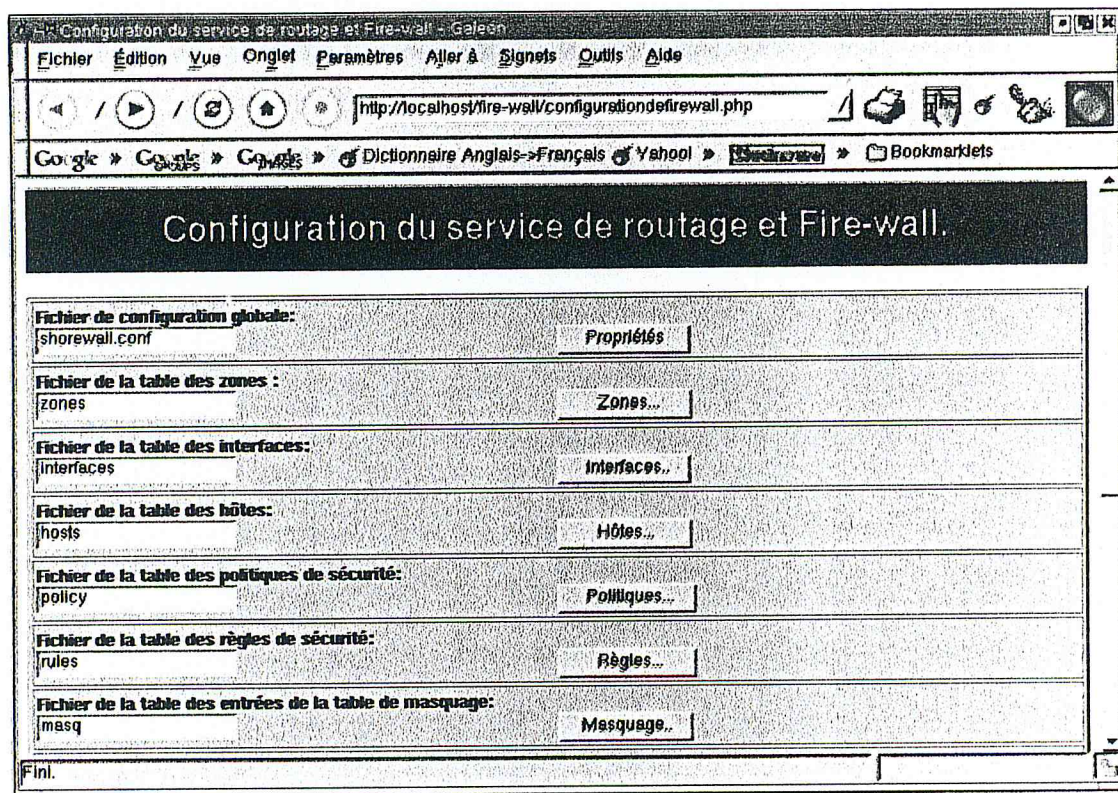


Figure-5.26 : l'interface du module de configuration du service de routage et Fire-wall.

Les figures 5.23, 5.24, 5.25, 5.26 et 5.27 montrent : la page des propriétés du serveur routeur/fire-wall, la page de configuration de la table de routage, la page de configuration des règles de sécurité, la page d'ajout d'une règle de sécurité et la page d'ajout d'une entrée ProxyArp.

