Table des matières

Section 1. Eléments de cours sur TCP/IP 12
1.1 Présentation de TCP/IP 12
1.2 OSI et TCP/IP 12
1.3 La suite de protocoles TCP / IP 13
1.3.1 IP (Internet Protocol, Protocole Internet) 13
1.3.2 TCP (Transmission Control Protocol, Protocole de contrôle de la transmission) 14
1.3.3 UDP (User Datagram Protocol) 15
1.3.4 ICMP (Internet Control Message Protocol) 15
1.3.5 RIP (Routing Information Protocol) 15
1.3.6 ARP (*Address Resolution* Protocol 15
1.3.7 Fonctionnement général 16
1.4 Les applications TCP-IP 16
1.4.1 Modèle client/serveur 16
1.4.2 L'adressage des applicatifs : les ports 18
1.4.3 Les ports prédéfinis à connaître 19
Section 2. Eléments de cours sur l'adressage IP 20
2.1 Adresses physiques (MAC) et adresses logiques (IP) 20
2.1.1 Notion d'adresse Physique et de trames 20
2.1.2 Notion d'adresse logique et de paquets 21
2.1.3 Résolution d'adresses logiques en adresses physiques 21
2.1.4 Attribution d'une adresse IP Internet 21
2.2 Adressage IP 22
2.2.1 Structure des adresses IP 22
2.2.2 Classes d'adresses 22
2.2.3 Identification du réseau 23
2.2.4 Adresses réservées 24
2.3 Les sous-réseaux 25
2.3.1 Pourquoi créer des sous réseaux ? 25
2.3.2Masque de sous-réseau 26
2.3.3 Sous-réseaux 26
2.3.4 Adressage de sur-réseaux 28

Section 3. Fichiers de configuration du réseau et 30
Présentation du document : les outils de l'administrateur réseau 30
3.1 Les fichiers de configuration 30
3.1.1 Le fichier `/etc/hosts` 30
3.1.2 Le fichier `/etc/networks` 30
3.1.3 Le fichier `/etc/host.conf` 31
3.1.4 Le fichier `/etc/resolv.conf` 31
3.1.5 Les fichiers de configuration des interfaces réseau 31
3.2 Les outils de l'administrateur réseau 31
3.2.1 La commande ifconfig 31
3.2.2 La commande arp 34
3.2.3 La commande route 36
3.2.4 La commande netstat 38
3.3.5La commande traceroute 39
3.2.6 La commande dig 40
3.2.7 La commande host 41
Section 4. Eléments de cours sur ARP 44
4.1 Le protocole ARP 44
4.2 Processus de recherche de l'adresse physique 45
Travaux Pratiques : La résolution d'adresses physiques 47
Section 5. Le routage 51
5.1 Principe 51
5.2 Acheminement des paquets TCP-IP 52
5.3 Les tables de routage 53
5.4 Acheminement Internet 55
5.4.1 Domaine d'acheminement 55
5.4.2 Principe du choix d'une voie d'acheminement 55
5.5 Routage dynamique 56
Section 6. Installation d'un serveur NFS 57
6.1 Résumé 57
6.2 Installation des produits clients et serveurs 58
6.3 Lancement et arrêt de NFS 59
6.4 Configuration Client / Serveur 60

6.4.1 Les fichiers de configuration du serveur NFS 60
6.4.2 La commande `exportfs` 63
6.4.3 Utilisation de la commande exportfs avec NFSv4 64
6.4.4 Autres commandes d'administration 65
6.4.5 L'identité des utilisateurs 65
6.4.6 Configuration et utilisation du client Unix/Linux 66
6.5 Ressources supplémentaires 71
6.5.1 Documentation installée 71
6.5.2 Sites Web utiles 72
6.5.3 Livres sur le sujet 72
Travaux pratiques : partages NFS 73
Première partie 73
Deuxième partie 74
Troisième partie 74
Section 7. Le service SAMBA 76
7.1 Introduction 76
7.2 Eléments d'installation et de configuration de SAMBA 76
7.3 Le fichier de configuration sous Linux 77
7.4 Les étapes de la configuration du serveur 77
7.5.1 Première étape - Configuration du fichier smb.conf 77
7.5.2 Deuxième étape - Déclarer les ressources partagées 78
7.6 Fichier de configuration d'un serveur SAMBA : 78
7.7 Création d'utilisateurs Samba 79
7.8 Accès depuis un poste client Linux 80
7.9 Accès depuis un poste client Windows 82
7.10 Serveur Samba en tant que contrôleur de domaine 84
Travaux pratiques : installation d'un serveur SAMBA 88
Déroulement des opérations 88
Configuration du fichier smb.conf et démarrage des services 88
Création de comptes utilisateurs 89
Vérification de la configuration sur le serveur SAMBA 89
Procédure de test à partir d'un client Linux 89
Procédure de test à partir d'un client Windows 90

Configuration d'un contrôleur de domaine. 91
Automatisation de création de comptes. 91
Administration graphique 92
Section 8. FTP 93
8.1. Protocole FTP (File Transport Protocol) 93
8.1.1. Ports multiples, Modes multiples 93
8.2. Serveurs FTP 94
8.2.1. vsftpd 94
8.3. Fichiers installés avec `vsftpd` 95
8.4. Démarrage et arrêt de `vsftpd` 96
8.4.1. Démarrage de multiples copies de `vsftpd` 97
8.5. Options de configuration de `vsftpd` 98
8.5.1. Options pour le démon 98
8.5.2. Options de connexion et contrôles d'accès 99
8.5.3. Options pour les utilisateurs anonymes 101
8.5.4. Options pour les utilisateurs locaux 102
8.5.5. Options pour les répertoires 103
8.5.6. Options pour le transfert de fichiers 104
8.5.7. Options de journalisation 105
8.5.8. Options réseau 106
8.6. Ressources supplémentaires 109
8.6.1. Documentation installée 109
8.6.2. Sites Web utiles 110
Section 9. Eléments de cours sur le service DHCP 111
9.1 Résumé 111
9.2 Rôle d'un service DHCP 111
9.2.1 Pourquoi mettre en place un réseau TCP/IP avec des adresses IP dynamiques 112
9.2.2 Protocole DHCP(Dynamic Host Configuration Protocol) 113
9.3 Fonctionnement de DHCP 113
9.3.1 Attribution d'une adresse DHCP 113
9.3.2 Renouvellement de bail IP 114
9.4 Configuration d'un serveur DHCP 114
9.5 Mise en oeuvre d'un client DHCP 114

9.6 Rôle de l'agent de relais DHCP 116
Travaux pratiques : installation d'un serveur DHCP 118
Réalisation du TP 122
Travaux pratiques : installation d'un agent relais DHCP 123
Routeur et agent relais DHCP (RFC 1542) 123
La maquette 124
Installation 124
Section 10 Installation d'un serveur DNS 128
10.1 Description et objectifs de la séquence 129
10.2 Qu'est ce que le service de résolution de noms de domaine 129
10.3 Présentation des concepts 130
10.3.1 Notion de domaine, de zone et de délégation 130
10.3.2 Le domaine in-addr.arpa 133
10.3.3 Fichiers, structure et contenus 133
10.3.4 La délégation 134
10.3.5 Serveur primaire et serveur secondaire 134
10.3.6 Le cache 134
10.4 Types de serveurs de noms 134
10.5 BIND en tant que serveur de noms 135
10.5.1 /etc/named.conf 135
10.5.2 Types courants de déclarations 136
10.5.3 Autres types de déclarations 141
10.5.4 Balises de commentaire 143
10.6 Fichiers de zone 143
10.6.1 Directives des fichiers de zone 143
10.6.2 Enregistrements de ressources des fichiers de zone 144
10.6.3 Exemples de fichiers de zone 148
10.6.4 Fichiers de résolution de noms inverse 148
10.7 Utilisation de rndc 149
10.7.1 Configuration de `/etc/named.conf` 150
10.7.2 Configuration de `/etc/rndc.conf` 150
10.7.3 Options de ligne de commande 151
10.8 Fonctionnalités avancées de BIND 152

10.8.1 Améliorations du protocole DNS 152
10.8.2 Vues multiples 153
10.8.3 Sécurité 153
10.8.4 IP version 6 153
10.9 Erreurs courantes à éviter 154
10.10 Ressources supplémentaires 154
10.10.1 Documentation installée 154
10.10.2 Sites Web utiles 156
10.10.3 Livres sur le sujet 156
Section 11 Installation d'un serveur DDNS avec bind et DHCP 157
11.1 Résumé 157
11.2 Eléments sur le service DDNS 158
11.3 Les aspects sur la sécurité 159
Travaux pratiques : DDNS 161
Réalisation 161
Procédure de tests des services 164
Intégration des services 165
Générer un nom dynamiquement pour les clients DHCP 167
Section 12 Serveur HTTP Apache 169
12.1 Serveur HTTP Apache 2.0 169
12.1.1 Fonctions du Serveur HTTP Apache 2.0 169
12.1.2 Changements au niveau des paquetages dans le Serveur HTTP Apache 2.0 170
12.2 Changements apportés au système de fichiers de la version 2.0 du Serveur HTTP Apache 170
12.2.1 Configuration de l'environnement global 171
12.2.2 Configuration du serveur principal 173
12.2.3 Modules et Serveur HTTP Apache 2.0 176
12.3 Après l'installation 182
12.4 Démarrage et arrêt de `httpd` 182
12.5 Directives de configuration dans `httpd.conf` 183
12.5.1 Astuces générales de configuration 184
12.5.2 `ServerRoot` 184
12.5.3 `PidFile` 184

12.5.4 Timeout 184
12.5.5 KeepAlive 184
12.5.6 MaxKeepAliveRequests 185
12.5.7 KeepAliveTimeout 185
12.5.8 IfModule 185
12.5.9 Directives de server-pool spécifiques aux MPM 185
12.5.10 Listen 187
12.5.11 Include 187
12.5.12 LoadModule 187
12.5.13 ExtendedStatus 187
12.5.14 IfDefine 188
12.5.15 SuexecUserGroup 188
12.5.16 User 188
12.5.17 Group 188
12.5.18 ServerAdmin 189
12.5.19 ServerName 189
12.5.20 UseCanonicalName 189
12.5.21 DocumentRoot 189
12.5.22 Directory 190
12.5.23 Options 190
12.5.24 AllowOverride 191
12.5.25 Order 191
12.5.26 Allow 191
12.5.27 Deny 191
12.5.28 UserDir 191
12.5.29 DirectoryIndex 192
12.5.30 AccessFileName 192
12.5.31 CacheNegotiatedDocs 192
12.5.32 TypesConfig 192
12.5.33 DefaultType 193
12.5.34 HostnameLookups 193
12.5.35 ErrorLog 193

12.5.36 LogLevel 193
12.5.37 LogFormat 193
12.5.38 CustomLog 194
12.5.39 ServerSignature 195
12.5.40. Alias 195
12.5.41. ScriptAlias 195
12.5.42. Redirect 195
12.5.43. IndexOptions 196
12.5.44. AddIconByEncoding 196
12.5.45. AddIconByType 196
12.5.46. AddIcon 197
12.5.47. DefaultIcon 197
12.5.48. AddDescription 197
12.5.49. ReadmeName 197
12.5.50. HeaderName 197
12.5.51. IndexIgnore 197
12.5.52. AddEncoding 197
12.5.53. AddLanguage 198
12.5.54. LanguagePriority 198
12.5.55. AddType 198
12.5.56. AddHandler 198
12.5.57. Action 198
12.5.58. ErrorDocument 198
12.5.59. BrowserMatch 199
12.5.60. Location 199
12.5.61. ProxyRequests 199
12.5.62. Proxy 200
12.5.63. Directives cache 200
12.5.64. NameVirtualHost 200
12.5.65. VirtualHost 201
12.5.66. Directives de configuration SSL 201
12.6 Ajout de modules 202

12.7. Hôtes virtuels 202
12.7.1 Configuration d'hôtes virtuels 203
12.7.2 Hôte virtuel du serveur Web sécurisé 204
12.8 Ressources supplémentaires 204
12.8.1 Sites Web utiles 205
12.8.2 Livres sur le sujet 205
TP 1 installation d'un serveur HTTP 206
Résumé 206
TP 2 : Création de pages Web 209
Résumé 209
Vérification de la configuration 209
Installation d'un site Web 209
Développement d'un site 210
Test de vos pages 210
Utilisation des alias 211
Auto évaluation sur le deuxième TP 211
TP 3 : Configuration des répertoires personnels 212
Configurer le compte personnel 212
Développer un site personnel 212
Tester l'accès au site personnel 213
Auto-évaluation sur le troisième TP 213
TP 4 : Mise en place d'un accès sécurisé 214
Déployer un site d'accès en ligne 214
Sécuriser l'accès à ce site par un mot de passe 214
Tester la configuration. 215
Les fichiers .htaccess 215
Auto-évaluation sur le quatrième TP 215
TP 5 : Serveurs webs virtuels et redirection 217
Avant de commencer sur les serveurs web virtuels 219
Serveur web virtuel basé sur les adresses ip 219
Serveur Web virtuel basé sur le nom 220
Application sur la redirection 221
Annexe pour le "web-hosting" 222

Chapitre 13. Courrier électronique 223
13.1. Protocoles de courrier électronique 223
13.1.1. Protocoles de transfert de courrier électronique 223
13.1.2. Protocoles d'accès au courrier 224
13.2. Classifications des programmes de messagerie électronique 226
13.2.1. Agent de transfert de courrier (ATC) 226
13.2.2. Agent de distribution du courrier (ADC) 226
13.2.3. Agent de gestion de courrier (AGC) 227
13.3. Agent de transfert de courrier (ATC) 227
13.3.1. Sendmail 227
13.3.2. Postfix 232
13.3.3. Fetchmail 234
13.4. Agent de distribution de courrier (ADC) 238
13.4.1. Configuration de Procmail 239
13.4.2. Recettes Procmail 241
13.5. Ressources supplémentaires 246
13.5.1. Documentation installée 246
13.5.2. Sites Web utiles 247
13.5.3. Livres sur le sujet 248
Section 14 Installation d'un service Web-mail 249
14.1 Présentation 249
14.2 Architecture générale du service 249
14.3 Installation et configuration OpenWebmail 250
14.3.1 Préparation de la machine 250
14.3.2 Installation d'OpenWebmail 253
14.3.3 Configuration de l'application OpenWebmail 253
14.3.4 Test de l'environnement 253
14.3.5 Configuration de l'environnement utilisateur 254
14.3.6 Test et environnement OpenWebmail 255
14.4 Application 256
Section 15 Protocole LDAP (Lightweight Directory Access Protocol) 257
15.1. Pourquoi utiliser LDAP ? 257
15.1.1. Caractéristiques d'OpenLDAP 258

15.2. Terminologie de LDAP 258
15.2.1 Le protocole 259
15.2.2 Le modèle de données 260
15.3. Démons et utilitaires d'OpenLDAP 265
15.3.1. Filtre de recherche 267
15.3.2. NSS, PAM et LDAP 269
15.3.3. PHP4, LDAP et le Serveur HTTP Apache 270
15.3.4. Applications client LDAP 271
15.4. Fichiers de configuration d'OpenLDAP 271
15.5. Répertoire `/etc/openldap/schema/` 271
15.6. Aperçu de la configuration d'OpenLDAP 272
15.6.1. Édition de `/etc/openldap/slapd.conf` 273
15.7. Configuration d'un système pour l'authentification avec OpenLDAP 274
15.7.1. PAM et LDAP 275
15.7.2. Migration de vos anciennes informations d'authentification vers le format LDAP 275
15.8. Ressources supplémentaires 276
15.8.1. Documentation installée 277
15.8.2. Sites Web utiles 278
15.8.3. Livres sur le sujet 278
TP Installation/Configuration d'un annuaire LDAP sur serveur GNU/Linux 279

Section 1. Eléments de cours sur TCP/IP

Présentation de TCP/IP
OSI et TCP/IP
La suite de protocoles TCP / IP
- IP **(*Internet Protocol*, Protocole Internet)**
- TCP **(Transmission Control Protocol, Protocole de contrôle de la transmission)**
- UDP **(*User Datagram Protocol*)**
- ICMP **(*Internet Control Message Protocol*)**
- RIP **(*Routing Information Protocol*)**
- ARP **(*Address Resolution Protocol***
- Fonctionnement général

Les applications TCP-IP
- Modèle client/serveur
- L'adressage des applicatifs : les ports
- Les ports prédéfinis à connaître

Abstract
Le document présente la suite de protocoles TCP/IP.
Ce document sert d'introduction à l'ensemble des cours et TP sur les différents protocoles

1.1 Présentation de TCP/IP

TCP/IP est l'abréviation de Transmission Control Protocol/Internet Protocol. Ce protocole a été développé, en environnement UNIX, à la fin des années 1970 à l'issue d'un projet de recherche sur les interconnexions de réseaux mené par la DARPA (Defense Advanced Research Projects Agency) dépendant du DoD (Department of Defense) Américain.
TCP/IP ,devenu standard de fait, **est actuellement la famille de protocoles réseaux qui gère le routage la plus répandue** sur les systèmes informatiques (Unix/Linux, Windows, Netware...) et surtout, c'est le protocole de l'Internet.
Plusieurs facteurs ont contribué à sa popularité :
Maturité, Ouverture, Absence de propriétaire, Richesse (il fournit un vaste ensemble de fonctionnalités), Compatibilité (différents systèmes d'exploitation et différentes architectures matérielles), et le développement important d'Internet.
La famille des protocoles TCP/IP est appelée **protocoles Internet**, et a donné son nom au réseau du même nom. Leurs spécifications sont définies dans des documents du domaine public appelés RFC (Request For Comments - Appels à commentaires). Ils sont produits par l'IETF (Internet Engineering Task Force) au sein de l'IAB (Internet Architecture Board).
La RFC 826, par exemple, définit le protocole ARP.

1.2 OSI et TCP/IP

Bien que le protocole TCP/IP ait été développé bien avant que le modèle OSI apparaisse, ils ne sont pas totalement incompatibles. L'architecture OSI est définie plus rigoureusement, mais ils disposent tous deux d'une architecture en couches.
Les protocoles **TCP** et **IP** ne sont que deux des membres de la suite de protocoles TCP/IP qui constituent le **modèle DOD** (modèle en 4 couches). Chaque couche du modèle TCP/IP correspond à une ou plusieurs couches du modèle OSI (*Open Systems Interconnection*) défini par l'ISO (*International Standards Organization*) :

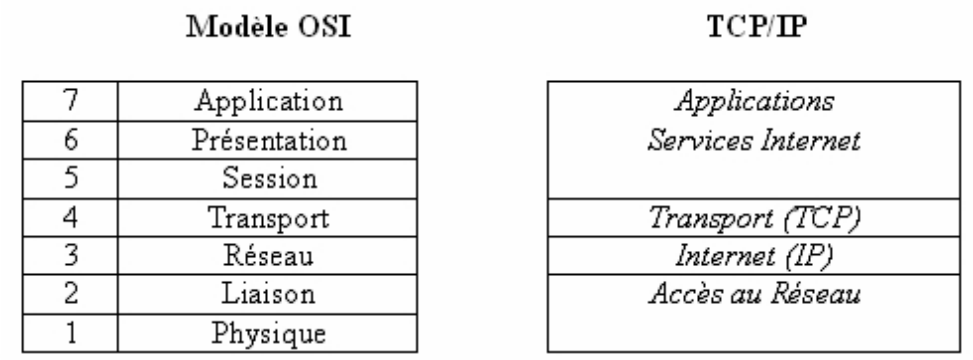


Figure 1.1. OSI et TCP/IP

Des relations étroites peuvent être établies entre la couche réseau et IP, et la couche transport et TCP.
TCP/IP peut utiliser une grande variété de protocoles en couche de niveau inférieur, notamment X.25, Ethernet et Token Ring. En fait, TCP/IP a été explicitement conçu sans spécification de couche physique ou de liaison de données car le but était de faire un protocole adaptable à la plupart des supports.

1.3 La suite de protocoles TCP / IP

Les protocoles TCP/IP se situent dans un modèle souvent nommé "famille de protocoles TCP/IP".
Les protocoles **TCP** et **IP** ne sont que deux des membres de la suite de protocoles IP.

1.3.1 IP (Internet Protocol, Protocole Internet)

*IP est un protocole qui se **charge de l'acheminement des paquets** pour tous les autres protocoles de la famille TCP/IP.* Il fournit un système de remise de données optimisé sans connexion. Le terme « optimisé » souligne le fait qu'il ne garantit pas que les paquets transportés parviennent à leur destination, ni qu'ils soient reçus dans leur ordre d'envoi. La fonctionnalité de somme de contrôle du protocole ne confirme que l'intégrité de l'en-tête IP. Ainsi, seuls les protocoles de niveau supérieur sont responsables des données contenues dans les paquets IP (et de leur ordre de réception).
Le protocole IP travaille en **mode non connecté**, c'est-à-dire que les paquets émis par le niveau 3 sont **acheminés de manière autonome** (datagrammes), sans garantie de livraison.
Le **datagramme** correspond au format de paquet défini par le protocole Internet. Les cinq ou six (sixième facultatif) premier mots de 32 bits représentent les informations de contrôle appelées en-tête.

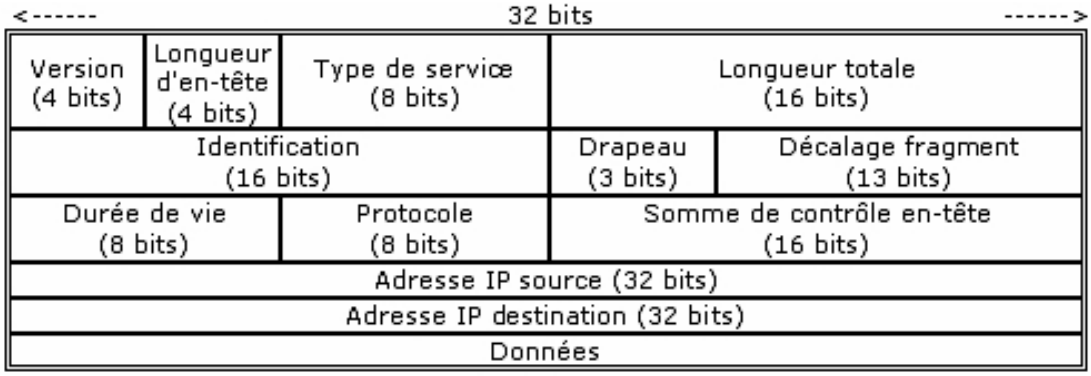


Figure 1.2. Datagramme IP

La longueur théorique maximale d'un datagramme IP est de 65535 octets. En pratique la taille maximale du datagramme est limitée par la longueur maximale des trames transportées sur le réseau physique. La fragmentation du datagramme (définie dans le 2ème mot de 32 bits) devient alors nécessaire dès que sa taille ne lui permet plus d'être directement transporté dans une seule trame physique. Les modules internet des équipements prennent en charge le découpage et le réassemblage des datagrammes.

Le protocole Internet transmet le datagramme en utilisant l'adresse de destination contenue dans le cinquième mot de l'en-tête. L'adresse de destination est une adresse IP standard de 32 bits permettant d'identifier le réseau de destination et la machine hôte connectée à ce réseau.

Dans un réseau TCP/IP, on assigne généralement une adresse IP à chaque hôte. Le terme d'hôte est pris dans son sens large, c'est à dire un "noeud de réseau". Une imprimante, un routeur, un serveur, un poste de travail sont des noeuds qui peuvent avoir également un nom d'hôte, s'ils ont une adresse IP.

1.3.2 TCP (Transmission Control Protocol, Protocole de contrôle de la transmission)

TCP est probablement le protocole IP de niveau supérieur le plus répandu. ***TCP fournit un service sécurisé de remise des paquets.*** *TCP fournit un protocole fiable, orienté connexion,* ***au-dessus d'IP*** *(ou encapsulé à l'intérieur d'IP).* TCP garantit l'ordre et la remise des paquets, il vérifie l'intégrité de l'en-tête des paquets et des données qu'ils contiennent. TCP est responsable de la retransmission des paquets altérés ou perdus par le réseau lors de leur transmission. Cette fiabilité fait de TCP/IP un protocole bien adapté pour la transmission de données basée sur la session, les applications client-serveur et les services critiques tels que le courrier électronique.

La fiabilité de TCP a son prix. Les en-têtes TCP requièrent l'utilisation de bits supplémentaires pour effectuer correctement la mise en séquence des informations, ainsi qu'un total de contrôle obligatoire pour assurer la fiabilité non seulement de l'en-tête TCP, mais aussi des données contenues dans le paquet. Pour garantir la réussite de la livraison des données, ce protocole exige également que **le destinataire accuse réception des données**.

Ces accusés de réception (ACK) génèrent une activité réseau supplémentaire qui diminue le débit de la transmission des données au profit de la fiabilité. Pour limiter l'impact de cette contrainte sur la performance, la plupart des hôtes n'envoient un accusé de réception que pour un segment sur deux ou lorsque le délai imparti pour un ACK expire.

Sur une connexion TCP entre deux machines du réseau, les messages (ou paquets TCP) sont acquittés et délivrés en séquence.