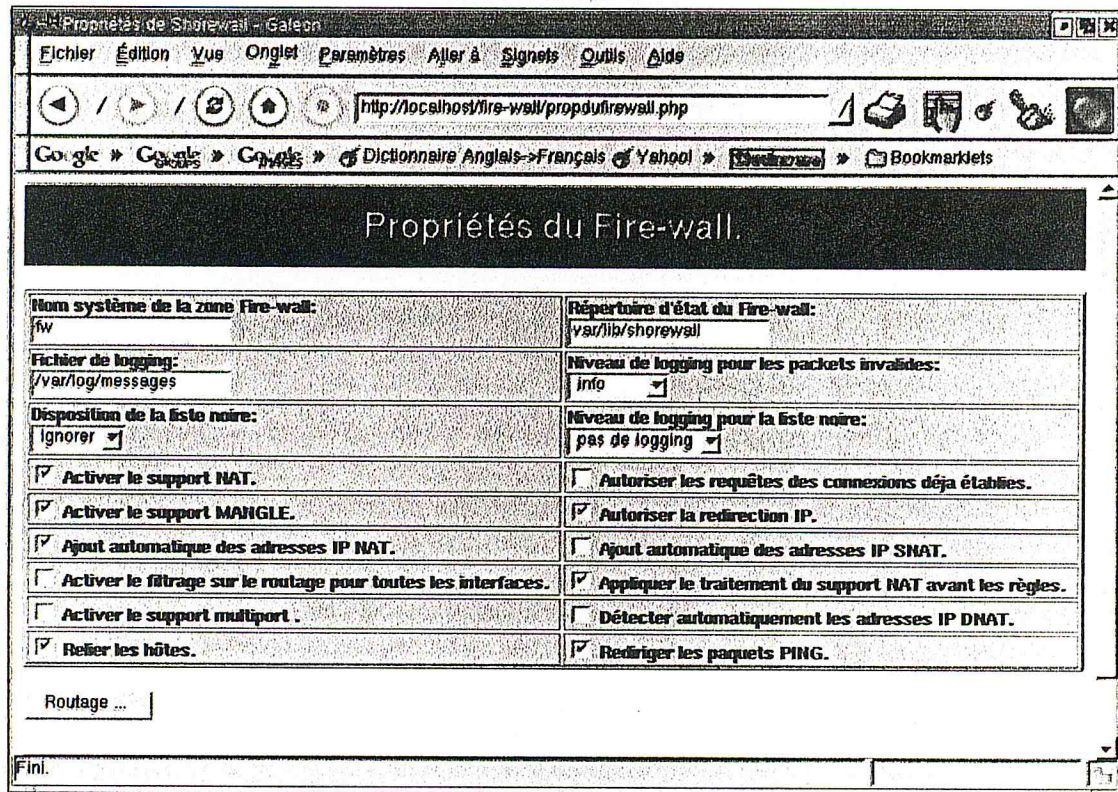


Figure-5.27 : La page des propriétés du serveur routeur/Fire-wall.