1.3.3 UDP (User Datagram Protocol)

`UDP` est un complément du protocole TCP qui offre un **service de datagrammes sans connexion** qui ne garantit ni la remise ni l'ordre des paquets délivrés. Les sommes de contrôle des données sont facultatives dans le protocole UDP. Ceci permet d'échanger des données sur des réseaux à fiabilité élevée sans utiliser inutilement des ressources réseau ou du temps de traitement. Les messages (ou paquets UDP) sont transmis de manière autonome (sans garantie de livraison.).
Le protocole UDP prend également en charge l'envoi de données d'un unique expéditeur vers plusieurs destinataires.
Ex: TFTP(trivial FTP) s'appuie sur UDP, DHCP également, Windows utilise UDP pour les Broadcast en TCP-IP

1.3.4 ICMP (Internet Control Message Protocol)

`ICMP` est un **protocole de maintenance** utilisé pour les tests et les diagnostics, qui véhicule des messages de contrôle. Il permet à deux systèmes d'un réseau IP de partager des informations d'état et d'erreur.
La commande **ping** utilise les paquets ICMP de demande d'écho et de réponse à un écho afin de déterminer si un système IP donné d'un réseau fonctionne. C'est pourquoi l'utilitaire **ping** est utilisé pour diagnostiquer les défaillances au niveau d'un réseau IP ou des routeurs.

1.3.5 RIP (Routing Information Protocol)

RIP est un protocole de routage dynamique qui permet l'échange d'informations de routage sur un inter-réseau. Chaque routeur fonctionnant avec RIP échange les identificateurs des réseaux qu'il peut atteindre, ainsi que la distance qui le sépare de ce réseau *(nb de sauts=nb de routeurs à traverser).* Ainsi chacun dispose de la liste des réseaux et peut proposer le meilleur chemin.

1.3.6 ARP (*Address Resolution* Protocol

Le protocole ARP permet de déterminer l'adresse physique (ou MAC) d'un noeud à partir de son adresse IP en effectuant une diffusion du type "*qui est X2.X2.X2.X2 ? "*

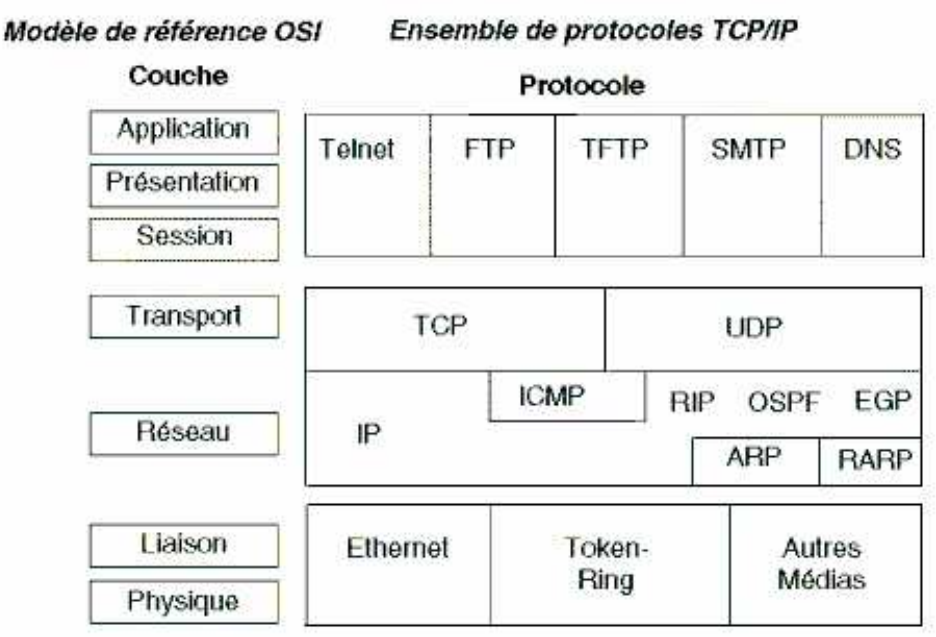


Figure 1.3. Protocoles TCP/IP et OSI

1.3.7 Fonctionnement général

Pour désigner les informations transmises et leur enveloppe, selon le niveau concerné, on parle de ***message****(ou de flux) entre applications, de* ***datagramme*** *(ou segment) au niveau TCP, de* ***paquet*** *au niveau IP, et enfin, de* ***trames*** *au niveau de l'interface réseau (Ethernet ou Token Ring).*

Les protocoles du niveau application les plus connus sont :

- **HTTP** (Hyper Text Transfer Protocol) permet l'accès aux documents HTML et le transfert de fichiers depuis un site WWW
- **FTP** (File Transfer Protocol) pour le transfert de fichiers s'appuie sur TCP et établit une connexion sur un serveur FTP
- **Telnet** pour la connexion à distance en émulation terminal, à un hôte Unix/Linux.
- **SMTP** (Simple Mail Transfer Protocol) pour la messagerie électronique (UDP et TCP)
- **SNMP** (Simple Network Management Protocol) pour l'administration du réseau
- **NFS** (Network File System) pour le partage des fichiers Unix/Linux.

1.4 Les applications TCP-IP

1.4.1 Modèle client/serveur

Les applications réseaux fonctionnent sur le **modèle client/serveur**. Sur la machine serveur un **processus serveur** (daemon) traite les **requêtes des clients**. Client et serveur dialoguent en échangeant des messages qui contiennent des requêtes et des réponses.

Prenons par exemple telnet.

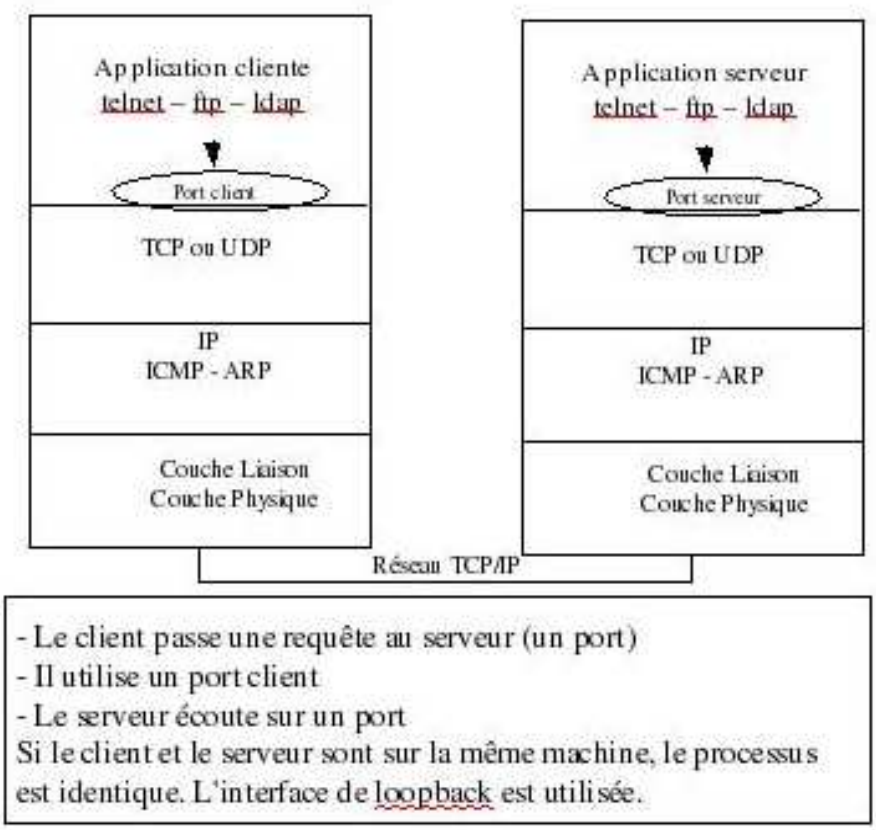


Figure 1.4. Exemple Telnet

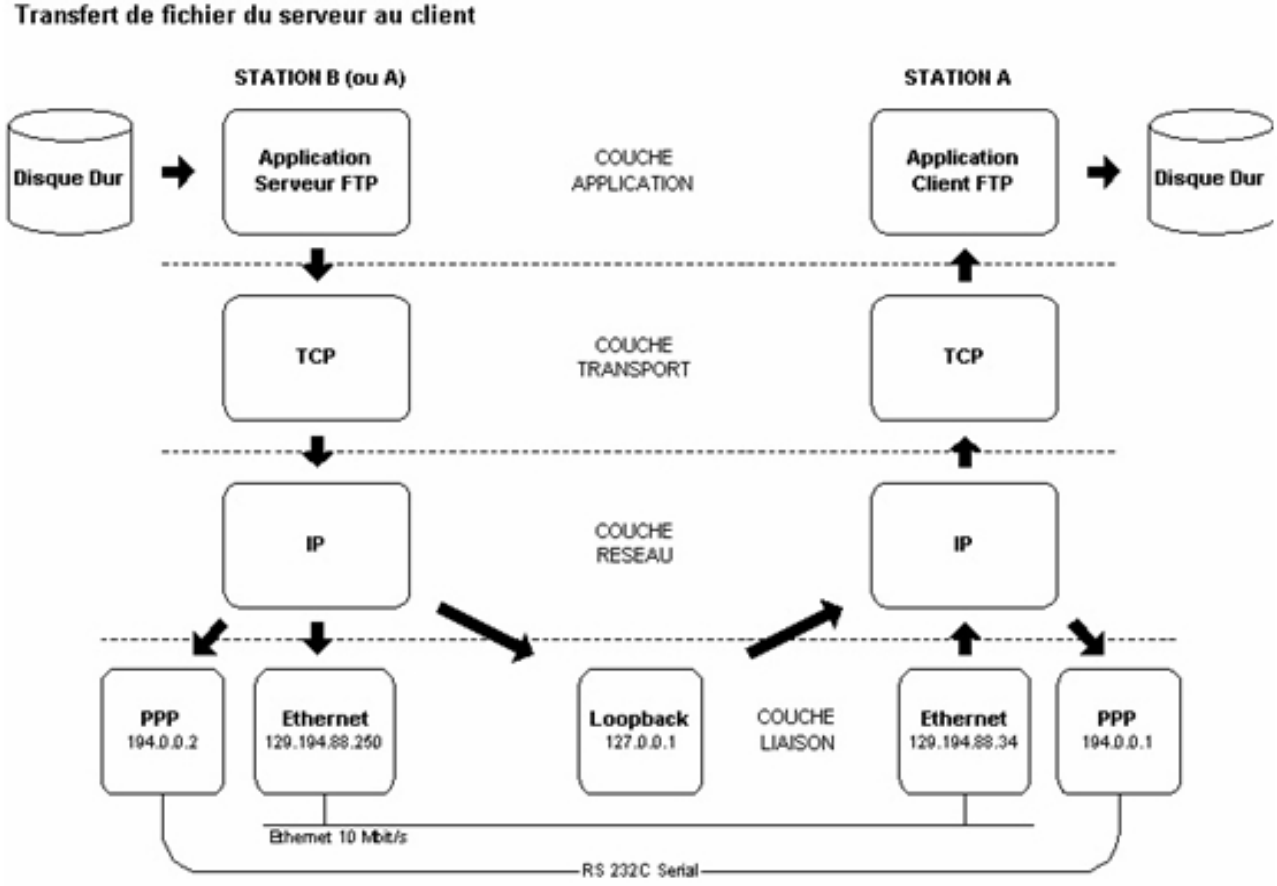


Figure 1.5. Modèle client/serveur

1.4.2 L'adressage des applicatifs : les ports

Une fois le datagramme transmis à l'hôte destinataire, il doit parvenir à l'utilisateur (si le système est multi-utilisateur) et à l'application visée (si le système est multi-tâches).

- sur la machine cliente, l'utilisateur (usager ou programme) effectue une requête vers une machine IP serveur sur le réseau. (par exemple *telnet host* ou *ftp host*). Cela se traduit par la réservation d'un port de sortie TCP ou UDP et l'envoi d'un paquet IP à la machine serveur. Ce paquet contient un message TCP ou UDP avec un numéro de port correspondant à l'application demandée sur le serveur.
- sur le serveur, la requête est réceptionnée par le pilote IP, aiguillée vers TCP ou UDP puis vers le port demandé. Le processus serveur correspondant est à l'écoute des appels sur ce port (par ex: le daemon *telnetd* traite les requêtes *telnet*, le daemon *ftpd* traite les requêtes *ftp*).
- processus client et processus serveur échangent ensuite des messages.

Des numéros de port (entre 0 et 1023) sont réservés pour les applications « standards : les ports « bien connus » (Well Known Ports), ils ont été assignés par l'IANA. Sur la plupart des systèmes ils peuvent être seulement employés par des processus du système (ou root) ou par des programmes exécutés par les utilisateurs privilégiés (liste complète : http://www.iana.org/assignments/port-numbers ou dans le fichier **/etc/services** y compris sous Windows).

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

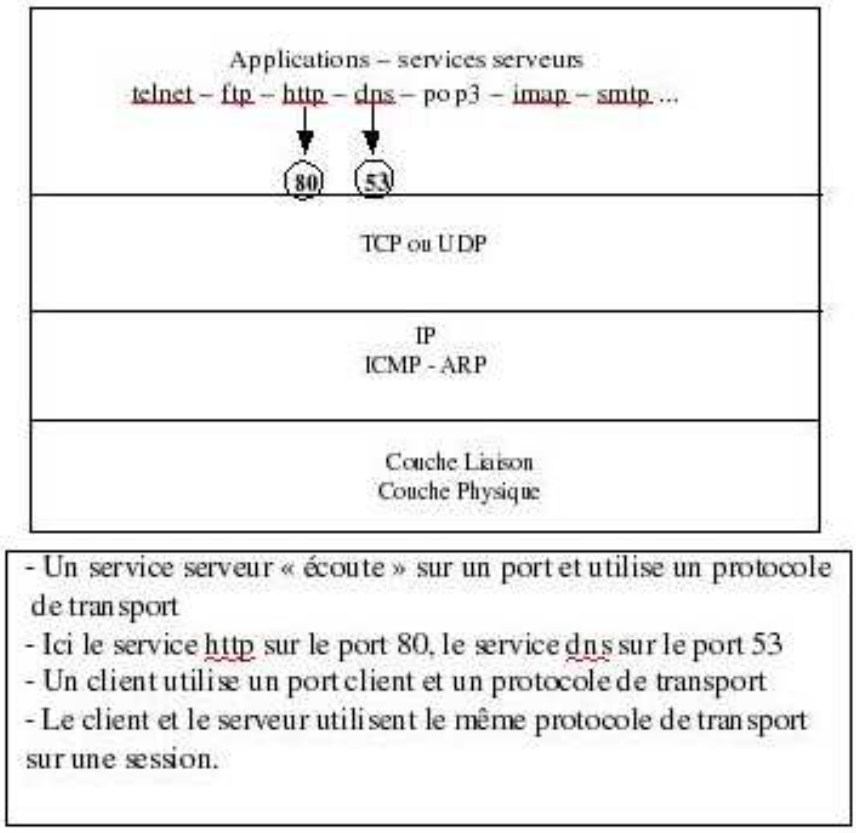


Figure 1.6. Ports applicatifs

On identifie le protocole de communication entre applications par un numéro de protocole etl'application par un numéro de port.
Par exemple, les serveurs HTTP dialoguent de manière traditionnelle par le port 80 :
http ://www.sncf.com/index.htm <=> http :// www.sncf.com:80/index.htm
Les numéros de protocole et de port sont inclus dans le datagramme.
Une fois la connexion établie entre le client et le serveur, ceux-ci peuvent s'échanger des informations selon un protocole défini selon l'applicatif. Le client soumet des requêtes auxquelles répondra le serveur.
Ce mode de communication s'appuie sur la couche "socket". Cette couche est une interface entre la couche présentation et transport. Elle permet la mise en place du canal de communication entre le client et le serveur. On peut schématiquement dire qu'un socket fournit un ensemble de fonctions. Ces fonctions permettent à une application client/serveur d'établir un canal de communication entre 2 ou plusieurs machines, qui utilisent un protocole de transport (TCP ou UDP) et un port de communication.

1.4.3 Les ports prédéfinis à connaître

Service réseau	*N° de Port*	*Type*	*Commentaire*
ICMP	7	TCP/UDP	Commandes Ping
Netstat	15	TCP/UDP	Etat du réseau
FTP	21	TCP	Transfert de fichiers
SSH	22	TCP/UDP	SSH Remote Login Protocol
Telnet	23	TCP	Connexion via terminal réseau
SMTP	25	TCP	Envoi de courrier
DNS	53	TCP/UDP	Serveurs de noms de domaine
HTTP	80	TCP	Serveur Web
Pop3	110	TCP	Réception de courrier
nntp	119	TCP	Service de news
ntp	123	UDP	Protocole temps réseau
nbname	137	TCP/UDP	Service de Nom Netbios
netbios-ssn	139	TCP/UDP	Service de Session Netbios
imap	143	TCP/UDP	Protocole d'accès messagerie Internet
SNMP	161	UDP	Administration de réseau

Tableau 1.1. Services et ports associés

Section 2. Eléments de cours sur l'adressage IP

Table of Contents

Adresses physiques (MAC) et adresses logiques (IP)
- Notion d'adresse Physique et de trames
- Notion d'adresse logique et de paquets
- Résolution d'adresses logiques en adresses physiques
- Attribution d'une adresse IP Internet

Adressage IP
- Structure des adresses IP
- Classes d'adresses
- Identification du réseau
- Adresses réservées

Les sous-réseaux
- Pourquoi créer des sous réseaux ?
- Masque de sous-réseau
- Sous-réseaux
- Adressage de sur-réseaux

Abstract

Le document présente l'adressage IP sur un réseau local et en environnement routé.
Ce document sert d'introduction à l'ensemble des cours et TP sur les différents protocoles
Mots clés : Adresse physique (MAC), Adresse IP, masque, sous-réseau, sur-réseau, CIDR

2.1 Adresses physiques (MAC) et adresses logiques (IP)

2.1.1 Notion d'adresse Physique et de trames

Deux cartes réseaux qui communiquent s'échangent des messages (suite de bits) appelés trames (frame). Tous les postes connectés au même câble reçoivent le message, mais seul celui à qui il est destiné le lit.

Comment sait-il que cette trame lui est adressée ?

Car il reconnaît l'adresse de destination, contenue dans la trame comme étant la sienne.

Comment sait-il qui lui a envoyé la trame ?

Car la trame contient aussi l'adresse de l'émetteur.

Au niveau de la couche liaison, les noeuds utilisent une adresse dite « physique » pour communiquer. L'adresse correspond à l'adresse de la carte réseau. On parle **d'adresse physique**, **d'adresse MAC** (Medium Access Control) ou d'adresse de couche 2 (référence au modèle OSI).

Cette adresse est identique pour les réseaux Ethernet, Token Ring et FDDI. **Sa longueur est de 48 bits** soit six octets (par exemple : 08-00-14-57-69-69) définie par le constructeur de la carte. Une adresse universelle sur 3 octets est attribuée par l'IEEE à chaque constructeur de matériel réseau. Sur les réseaux CCITT X.25, c'est la norme X.121 qui est utilisée pour les adresses physiques, qui consistent en un nombre de 14 chiffres.

L'adresse MAC identifie de manière unique un noeud dans le monde. Elle est physiquement liée au matériel (écrite sur la PROM), c'est à dire à la carte réseau.

2.1.2 Notion d'adresse logique et de paquets

L'adresse d'une carte réseau correspond à l'adresse d'un poste et d'un seul. Or les postes sont généralement regroupés en réseau.

Comment identifier le réseau auquel appartient le poste ?

Il faut une adresse logique qui soit indépendante de l'adresse physique.

C'est ce que propose le protocole IP et le protocole IPX.

Pourquoi identifier le réseau ?

Pour permettre à 2 postes qui ne sont pas connectés au même réseau de communiquer.

Cela est impossible avec une adresse MAC, il faut une adresse de niveau supérieur, comme nous le verrons un peu plus loin et surtout avec le routage IP.

Le message véhiculé par la trame va contenir une autre adresse destinataire dont un des objectifs sera de définir le réseau destinataire du message. On appelle le message contenu dans une trame un **paquet**.

Ce qu'il nous faut savoir à ce stade, c'est qu'une machine sait que le paquet n'est pas destiné au réseau si l'adresse réseau de destination est différente de la sienne, dans ce cas elle envoie le paquet à une machine spéciale (la passerelle ou routeur) dont le rôle est d'acheminer les paquets qui sortent du réseau.

Cette adresse dite **logique** du noeud (car elle est attribuée par logiciel à un **hôte**, plus précisément à une carte réseau) contenue dans le paquet est l'adresse IP, est définie indépendamment de toute topologie d'ordinateur ou de réseau. Son format reste identique quel que soit le support utilisé.

Les machines (hôtes) d'un réseau TCP/IP sont identifiées par leur adresse IP.

2.1.3 Résolution d'adresses logiques en adresses physiques

Toute machine sur un réseau IP a donc 2 adresses, une adresse MAC et une adresse IP.

Les processus de niveaux supérieurs utilisent toujours l'adresse IP et donc lorsqu'un processus communique avec un autre processus, il lui envoie un message dont l'adresse destinataire est une adresse IP, mais pour pouvoir atteindre la carte réseau du destinataire, il faut connaître son adresse MAC. Le rôle du protocole ARP (Adress Resolution Protocol) est d'assurer la correspondance entre l'adresse IP et l'adresse MAC.

2.1.4 Attribution d'une adresse IP Internet

Les réseaux connectés au réseau Internet mondial doivent obtenir un identificateur de réseau officiel auprès du bureau de l'Icann de l'Inter-NIC (Network Information Center) afin que soit garantie **l'unicité** des identificateurs de réseau IP sur toute la planète. Une adresse est attribuée au réseau privé dont l'administrateur en fait la demande auprès du NIC (http://www.nic.fr).

Après réception de l'identificateur de réseau, l'administrateur de réseau local doit attribuer des identificateurs d'hôte uniques aux ordinateurs connectés au réseau local. Les réseaux privés qui ne sont pas connectés à Internet peuvent parfaitement utiliser leur propre identificateur de réseau. Toutefois, l'obtention d'un identificateur de réseau valide de la part du centre InterNIC

leur permet de se connecter ultérieurement à Internet sans avoir à changer les adresses des équipements en place.
Chaque nœud (interface réseau) **relié à l'Internet doit posséder une adresse IP unique**.

2.2 Adressage IP

2.2.1 Structure des adresses IP

Les **adresses IP** sont des **nombres de 32 bits** qui contiennent 2 champs :

- Un **identificateur de réseau** (NET-ID): tous les systèmes du même réseau physique doivent posséder le même identificateur de réseau, lequel doit être unique sur l'ensemble des réseaux gérés.
- Un **identificateur d'hôte** (HOST-ID): un noeud sur un réseau TCP/IP est appelé hôte, *il identifie une station de travail, un serveur, un routeur ou tout autre périphérique TCP/IP au sein du réseau.*

La concaténation de ces deux champs constitue une **adresse IP unique** sur le réseau.
Pour éviter d'avoir à manipuler des nombres binaires trop longs, les adresses 32 bits sont divisées en 4 octets. Ce format est appelé la **notation décimale pointée,** cette notation consiste à découper une adresse en quatre blocs de huit bits. Chaque bloc est ensuite converti en un nombre décimal.
Chacun des octets peut être représenté par un nombre de 0 à 255.
Ex : 130.150.0.1
Exemple :
L'adresse IP 10010110110010000000101000000001 est d'abord découpée en quatre blocs : 10010110.11001000.00001010.00000001 puis, chaque bloc est converti en un nombre décimal pour obtenir finalement 150.200.10.1
= >4 nombres entiers (entre 0 et 255) séparés par des points.
= >4 octets
L'écriture avec les points est une convention, le codage en machine est binaire.

2.2.2 Classes d'adresses

La communauté Internet a défini **trois classes d'adresses** appropriées à des réseaux de différentes tailles. Il y a, a priori, peu de réseaux de grande taille (classe A), il y a plus de réseaux de taille moyenne (classe B) et beaucoup de réseaux de petite taille (classe C). La taille du réseau est exprimée en nombre d'hôtes potentiellement connectés.
Le premier octet d'une adresse IP permet de déterminer la classe de cette adresse.
Les adresses disponibles (de 0.0.0.0 à 255.255.255.255) ont donc été découpées en plages réservées à plusieurs catégories de réseaux.
Pour éviter d'avoir recours aux organismes NIC à chaque connexion d'un nouveau poste, chaque société se voit attribuer une plage d'adresse pour son réseau. Le nombre d'adresses disponibles dans chaque plage dépend de la taille du réseau de la société. Les grands réseaux sont dits de classe A (IBM, Xerox , DEC, Hewlett-Packard), les réseaux de taille moyenne sont de classe B (Microsoft en fait partie !), et les autres sont de classe C.

Classe	Début en binaire	Valeurs	Identificateur de réseau	Identificateur d'hôte
A	0...	*1 à 126*	*a*	*b,c,d*
B	10...	*128 à 191*	*a,b*	*c,d*
C	110...	*192 à 223*	*a,b,c*	*d*
D	1110...	*224 à 239*	multicast	*a,b,c,d*
E	1111...	*240 à 255*	réservées	expérimental

Figure 2.1. Classes d'adresses

Par exemple, l'adresse d'un poste appartenant à un réseau de classe A est donc de la forme : **0AAAAAAA**.xxxxxxxx.xxxxxxxx.xxxxxxxx, avec A fixé par le NIC et x quelconque.