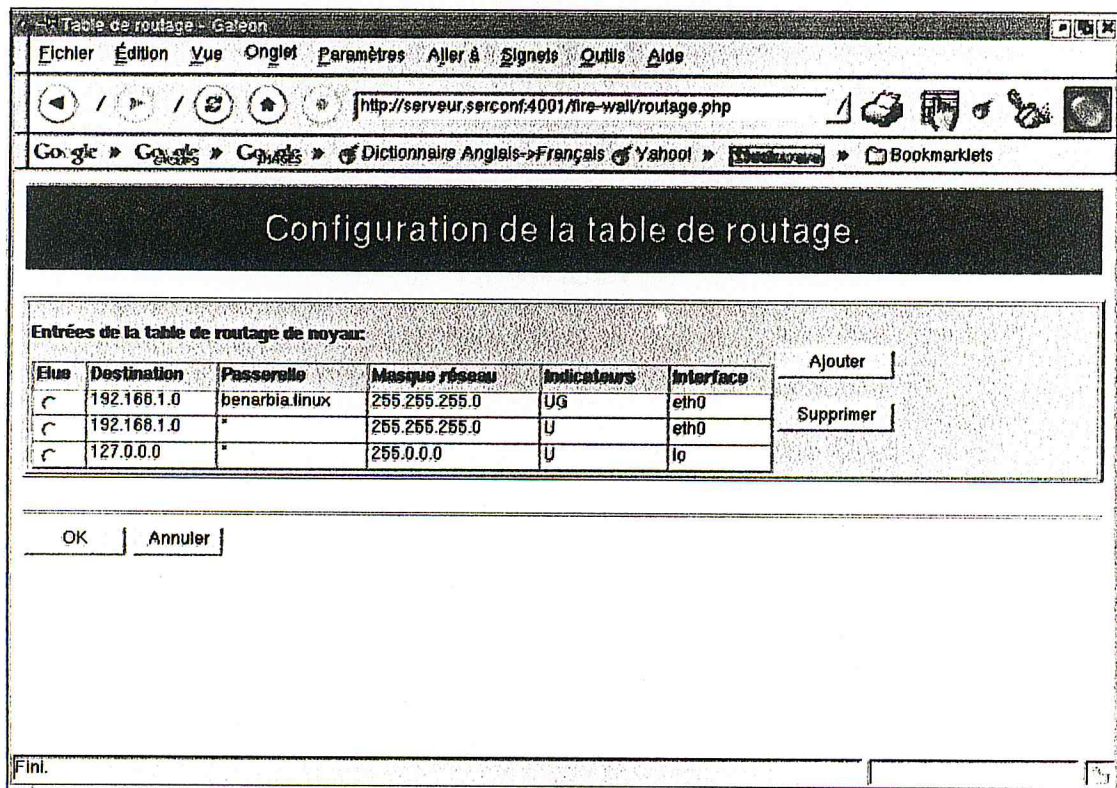


Figure-5.28 : La page de configuration de la table de routage.

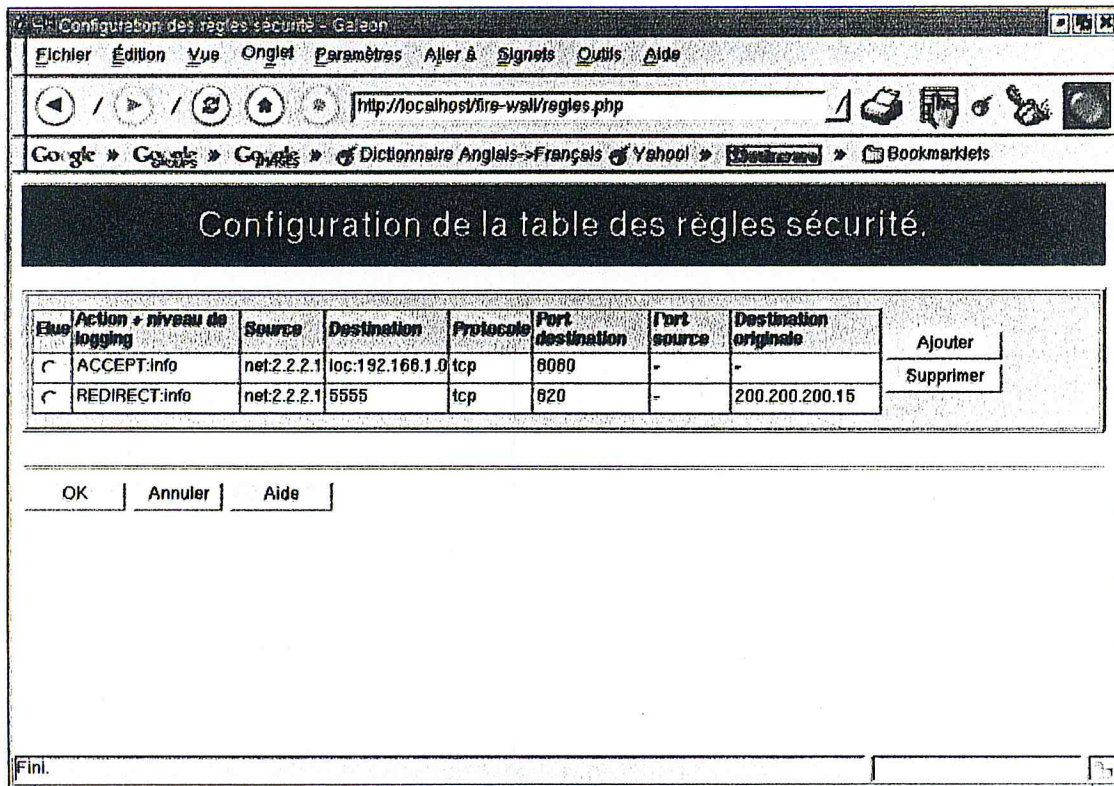


Figure-5.29: La page de configuration des règles de sécurité.