Exemple

IBM a obtenu l'adresse 9 (en fait, on devrait dire 9.X.X.X, mais il est plus rapide de n'utiliser que la valeur du premier octet). 9 est bien de classe A car 9d=**0**0001001b

Cela signifie que chaque adresse IP du type 00001001.xxxxxxxx.xxxxxxxx.xxxxxxxx, avec x prenant la valeur 0 ou 1, fait partie du réseau d'IBM.

Malgré ces possibilités d'adressage, la capacité initialement prévue est insuffisante et sera mise à défaut d'ici quelques années. L'**IPNG** (*Internet Protocol Next Generation*) ou Ipv6 devrait permettre de résoudre ces difficultés en utilisant un adressage sur 16 octets noté en héxadécimal.

2.2.3 Identification du réseau

L'adresse IP se décompose, comme vu précédemment, en un numéro de réseau et un numéro de noeud au sein du réseau.

Afin de s'adapter aux différents besoins des utilisateurs, la taille de ces 2 champs peut varier.

On définit ainsi les 5 classes d'adresses notées A à E:

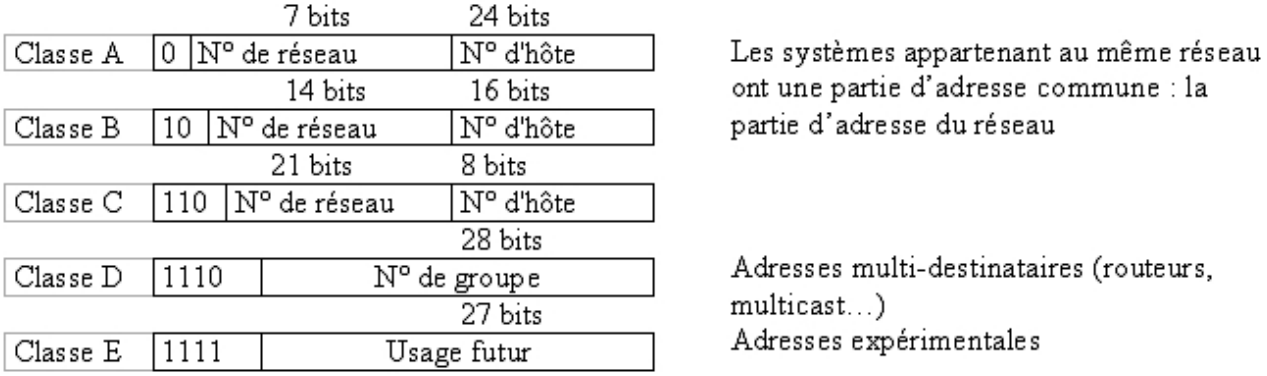


Figure 2.2. Classes d'adresses

ex. : Soit l'adresse IP suivante : 142.62.149.4

142 en décimal = 100011102 en binaire

Le mot binaire commence par les bits 102 donc il s'agit d'une adresse de classe B. Ou, plus simple : 142 est compris entre 128 et 191.

S'agissant d'une adresse de classe B, les deux premiers octets (a et b) identifient le réseau. Le numéro de réseau est donc : 142.62.0.0

Les deux derniers octets (c et d) identifient l'équipement hôte sur le réseau.

Finalement, cette adresse désigne l'équipement numéro 149.4 sur le réseau 142.62.

2.2.4 Adresses réservées

Les adresses réservées ne peuvent désigner une machine TCP/IP sur un réseau.

L'adresse d'acheminement par défaut (route par défaut.) est de type **0.X.X.X**. Tous les paquets destinés à un réseau non connu, seront dirigés vers l'interface désignée par **0.0.0.0**.

NB : 0.0.0.0 est également l'adresse utilisée par une machine pour connaître son adresse IP durant une procédure d'initialisation (DHCP).

L'adresse de bouclage(*loopback*): l'adresse de réseau 127 n'est pas attribuée à une société, elle est utilisée comme adresse de bouclage dans tous les réseaux. Cette adresse sert à tester le fonctionnement de votre carte réseau. Un ping 127.0.0.1 doit retourner un message correct. Le paquet envoyé avec cette adresse revient à l'émetteur.

Toutes les adresses de type 127.X.X.X ne peuvent pas être utilisées pour des hôtes. La valeur de 'x' est indifférente. On utilise généralement **127.0.0.1**

L'adresse de réseau est une adresse dont tous les bits d'hôte sont positionnés à 0 (ex 128.10.0.0 adresse de réseau du réseau 128.10 de classe B). Elle est utilisée pour désigner tous les postes du réseau. On utilise cette adresse dans les tables de routage.

Les noms de réseaux de type :

- X.Y.Z.0 (de 192.0.0.0 à 223.255.255.0) sont dits de classe C
- X.Y.0.0 (de 128.0.0.0 à 191.255.0.0) sont dits de classe B
- X.0.0.0. (de 1.0.0.0 à 126.255.255.254) sont dits de classe A

L'adresse de diffusion est *une adresse dont tous les bits d'hôte sont positionnés à 1 (ex : 128.10.255.255 adresse de diffusion du réseau 128 de classe B).*

Elle est utilisée pour envoyer un message à tous les postes du réseau.

Les adresses "privées"

Les adresses suivantes (RFC 1918) peuvent également être librement utilisées pour **monter un réseau privé** :

A 10.0.0.0 255.0.0.0

B 172.16.0.0 à 172.31.255.255 255.240.0.0

C 192.168.0.0 à 192.168.255.255 255.255.0.0

Aucun paquet provenant de ces réseaux ou à destination de ces réseaux, ne sera routé sur l'Internet (ces adresses sont néanmoins « routables » sur le réseau local).

Tableau Récapitulatif

	N°réseau	N°Hôte	
Classe A *126 réseaux*	**1 à 126**	0.0.1 à 255.255.254	*2^{24}-2= 16 777 214 adresses*
Classe B *2^14=16 384 réseaux*	**128.1** à 191.254	0.1 à 255.254	*2^{16}-2= 65 534 adresses*
Classe C *2^21-2=2 100 000 réseaux*	**192.0.1** à 223.255.254	1 à 254	*2^8-2= 254 adresses*

Figure 2.3. Récapitulatif Classes d'adresses

Le rôle du masque de réseau (netmask) est d'identifier précisément les bits qui concernent le N° de réseau d'une adresse (il "masque" la partie hôte de l'adresse).

Un bit à 1 dans le masque précise que le bit correspondant dans l'adresse IP fait partie du **N° de réseau ;** à l'inverse, **un bit à 0** spécifie un bit utilisé pour coder le **N° d'hôte**.

Ainsi, on a un masque dit "par défaut" qui correspond à la classe de ce réseau.

Exemple: dans un réseau de classe A sans sous-réseau, le premier octet correspond à l'adresse du réseau donc le **netmask** commence par 11111111 suivi de zéros soit **255.0.0.0.**

D'où le tableau suivant :

Classe	*Netmask*
A	*255.0.0.0*
B	*255.255.0.0*
C	*255.255.255.0*

Tableau 2.1 Masques

Ex : Si mon adresse IP est 149.127.1.110 alors je travaille avec une adresse de classe B. Mon N° de réseau est 149.127.0.0 et mon masque 255.255.0.0.

2.3 Les sous-réseaux

2.3.1 Pourquoi créer des sous réseaux ?

Les avantages de la segmentation en sous-réseau sont les suivants :

1. **Utilisation de plusieurs media** (câbles, supports physiques). La connexion de tous les noeuds à un seul support de réseau peut s'avérer impossible, difficile ou coûteuse lorsque les noeuds sont trop éloignés les uns des autres ou qu'ils sont déjà connectés à un autre media.
2. **Réduction de l'encombrement**. Le trafic entre les noeuds répartis sur un réseau unique utilise la largeur de bande du réseau. Par conséquent, plus les noeuds sont nombreux, plus la largeur de bande requise est importante. La répartition des noeuds sur des réseaux séparés permet de réduire le nombre de noeuds par réseau. Si les noeuds d'un réseau de petite taille communiquent principalement avec d'autres noeuds du même réseau, l'encombrement global est réduit.
3. **Economise les temps de calcul**. Les diffusions (paquet adressé à tous) sur un réseau obligent chacun des noeuds du réseau à réagir avant de l'accepter ou de la rejeter.
4. **Isolation d'un réseau**. La division d'un grand réseau en plusieurs réseaux de taille inférieure permet de limiter l'impact d'éventuelles défaillances sur le réseau concerné. Il peut s'agir d'une erreur matérielle du réseau (une connexion
5. **Renforcement de la sécurité**. Sur un support de diffusion du réseau comme Ethernet, tous les noeuds ont accès aux paquets envoyés sur ce réseau. Si le trafic sensible n'est autorisé que sur un réseau, les autres hôtes du réseau n'y ont pas accès.
6. **Optimisation de l'espace réservé à une adresse IP**. Si un numéro de réseau de classe A ou B vous est assigné et que vous disposez de plusieurs petits réseaux physiques, vous pouvez répartir l'espace de l'adresse IP en multiples sous-réseaux IP et les

assigner à des réseaux physiques spécifiques. Cette méthode permet d'éviter l'utilisation de numéros de réseau IP supplémentaires pour chaque réseau physique.

2.3.2Masque de sous-réseau

Les masques de sous-réseaux (*subnet mask*) permettent de **segmenter un réseau en plusieurs sous-réseaux**. On utilise alors une partie des bits de l'adresse d'hôte pour identifier des sous-réseaux.

L'adressage de sous-réseau permet de définir des organisations internes de réseaux qui ne sont pas visibles à l'extérieur de l'organisation. Cet adressage permet par exemple l'utilisation d'un routeur externe qui fournit alors une seule connexion Internet.

Toutes les machines appartenant à un sous-réseau possèdent le même numéro de réseau.

On utilise le même principe que pour le masque par défaut sur l'octet de la partie hôte auquel on va prendre des bits. Ainsi, le masque de sous-réseau d'une adresse de classe B commencera toujours par 255.255.xx.xx

Pour connaître l'adresse du sous-réseau auquel une machine appartient, on effectue en réalité un **ET logique** entre l'adresse de la machine et le masque.

Adresse : 200.100.40.33 11001000.01100100.00101000.00100001

Masque : 255.255.255.224 11111111.11111111.11111111.11100000

Opération ET 11001000.01100100.00101000.00100000

=> La machine appartient au sous-réseau : 200.100.40.32

Nous voyons dans ce deuxième exemple que nous avons pris 2 bits sur le dernier octet de notre adresse. Ces 2 bits vont nous permettre de construire plusieurs sous-réseaux:

Ex : adresse : 192.0.0.131

Masque : 255.255.255.192

Conversion de l'adresse en binaire : 11000000 00000000 00000000 10000011

Conversion du masque en binaire : 11111111 11111111 11111111 11000000

Décomposition de l'adresse (R,H) : **11000000 00000000 00000000 10**000011

La machine appartient au sous-réseau *192.0.0.128 et a l'adresse 3(11 en binaire)*

Pour des raisons de commodité, on préférera **réserver un octet entier** pour coder le numéro de sous réseau. De même la théorie ne nous oblige pas à prendre les **bits contigus d'un masque**, même si c'est ce que nous utiliserons en pratique.

Important : pour parer à d'éventuels problèmes de routage et d'adressage, tous les ordinateurs d'un réseau logique doivent utiliser le même masque de sous-réseau et le même identificateur de réseau.

2.3.3 Sous-réseaux

2.3.3.1 Nombre de sous-réseaux

Le nombre théorique de sous-réseaux est égal à 2^n, n étant le nombre de bits à 1 du masque, utilisés pour coder les sous-réseaux.

Exemple :

Adresse de réseau : 200.100.40.0

Masque : 255.255.255.224

224 = 11100000 donc **3 bits** pour le N° de **sous-réseau** et 5 bits pour l'hôte.

Le nombre de sous-réseaux est donc de *: 23 =8.*

Remarque : la RFC 1860 (remplacée par la RFC 1878) stipulait qu'un numéro de sous réseau ne peut être composé de bits tous positionnés à zéro ou tous positionnés à un.
Autrement dit, dans notre exemple, on ne pouvait pas utiliser le sous-réseau 0 et le sous-réseau 224. Le premier nous donnant une adresse de sous-réseau équivalente à l'adresse du réseau soit 200.100.40.0. Le deuxième nous donnant une adresse de sous-réseau dont l'adresse de diffusion se confondrait avec l'adresse de diffusion du réseau. Le nombre de sous-réseaux aurait alors été de seulement *: 2^3-2 =6.*
Il est donc important de savoir quelle RFC est utilisée par votre matériel pour savoir si les adresses de sous-réseau composées de bits tous positionnés à zéro ou tous positionnés à un sont prises en compte ou non.