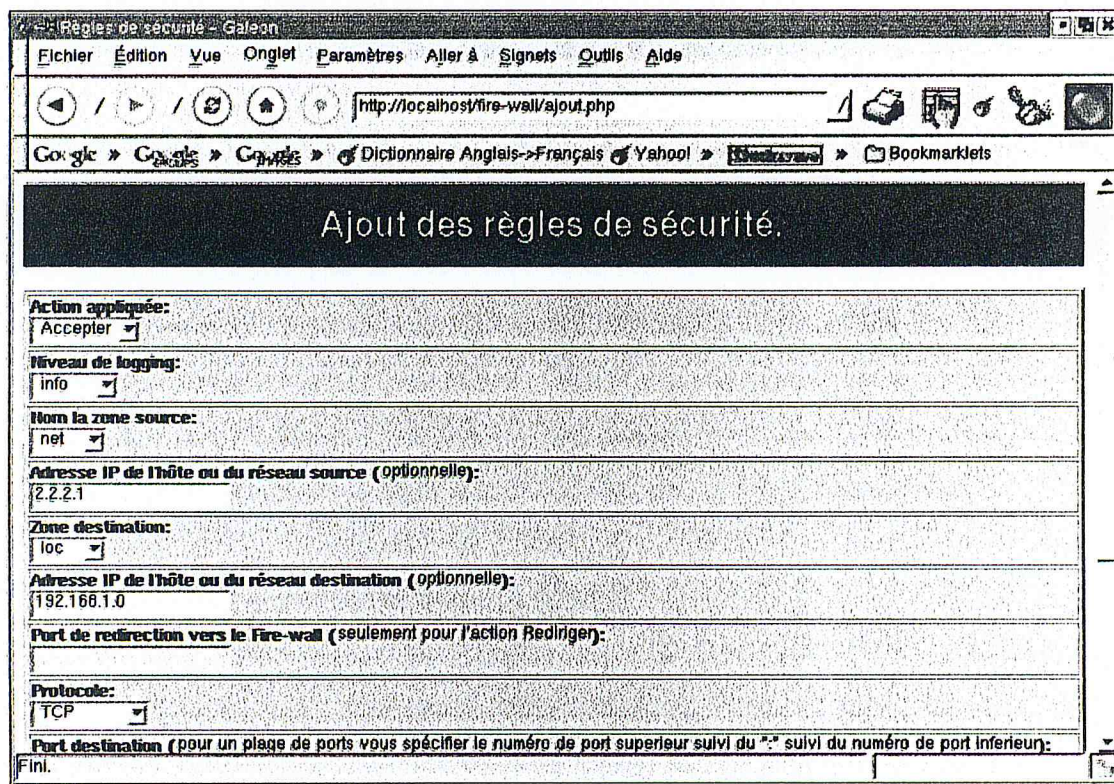


Figure-5.30 : La page d'ajout d'une règle de sécurité.