2.3.3.2 Adresse des sous-réseaux

Il faut donc maintenant trouver les adresses des sous-réseaux valides en utilisant les bits à 1 du masque.
Pour l'exemple précédent, il faut utiliser les 3 premiers bits:
000 00000 = 0
001 00000 = 32
010 00000 = 64
011 00000 = 96
100 00000 = 128
101 00000 = 160
110 00000 = 192
111 00000 = 224
On constate que le pas entre 2 adresses de sous-réseau est de *32 = 25* correspondant *au nombre théorique d'hôtes par sous-réseau.*

2.3.3.3 Adresse de diffusion d'un sous-réseau

Il faut mettre **tous les bits de la partie hôte à 1**.
Cherchons l'adresse de diffusion des sous réseaux précédents.

- Avec le masque 255.255.255.224

Pour le sous-réseau 200.100.40.**32**
32 = **001** 00000 donc l'adresse de diffusion est *001 11111 = 63.*
L'adresse de diffusion complète est donc *200.100.40.63*
Pour le sous-réseau 200.100.40.**64** l'adresse de diffusion est 200.100.40.*95*
...ETC ...
Avec le masque 255.255.255.129
Pour le sous-réseau 200.100.40.1 l'adresse de diffusion est 200.100.40.127
Pour le sous-réseau 200.100.40.128 l'adresse de diffusion est 200.100.40.254
Pourquoi 254 et pas 255 car avec 255 le dernier bit serait à 1 donc on serait dans le sous-réseau 10000001 , en décimal 129.

2.3.3.4 Nombre de postes d'un sous-réseau

Le nombre de postes est égal à **2n**, **n** étant le nombre de **bits à 0** du masque permettant de coder l'hôte. A ce chiffre il faut enlever 2 numéros réservés :

- tous les bits à zéro qui identifie le sous-réseau lui-même.

- tous les bits à 1 qui est l'adresse de diffusion pour le sous-réseau.

Exemples :
Soit le masque 255.255.255.224
224 = 11100000 donc 3 bits pour le N° de sous-réseau et 5 bits pour l'hôte
le nombre de poste est donc de : 2^5 -2 =30 postes.
De même, avec le masque non contigu 255.255.255.129 le nombre de postes sera de *2^6-2 = 62 postes*

2.3.4 Adressage de sur-réseaux

En 1992 la moitié des classes B étaient allouées, et si le rythme avait continué, au début de 1994 il n'y aurait plus eu de classe B disponible et l'Internet aurait bien pu mourir par asphyxie ! Pour éviter la diminution des identificateurs de réseau, et la saturation des routeurs (nombre de routes trop important) les autorités d'lnternet ont conçu un schéma appelé **adressage de sur-réseaux** (ou super-réseaux).
L'adressage de sur-réseaux par opposition à la segmentation en sous-réseaux, emprunte des bits de l'identificateur de réseau pour les attribuer aux identificateurs d'hôtes afin d'optimiser le routage.
Par exemple, au lieu d'allouer un identificateur de réseau de classe B, dans une entreprise comportant 2000 hôtes, InterNic alloue une plage séquentielle de 8 identificateurs de réseau de classe C. Chaque identificateur de réseau de classe C gère 254 hôtes pour un total de 2 032 identificateurs d'hôte.
Alors que cette technique permet de conserver des identificateurs de réseau de classe B, elle crée un nouveau problème.
En utilisant des techniques de routage conventionnelles, les routeurs d'lnternet doivent désormais comporter huit entrées (en RAM) dans leurs tables de routage pour acheminer les paquets IP vers l'entreprise. La technique appelée **CIDR (Classless Inter-Domain Routing)** permet de réduire les huit entrées utilisées dans l'exemple précédent à une seule entrée correspondant à tous les identificateurs de réseau de classe C utilisés par cette entreprise.
Soit les huit identificateurs de réseau de classe C commençant par l'identificateur de réseau 220.78.**168.0** et se terminant par l'identificateur de réseau 220.78.**175.0**, l'entrée de la table de routage des routeurs d'lnternet devient :

Identificateur de réseau	*Masque de sous réseau*	*Masque de sous réseau (en binaire)*
220.78.168.0	*255.255.248.0*	11111111 11111111 *11111*000 00000000

Tableau 2.2 Entrée table de routage

En effet 168 en binaire donne : **10101**000
et 175 donne : **10101**111
la partie commune porte bien sur les 5 1ers bits
d'où le masque : **11111**000
Dans l'adressage de sur-réseaux, la destination d'un paquet est déterminée en faisant un ET logique entre l'adresse IP de destination et le masque de sous-réseau de l'entrée de routage. En cas de correspondance avec l'identificateur de réseau, la route est utilisée. Cette procédure est identique à celle définie pour l'adressage de sous-réseaux.

La notation CIDR définit une convention d'écriture qui spécifie le nombre de bits utilisés pour identifier la partie réseau (les bits à 1 du masque).
Les adresses IP sont alors données sous la forme :
142.12.42.145 / **24** <=> 142.12.42.145 255.255.255.0
153.121.219.14 / **20**<=> 153.121.219.14 255.255.240.0
Dans cette écriture les nombres **24** et **20** représentent le nombre de bits consacrés à la codification du réseau (et sous réseau).
Remarque : Les RFC 1518 et 1519 définissent le CIDR *(Classless Inter-Domain Routing).*

Section 3. Fichiers de configuration du réseau et commandes de base

Table of Contents

Présentation du document : les outils de l'administrateur réseau
Les fichiers de configuration
- Le fichier `/etc/hosts`
- Le fichier `/etc/networks`
- Le fichier `/etc/host.conf`
- Le fichier `/etc/resolv.conf`
- Les fichiers de configuration des interfaces réseau

Les outils de l'administrateur réseau
- La commande **ifconfig**
- La commande **arp**
- La commande **route**
- La commande **netstat**
- La commande **traceroute**
- La commande **dig**
- La commande **host**

Abstract

Présentation des principaux fichiers de configuration du réseau et des commandes d'administration système et réseau.

Présentation du document : les outils de l'administrateur réseau

Ce document présente les principaux fichiers de configuration d'une machine en réseau, et les commandes d'administration réseau.

Il est composé de 6 parties:

1. Les fichiers de configuration réseau
2. La commande **ifconfig**
3. La commande **arp**
4. La commande **route**
5. La commande **netstat**
6. La commande **traceroute**

3.1 Les fichiers de configuration

3.1.1 Le fichier `/etc/hosts`

Le fichier `hosts` donne un moyen d'assurer la résolution de noms, de donner un nom FQDN à un hôte

Exemple de fichier `hosts`

```
127.0.0.1  localhost localhost.localdomain
192.168.1.1  uranus.foo.org uranus
```

3.1.2 Le fichier `/etc/networks`

Il permet d'affecter un nom logique à un **réseau**

```
localnet   127.0.0.0
foo-net 192.168.1.0
```

Cette option permet par exemple d'adresser un réseau sur son nom, plutôt que sur son adresse. **route add *foo-net*** au lieu de **route add -net *192.168.1.0***.

3.1.3 Le fichier `/etc/host.conf`

Il donne l'ordre dans lequel le processus de résolution de noms est effectué. Voici un exemple de ce que l'on peut trouver dans ce fichier :

```
order hosts,bind
```

La résolution est effectuée d'abord avec le fichier `hosts`, en cas d'échec avec le DNS.

3.1.4 Le fichier `/etc/resolv.conf`

Il permet d'affecter les serveurs de noms.

Exemple

```
Nameserver 192.168.1.1
Nameserver 192.168.1.2
Nameserver 192.168.1.3
```

Ici le fichier déclare le nom de domaine et les 3 machines chargées de la résolution de noms.

3.1.5 Les fichiers de configuration des interfaces réseau

Vous trouverez ces fichiers dans `/etc/network/interfaces`. Voici un exemple qui contient 3 interfaces.

```
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)
# The loopback interface
# automatically added when upgrading

auto lo eth0 eth1

iface lo inet loopback

iface eth0 inet static
        address 192.168.90.1
        netmask 255.255.255.0
        network 192.168.90.0
        broadcast 192.168.90.255
        gateway 192.168.90.1

iface eth1 inet static
        address 192.168.0.1
        netmask 255.255.255.0
        network 192.168.0.0
        broadcast 192.168.0.255
```

3.2 Les outils de l'administrateur réseau

3.2.1 La commande ifconfig

La commande **ifconfig** permet la configuration locale ou à distance des interfaces réseau de tous types d'équipements (unité centrale, routeur). Sans paramètres, la commande **ifconfig** permet d'afficher les paramètres réseau des interfaces.

La ligne de commande est :

`ifconfig` *`interface adresse [parametres]`*.

Exemple : **`ifconfig` *`eth0 192.168.1.2`*** (affecte l'adresse 192.168.1.2 à la première interface physique).

Voici les principaux arguments utilisés :

`interface` logique ou physique, il est obligatoire,

`up` active l'interface

`down` désactive l'interface

`mtu` définit l'unité de transfert des paquets

`netmask` affecter un masque de sous-réseau

`broadcast` définit l'adresse de broadcast

`arp` ou `-arp` activer ou désactiver l'utilisation du cache arp de l'interface

`metric` paramètre utilisé pour l'établissement des routes dynamiques, et déterminer le " coût " (nombre de sauts ou " hops ") d'un chemin par le protocole RIP.

`multicast` active ou non la communication avec des machines qui sont hors du réseau.

`promisc` ou `-promisc` activer ou désactiver le mode promiscuité de l'interface. En mode *promiscuous*, tous les paquets qui transitent sur le réseau sont reçus également par l'interface. Cela permet de mettre en place un analyseur de trame ou de protocole.

*Description du résultat de la commande **ifconfig `eth0`** :*

1. eth0 Link encap:Ethernet HWaddr 00:80:C8:32:C8:1E
2. inet addr:192.168.1.1 Bcast:192.168.1.255 Mask:255.255.255.0
3. UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
4. RX packets:864 errors:0 dropped:0 overruns:0 frame:0
5. TX packets:654 errors:0 dropped:0 overruns:0 carrier:0
6. collisions:0
7. Interrupt:10 Base address:0x6100

Explications :

Ligne 1: l'interface est de type Ethernet. La commande nous donne l'adresse MAC de l'interface.

Ligne 2 : on a l'adresse IP celle de broadcast, celle du masque de sous-réseau

Ligne 3 : l'interface est active (UP), les modes broadcast et multicast le sont également, le MTU est de 1500 octets, le Metric de 1

Ligne 4 et 5 : RX (paquets reçus), TX (transmis), erreurs, suppressions, engorgements, collision

Mode d'utilisation :

Ce paragraphe décrit une suite de manipulation de la commande **ifconfig**.

Ouvrez une session en mode console sur une machine.

1 - Relevez les paramètres de votre machine à l'aide de la commande **ifconfig**. Si votre machine n'a qu'une interface physique, vous devriez avoir quelque chose d'équivalent à cela.

```
Lo  Link encap:Local Loopback
inet addr:127.0.0.1  Bcast:127.255.255.255  Mask:255.0.0.0
UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
RX packets:146 errors:0 dropped:0 overruns:0 frame:0
TX packets:146 errors:0 dropped:0 overruns:0 carrier:0
```

```
collisions:0

eth0 Link encap:Ethernet  HWaddr 00:80:C8:32:C8:1E
inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:864 errors:0 dropped:0 overruns:0 frame:0
TX packets:654 errors:0 dropped:0 overruns:0 carrier:0
collisions:0

Interrupt:10 Base address:0x6100
```

2 - Désactivez les 2 interfaces lo et eth0

ifconfig *lo* down

ifconfig *eth0* down

3 - Tapez les commandes suivantes :

ping *localhost*

ping *192.168.1.1*

telnet *localhost*

Aucune commande ne fonctionne, car même si la configuration IP est correcte, les interfaces sont désactivées.

4 - Activez l'interface de loopback et tapez les commandes suivantes :

ifconfig *lo* up /* activation de l'interface de loopback */

ping *localhost* ou telnet localhost /* ça ne marche toujours pas */

route add *127.0.0.1* /* on ajoute une route sur l'interface de loopback */

ping *localhost* ou **telnet *localhost*** /* maintenant ça marche */

ping *192.168.1.1* /* ça ne marche pas car il manque encore une route*/

On peut déduire que :

- pour chaque interface il faudra indiquer une route au protocole.
- dans la configuration actuelle, aucun paquet ne va jusqu'à la carte, donc ne sort sur le réseau.

Voici le rôle de l'interface loopback. Elle permet de tester un programme utilisant le protocole IP sans envoyer de paquets sur le réseau. Si vous voulez écrire une application réseau, (telnet, ftp, ou autre), vous pouvez la tester de cette façon.

5 - Activez l'interface eth0 et tapez les commandes suivantes :

ifconfig *eth0* up /* activation de l'interface */

route add *192.168.1.1*

ifconfig /* l'information Tx/Rx de l'interface eth0 vaut 0 */

/* Aucun paquet n'est encore passé par la carte.*/

ping *127.0.0.1*

ifconfig /* on voit que l'information Tx/Rx de lo est modifiée */

/* pas celle de eth0, on en déduit que les paquets */

/* à destination de lo ne descendent pas jusqu'à l'interface physique */

ping *192.168.1.1* /* test d'une adresse locale */

ifconfig /* Ici on peut faire la même remarque. Les paquets ICMP */

/* sur une interface locale, ne sortent pas sur le réseau */

/* mais ceux de l'interface lo sont modifiés*/

ping *192.168.1.2* /* test d'une adresse distante */

ifconfig /* Ici les paquets sont bien sortis. Les registres TX/RX de eth0 */
/* sont modifiés, mais pas ceux de lo */
6 -Réalisez les manipulations suivantes, nous allons voir le comportement de la commande **ping** sur les interfaces.
Sur la machine tapez la commande
192.168.1.1 ifconfig /* relevez les valeurs des registres TX/RX */
192.168.1.2 ping 192.168.1.1
192.168.1.1 ifconfig /* relevez les nouvelles valeurs des registres TX/RX */
/* il y a bien eu échange Réception et envoi de paquets*/
192.168.1.2 ping 192.168.1.3
192.168.1.1 ifconfig /* On voit que le registre Rx est modifié mais */
/* le registre Tx n'est pas modifié. La machine a bien reçu*/
/* le paquet mais n'a rien renvoyé */
192.168.1.2 ping 192.168.1.2
192.168.1.2 ifconfig /* aucun registre n'est modifié, donc les paquets */
/* ne circulent pas jusqu'à l'interface physique avec un ping*/
/* sur l'interface locale */
7 - le MTU (*Message Transfert Unit*) détermine l'unité de transfert des paquets.
Vous allez, sur la machine 192.168.1.1 modifier le MTU par défaut à 1500, pour le mettre à 300, avec la commande :
ifconfig *eth0 mtu 300*
Sur la machine d'adresse 192.168.1.2, vous allez ouvrir une session ftp et chronométrer le temps de transfert d'un fichier de 30 MO. Relevez le temps et le nombre de paquets transmis ou reçus (commande **ifconfig**, flags TX/RX).
Restaurez le paramètre par défaut sur la première machine.
Refaites le même transfert et comparez les chiffres. La différence n'est pas énorme sur le temps car le volume de données est peu important. Par contre la différence sur le nombre de paquets, elle, est importante.

3.2.2 La commande arp

Description de la commande
La commande **arp** permet de visualiser ou modifier la table du cache arp de l'interface. Cette table peut être statique et (ou) dynamique. Elle donne la correspondance entre une adresse IP et une adresse **MAC** (Ethernet).
A chaque nouvelle requête, le cache ARP de l'interface est mis à jour. Il y a un nouvel enregistrement. Cet enregistrement à une durée de vie (ttl ou *Time To Live*).
Voici un exemple de cache ARP obtenu avec la commande **arp -va** :

```
? (192.168.1.2) at 00:40:33:2D:B5:DD [ether] on eth0
>Entries: 1      Skipped: 0      Found: 1
```

On voit l'adresse IP et l'adresse MAC correspondante. Il n'y a qu'une entrée dans la table.
Voici les principales options de la commande **arp** :
arp -s (ajouter une entrée statique), exemple : **arp -s *192.168.1.2 00:40:33:2D:B5:DD***
arp -d (supprimer une entrée), exemple : **arp -d *192.168.1.2***
Voir la page **man** pour les autres options.
La table ARP et le fonctionnement du cache ARP.

Cela est réalisé par la configuration de tables ARP statiques.
Mode d'utilisation :
Attention à certaines interprétations de ce paragraphe. Il dépend de votre configuration. Soit vous êtes en réseau local avec une plage d'adresse déclarée, soit vous utilisez une carte d'accès distant.
Première partie :

1. Affichez le contenu de la table ARP avec la commande **`arp -a`**,
2. Supprimez chaque ligne avec la commande **`arp -d`** ***`@ip`***, où *`@ip`* est l'adresse IP de chaque hôte apparaissant dans la table,
3. La commande **arp `-a`** ne devrait plus afficher de ligne,
4. Faites un **ping**, sur une station du réseau local,
5. **arp `-a`**, affiche la nouvelle entrée de la table,
6. Ouvrez une session sur Internet, puis ouvrez une session ftp anonyme sur un serveur distant en utilisant le nom, par exemple `ftp.cdrom.com`. Utilisez une adresse que vous n'avez jamais utilisée, supprimez également tout gestionnaire de cache.
7. Affichez le nouveau contenu de la table avec **`arp -a`**. Le cache ARP ne contient pas l'adresse Ethernet du site distant, mais celle de la passerelle par défaut. Cela signifie que le client n'a pas à connaître les adresses Ethernet des hôtes étrangers au réseau local, mais uniquement l'adresse de la passerelle. Les paquets sont ensuite pris en charge par les routeurs.
8. Refaites une tentative sur le site choisi précédemment. Le temps d'ouverture de session est normalement plus court. Cela est justifié, car les serveurs de noms ont maintenant dans leur cache la correspondance entre le nom et l'adresse IP.

Deuxième partie :
La commande **arp** permet de diagnostiquer un dysfonctionnement quand une machine prend l'adresse IP d'une autre machine.

1. Sur la machine 192.168.1.1, faites un **ping** sur 2 hôtes du réseau 192.168.1.2 et 192.168.1.3,
2. A l'aide de la commande **arp**, relevez les adresses MAC de ces noeuds,
3. Modifiez l'adresse IP de la machine 192.168.1.2 en 192.168.1.3
4. relancez les 2 machines en vous arrangeant pour que la machine dont vous avez modifié l'adresse ait redémarré la première,
5. Sur la machine d'adresse 192.168.1.1, remettez à jour les tables ARP.
6. Quel est le contenu, après cela de la table ARP ?

Conclusion : vous allez avoir un conflit d'adresses. Vous allez pouvoir le détecter avec la commande **arp**. Autre problème, si vous faites un **telnet** sur 192.168.1.3, il y a de fortes chances pour que ce soit la machine qui était d'adresse 192.168.1.2, qui vous ouvre la session. Nous sommes (par une action volontaire bien sûr) arrivés à mettre la pagaille sur un réseau de 3 postes. Cette pagaille pourrait tourner vite au chaos sur un grand réseau, d'où la nécessité pour un administrateur de faire preuve d'une grande rigueur.

3.2.3 La commande route

La commande **route** a déjà été entrevue un peu plus haut, avec la commande **ifconfig**. Le routage définit le chemin emprunté par les paquets entre son point de départ et son point d'arrivée. Cette commande permet également la configuration de pc, de switchs de routeurs.
Il existe 2 types de routages :
- le routage statique
- le routage dynamique.

Le routage statique consiste à imposer aux paquets la route à suivre.
Le routage dynamique met en oeuvre des algorithmes, qui permettent aux routeurs d'ajuster les tables de routage en fonction de leur connaissance de la topologie du réseau. Cette actualisation est réalisée par la réception des messages reçus des noeuds (routeurs) adjacents.
Le routage dynamique permet d'avoir des routes toujours optimisées, en fonction de l'état du réseau (nouveaux routeurs, engorgements, pannes).
On combine en général le routage statique sur les réseaux locaux au routage dynamique sur les réseaux importants ou étendus.
Un administrateur qui dispose par exemple de 2 routeurs sur un réseau, peut équilibrer la charge en répartissant un partie du flux sur un port avec une route, et une autre partie sur le deuxième routeur.
Exemple de table de routage :

```
Kernel IP routing table
Destination Gateway Genmask    Flags  Metric  Ref Use  Iface
192.168.1.0    *  255.255.255.0 U      0       0   2    eth0
127.0.0.0      *  255.0.0.0     U      0       0   2    lo
default  192.168.1.9 0.0.0.0    UG     0       0   10   eth0
```

Commentaire généraux :
Destination : adresse de destination de la route
Gateway : adresse IP de la passerelle pour atteindre la route, * sinon
Genmask : masque à utiliser.
Flags : indicateur d'état (U - Up, H - Host - G - Gateway, D - Dynamic, M - Modified)
Metric : coût métrique de la route (0 par défaut)
Ref : nombre de routes qui dépendent de celle-ci
Use : nombre d'utilisation dans la table de routage
Iface : interface eth0, eth1, lo
Commentaire sur la 3ème ligne :
Cette ligne signifie que pour atteindre tous les réseaux inconnus, la route par défaut porte l'adresse 192.168.1.9. C'est la passerelle par défaut, d'où le sigle UG, G pour gateway.
Ajout ou suppression d'une route :

`route add [net | host] addr [gw passerelle] [métric coût] [ netmask masque] [dev interface]`

- *`net ou host`* indique l'adresse de réseau ou de l'hôte pour lequel on établit une route,
- adresse de destination,
- adresse de la passerelle,
- valeur métrique de la route,
- masque de la route à ajouter,
- interface réseau à qui on associe la route.

Exemples :

route add *127.0.0.1 lo* /* ajoute une route pour l'adresse 127.0.0.1 sur l'interface lo */

route add -net *192.168.2.0 eth0* /* ajoute une route pour le réseau 192.168.2.0 sur l'interface eth0 */

route add *saturne.foo.org* /* ajoute une route pour la machine machin sur l'interface eth0 */

route add default gw *ariane* /* ajoute ariane comme route par défaut pour la machine locale */

/* ariane est le nom d'hôte d'un routeur ou d'une passerelle */

/* gw est un mot réservé */

route add *duschmoll* netmask *255.255.255.192*

/* Encore un qui a créé des sous réseaux., Il s'agit ici d'une classe C */

/* avec 2 sous réseaux, il faut indiquer le masque. */

Suppression d'une route :

route del -net *192.168.1.0*

route del -net *toutbet-net*

Warning

Attention, si on utilise des noms de réseau ou des noms d'hôtes, il faut qu'à ces noms soient associés les adresses de réseau ou des adresses IP dans le fichier `/etc/networks` pour les réseaux, et `/etc/hosts` ou DNS pour les noms d'hôtes.

Vous pouvez également voir l'atelier sur la mise en place d'un routeur logiciel.

Petite étude de cas :

Première partie - réalisation d'une maquette

On dispose de 2 réseaux (A et B) reliés par une passerelle. Le réseau A est également relié à Internet par un routeur. Le réseau A dispose d'un serveur de noms. Chaque réseau a deux machines.

```
Réseau Nom du réseau  Machine  Nom des machines
A      metaux-net   192.3.2.2  platine
                    192.3.2.3  uranium
                    192.3.2.4  mercure(serveur de noms)
B      roches-net   130.2.0.2  quartz
                    130.2.0.3  silex
```

La passerelle entre le réseau A et B a 2 interfaces :

- eth0 192.3.2.1
- eth1 130.2.0.1

Le réseau A, a une passerelle par défaut pour Internet 130.2.0.9, qui est l'interface d'un autre routeur.

On veut :

- que les stations de chaque réseau puissent accéder à Internet,
- que les stations de chaque réseau puissent communiquer entre-elles,
- que les stations du réseau B, utilisent le serveur de noms le moins possible.

On demande :

1 - d'expliquer comment seront configurés les postes du réseau B,
2 - de donner la configuration des fichiers suivants pour chaque machine (`hosts`, `resolv.conf`, fichier de configuration de carte).
3 - de donner la liste des routes à mettre :
- sur les postes du réseau B,
- sur les postes du réseau A,
- sur la passerelle qui relie les 2 réseaux,
- sur le routeur du réseau A.