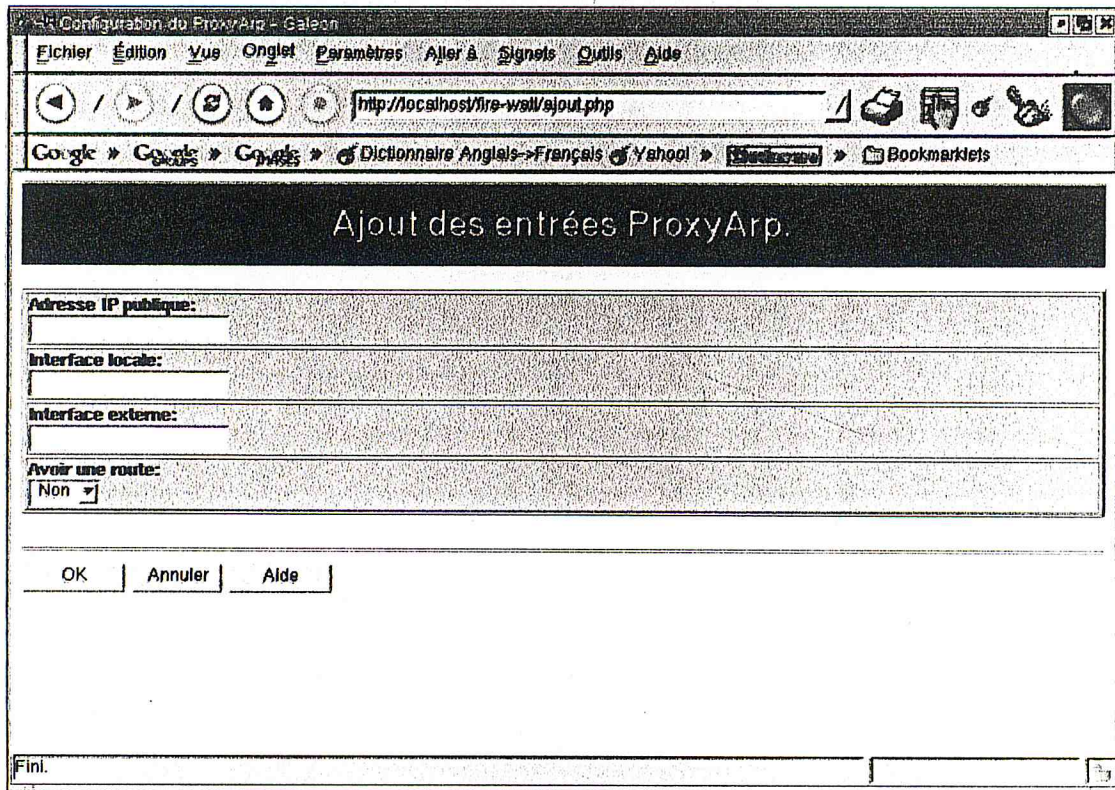


Figure-5.31 : La page d'ajout d'une entrée ProxyArp.

5.6- Conclusion

Après avoir défini, en premier lieu, l'environnement de développement matériel et logiciel, nous avons implémenté notre logiciel « SERCONF » qui présente à l'utilisateur plusieurs interfaces, nous avons vu certaines d'elles dans la section présentation de l'interface utilisateur.

Cette démarche nous a permis d'aboutir à un outil de configuration des services réseaux vus précédemment, facile à utiliser, muni d'une interface graphique conviviale et offre la possibilité de tester le bon fonctionnement ou la bonne configuration de ces services et un mécanisme d'aide pour l'utilisation de cet outil et la configuration de ces services réseaux.

Conclusion générale

Pendant l'étude faite pour la réalisation de notre projet de fin d'étude, nous avons constaté à quel point la configuration des services réseaux est importante pour leur bon fonctionnement dans les réseaux locaux (environnements *Intranet*) qui peuvent être connectés à Internet et aussi leur bon fonctionnement dans les machines reliées à Internet.

L'étude détaillée des services réseaux et leurs configurations nous a permis d'avoir une vision sur les besoins matériels et logiciels pour le fonctionnement des différentes applications informatiques qui tournent sous les réseaux **TCP**.

Notre travail consiste à concevoir et réaliser un outil de configuration des divers services réseaux sous Linux qui présente à l'utilisateur en plus de l'interface de configuration, une interface pour le test du bon fonctionnement ou de la bonne configuration de ces services.

Nous tenons à souligner que les services réseaux configurés par notre outil disposent de plusieurs options et paramètres de configuration dont les plus importants et les plus utilisés dans le quotidien des administrateurs réseaux sont pris en compte par notre outil.

La réalisation de notre logiciel dans une architecture qui s'articule autour d'un serveur web nous a permis aussi d'aborder des problèmes de programmation notamment avec un langage de script comme **PHP**, mais le plus important est que le développement de cette application a contribué à la familiarisation avec le système d'exploitation Linux et l'apprentissage d'un langage de script comme **PHP** qui s'avère très intéressant pour le développement des applications web puisque la plus part des sites web sur Internet sont réalisés à base de **PHP**.

Bibliographie

[Mic_99] : Linux Installation, Configuration et administration.

De Michael Kofler (CompuPresse France 1999).

[Tim_97] : Slackware Linux.

De Timathy Parker (Macmillan France 1997).

[Kar_99] : TCP/IP.

De Karanjit, S-Sigan (CompuPresse France 1997).

[www_1] : <http://www.commentcamarche.net/internet/tcpip.php3> .

[www_2] : <http://www.linux-france.org/article/memo/dns/node2.html> .

[www_3] : <http://www.africacomputing.org/pcours45.html> .

[www_4] : <http://www.nroullet.fr/dns.php> .

[www_5] : <http://www.commentcamarche.net/internet/http.php3> .

[www_6] : <http://www.africacomputing.org/pcours46.html> .

[www_7] : <http://www.commentcamarche.net/internet/ftp.php3> .

[www_8] : <http://www.proftpd.org>.

[www_9] : <http://www.commentcamarche.net/internet/smtp.php3> .

[www_10] : <http://cjovent.free.fr/cours/postfix.html> .

[www_11] : <http://www.liafa.jussieu.fr/~yunes/internet/nntp> .

[www_12] : <http://www.ensta.fr/~pollet/linux/inn.html> .

[www_13] : <http://www.commentcamarche.net/reslinux/reslsamba.php3> .

[www_14] : <http://www.chez.com/imer/linux/SMB-HOWTO-5.htm> .

[www_15] : <http://www.shorewall.net> .

[how_99] : Le HOW-TO du Pare-feu et des serveurs mandataires.

Traduit en français par Bernard Choppy en Décembre 1999,

(version original en anglais disponible à

<http://www.grenan.com/Fire-wall-HOWTO.html>).

Résumé

Le développement d'un système d'exploitation puissant comme Linux, une variante du système d'exploitation traditionnel UNIX muni d'une interface graphique, a beaucoup facilité l'utilisation de la plus part de ses applications par les utilisateurs non familiarisés avec UNIX. Cependant pour effectuer certains travaux comme la configuration des différents services réseaux par exemple, il est nécessaire d'utiliser des lignes de commandes pour manipuler des longs fichiers de configuration et d'avoir une connaissance sur les commandes de redémarrage et de teste.

Notre objectif est de concevoir un outil de configuration de divers services réseaux muni d'une interface graphique conviviale qui va permettre de plus de la configuration, la possibilité d'effectuer des testes de bon fonctionnement ou de bonne configuration de ces services et de le réaliser dans une architecture qui s'articule autour d'un serveur web.

Abstract

The development of a powerful operating system like Linux, a clone of the traditional operating system UNIX that have a graphical interface, had facilitate the use of its applications by the beginners that are not familiar with UNIX. However, for doing some works like the configuration of the different network services, it is necessary to use a command lines for dealing with a huge configuration files and have a knowledge of the different commands of restarting and testing of this services.

Our goal is to conceive a tool for the configuration of the different network services with a convivial graphical interface that give in addition of the configuration, the possibility of testing the good functioning or the good configuration of those services and realize it in an architecture that articulate around a web server.