3.2.4 La commande netstat

La commande **netstat**, permet de tester la configuration du réseau, visualiser l'état des connexions, établir des statistiques, notamment pour surveiller les serveurs.
Liste des paramètres utilisables avec **netstat** :
Sans argument, donne l'état des connexions,
-a afficher toutes les informations sur l'état des connexions,
-i affichage des statistiques,
-c rafraîchissement périodique de l'état du réseau,
-n affichage des informations en mode numérique sur l'état des connexions,
-r affichage des tables de routage,
-t informations sur les sockets TCP
-u informations sur les sockets UDP.
Etat des connexions réseau avec **netstat**, dont voici un exemple :

```
 Proto  Recv-Q Send-Q Local Address  Foreign Address  State
 Tcp       0          126          uranus.planete.n:telnet  192.168.1.2:1037
ESTABLISHED
 Udp    0       0       uranus.plan:netbios-dgm  *:*
 Udp    0       0       uranus.plane:netbios-ns  *:*

Active UNIX domain sockets (w/o servers)
 Proto  RefCnt  Flags        Type        State         I-Node Path
 unix   2       [ ]          STREAM      1990     /dev/log
 unix   2       [ ]          STREAM CONNECTED 1989
 unix   1       [ ]          DGRAM        1955
```

Explications sur la première partie qui affiche l'état des connexions :
Proto : Protocole utilisé
Recv-Q : nbre de bits en réception pour ce socket
Send-Q : nbre de bits envoyés
LocalAdress : nom d'hôte local et port
ForeignAdress : nom d'hôte distant et port
State : état de la connexion
Le champ state peut prendre les valeurs suivantes:
Established : connexion établie
Syn snet : le socket essaie de se connecter
Syn recv : le socket a été fermé
Fin wait2 : la connexion a été fermée
Closed : le socket n'est pas utilisé

Close wait : l'hôte distant a fermé la connexion; Fermeture locale en attente.
Last ack : attente de confirmation de la fermeture de la connexion distante
Listen : écoute en attendant une connexion externe.
Unknown : état du socket inconnu
Explications sur la deuxième partie qui affiche l'état des sockets (IPC - *Inter Processus Communication*) actifs :
Proto : Protocole, en général UNIX,
Refcnt : Nombre de processus associés au socket
Type : Mode d'accès datagramme (DGRAM), flux orienté connexion (STREAM), brut (RAW), livraison fiable des messages (RDM)
State : Free, Listening, Unconnected, connecting, disconnecting, unknown
Path : Chemin utilisé par les processus pour utiliser le socket.
Affichage et état des tables de routage avec netstat : ***netstat -nr*** *ou* ***netstat -r***

```
Kernel IP routing table
Destination   Gateway   Genmask         Flags   MSS    Window  irtt  Iface
192.168.1.0   *         255.255.255.0   U       1500   0       0     eth0
127.0.0.0     *         255.0.0.0       U       3584   0       0     lo
```

Explications sur la commande ***netstat -r***
Destination : adresse vers laquelle sont destinés les paquets
Gateway : passerelle utilisée, * sinon
Flags : G la route utilise une passerelle, U l'interface est active, H on ne peut joindre qu'un simple hôte par cette route)
Iface : interface sur laquelle est positionnée la route.
Affichage de statistiques avec ***netstat -i***

```
Iface MTU  Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flags
Lo    3584 0   89    0      0      0      89    0      0      0      BLRU
eth0  1500 0   215   0      0      0      210   0      0      0      BRU
```

Explications sur la commande ***netstat -i***
RX-OK et TX-OK rendent compte du nombre de paquets reçus ou émis,
RX-ERR ou TX-ERR nombre de paquets reçus ou transmis avec erreur,
RX-DRP ou TX-DRP nombre de paquets éliminés,
RX-OVR ou TX-OVR recouvrement, donc perdus à cause d'un débit trop important.
Les Flags (B adresse de diffusion, L interface de loopback, M tous les paquets sont reçus, O arp est hors service, P connexion point à point, R interface en fonctionnement, U interface en service)

3.3.5La commande traceroute

La commande **traceroute** permet d'afficher le chemin parcouru par un paquet pour arriver à destination. Cette commande est importante, car elle permet d'équilibrer la charge d'un réseau, en optimisant les routes.
Voici le résultat de la commande **traceroute** ***www.nat.fr***, tapée depuis ma machine.

```
traceroute to sancy.nat.fr (212.208.83.2), 30 hops max, 40 byte packets
 1  195.5.203.9 (195.5.203.9)  1.363 ms  1.259 ms  1.270 ms
 2  194.79.184.33 (194.79.184.33)  25.078 ms  25.120 ms  25.085 ms
 3  194.79.128.21 (194.79.128.21)  88.915 ms  101.191 ms  88.571 ms
 4  cisco-eth0.frontal-gw.internext.fr (194.79.190.126)  124.796 ms[]
```

```
  5  sfinx-paris.remote-gw.internext.fr (194.79.190.250)  100.180 ms[]
  6  Internetway.gix-paris.ft.NET (194.68.129.236)  98.471 ms       []
  7  513.HSSI0-513.BACK1.PAR1.inetway.NET (194.98.1.214)  137.196 ms[]
  8  602.HSSI6-602.BACK1.NAN1.inetway.NET (194.98.1.194)  101.129 ms[]
  9  FE6-0.BORD1.NAN1.inetway.NET (194.53.76.228)  105.110 ms       []
 10  194.98.81.21 (194.98.81.21)  175.933 ms  152.779 ms  128.618 ms[]
 11  sancy.nat.fr (212.208.83.2)  211.387 ms  162.559 ms  151.385 ms[]
```

Explications :

Ligne 0 : le programme signale qu'il n'affichera que les 30 premiers sauts, et que la machine www du domaine nat.fr, porte le nom effectif de sancy, dans la base d'annuaire du DNS du domaine nat.fr. Cette machine porte l'adresse IP 212.208.83.2. Pour chaque tronçon, on a également le temps maximum, moyen et minimum de parcours du tronçon.

Ensuite, on a pour chaque ligne, l'adresse du routeur que le paquet a traversé pour passer sur le réseau suivant.

Ligne 4 et 5, le paquet a traversé 2 routeurs sur le même réseau 194.79.190.

Ligne 4, 5, 6, 7, 8, 9, 11, on voit que les routeurs ont un enregistrement de type A dans les serveurs de noms, puisqu'on voit les noms affichés.

Conclusion : depuis ma machine, chaque requête HTTP passe par 11 routeurs pour accéder au serveur www.nat.fr.

L'accès sur cet exemple est réalisé sur Internet. Un administrateur, responsable d'un réseau d'entreprise sur lequel il y a de nombreux routeurs, peut, avec cet outil, diagnostiquer les routes et temps de routage. Il peut ainsi optimiser les trajets et temps de réponse.

3.2.6 La commande dig

La commande **dig** remplace ce qui était la commande **nslookup**. Cette commande sert à diagnostiquer des dysfonctionnements dans la résolution de noms (Service DNS).

Utilisation simple de **dig** :

```
$ dig any freenix.org
; <<>> DiG 9.2.2 <<>> any freenix.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21163
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;freenix.org.                   IN      ANY

;; ANSWER SECTION:
freenix.org.            92341   IN      SOA     ns2.freenix.org.\
                                         hostmaster.freenix.org.\
                                                     2003042501\
                                                          21600\
                                                           7200\
                                                        3600000\
                                                         259200\

freenix.org.            117930  IN      NS      ns2.freenix.fr.
freenix.org.            117930  IN      NS      ns.frmug.org.
freenix.org.            117930  IN      NS      ns6.gandi.net.
```

```
;; AUTHORITY SECTION:
freenix.org.            117930  IN      NS      ns2.freenix.fr.
freenix.org.            117930  IN      NS      ns.frmug.org.
freenix.org.            117930  IN      NS      ns6.gandi.net.

;; ADDITIONAL SECTION:
ns2.freenix.fr.         16778   IN      A       194.117.194.82
ns.frmug.org.           40974   IN      A       193.56.58.113
ns6.gandi.net.          259119  IN      A       80.67.173.196

;; Query time: 197 msec
;; SERVER: 213.36.80.1#53(213.36.80.1)
;; WHEN: Tue May 27 15:16:23 2003
;; MSG SIZE  rcvd: 248
```

retourne les informations sur le domaine concerné.

Il est ensuite possible d'intérroger sur tout type d'enregistrement : SOA, MX, A, CNAME, PTR...

3.2.7 La commande host

La commande **host** interroge les serveurs de noms. Elle peut par exemple être utilisée pour détecter des dysfonctionnements sur un réseau (serveurs hors services). Attention, n'utilisez pas cette commande sur des réseaux dont vous n'avez pas l'administration.

Travaux Dirigés

Exercice 1 :

Vous êtes sur un réseau d'adresse 192.168.1.0 avec une interface d'adresse MAC 00:40:33:2D:B5:DD, vous n'avez aucun fichier *hosts* sur votre machine, il n'y a pas de DNS, la passerelle par défaut est 192.168.1.9.

Vous faites un **`ping` *`195.6.2.3`*** qui a une interface d'adresse MAC 00:45:2D:33:C2 est localisée sur Internet

Le réseau fonctionne parfaitement et tout est parfaitement configuré

Cochez la bonne réponse:

A - On a dans la table arp ? (192.168.1.2) at 00:40:33:2D:B5:DD [ether] on eth0

B - On a dans la table arp ? (192.168.1.2) at 00:45:2D:33:C2 [ether] on eth0

C - On a dans la table arp ? (195.6.2.3) at 00:40:33:2D:B5:DD [ether] on eth0

D - On a dans la table arp ? (195.6.2.3) at 00: 00:45:2D:33:C2 [ether] on eth0

E - Il faut un fichier `host`, ou DNS pour réaliser l'opération **ping** demandée

F - Il n'est pas possible dans la configuration actuelle d'atteindre l'hôte 195.6.2.3

Réponse F, car la plage d'adresse 192.168.1.1 à 192.168.1.254 n'est pas routée sur l'Internet, sinon vous auriez l'adresse de la passerelle par défaut dans le cache ARP.

Exercice 2 :

Vous êtes sur un réseau d'adresse 192.5.1.0 avec une interface d'adresse MAC 00:40:33:2D:B5:DD,

Vous n'avez aucun fichier host sur votre machine,

Il n'y a pas de DNS,

La passerelle par défaut est 192.5.1.9

Vous faites un **`ping` *`www.existe.org`*** dont l'adresse IP est 195.6.2.3, et qui a une interface d'adresse MAC 00:45:2D:33:C2

Le réseau fonctionne parfaitement et tout est parfaitement configuré

Cochez la bonne réponse:

A - On a dans la table arp ? (192.5.1.0) at 00:40:33:2D:B5:DD [ether] on eth0

B - On a dans la table arp ? (192.5.1.0) at 00:45:2D:33:C2 [ether] on eth0

C - On a dans la table arp ? (195.6.2.3) at 00:40:33:2D:B5:DD [ether] on eth0

D - On a dans la table arp ? (195.6.2.3) at 00: 00:45:2D:33:C2 [ether] on eth0

E - Il faut un fichier `host`, ou DNS pour réaliser l'opération **ping** demandée

F - Il n'est pas possible dans la configuration actuelle d'atteindre l'hôte 195.6.2.3

Réponse E, car la résolution de noms ne peut être effectuée

Exercice 3 :

Vous êtes sur un réseau d'adresse 192.5.1.0, sur une machine d'adresse 192.5.1.1, et une interface d'adresse MAC 00:40:33:2D:B5:DD,

Vous n'avez aucun fichier `host` sur votre machine,

Il n'y a pas de DNS

La passerelle par défaut est 192.5.1.9, d'adresse MAC 09:44:3C:DA:3C:04
Vous faites un **`ping 195.6.2.3`**, et qui a une interface d'adresse MAC 00:45:2D:33:C2
Le réseau fonctionne parfaitement et tout est parfaitement configuré
Cochez la bonne réponse:
A - On a dans la table arp ? (192.5.1.0) at 00:40:33:2D:B5:DD [ether] on eth0
B - On a dans la table arp ? (192.5.1.0) at 00:45:2D:33:C2 [ether] on eth0
C - On a dans la table arp ? (195.6.2.3) at 00:40:33:2D:B5:DD [ether] on eth0
D - On a dans la table arp ? (192.5.1.9) at 09:44:3C:DA:3C:04 [ether] on eth0
E - Il faut un fichier `host`, ou DNS pour réaliser l'opération **ping** demandée
F - Il n'est pas possible dans la configuration actuelle d'atteindre l'hôte 195.6.2.3

Réponse D, l'hôte a bien été trouvé, la table ARP a été mise à jour avec l'adresse IP de la passerelle par défaut et son adresse Ethernet.

Exercice 4 :
On donne les résultats de 3 commandes netstat ci-dessous, extraites de la même machine :
$ netstat -nr
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
198.5.203.0 0.0.0.0 255.255.255.0 U 1500 0 0 eth0
127.0.0.0 0.0.0.0 255.0.0.0 U 3584 0 0 lo
0.0.0.0 198.5.203.3 0.0.0.0 UG 1500 0 0 eth0
$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
Tcp 0 127 uranus.toutbet:telnet 194.206.6.143:1027 ESTABLISHED
$ netstat -i
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flags
Lo 3584 0 764 0 0 764 89 0 0 0 BLRU
eth0 1500 0 410856 0 0 33286 210 0 0 0 BRU
On demande :

1. Quels sont les noms et adresse de la machine consultée ?
2. Quel type de session est-elle en train de supporter ?
3. A quoi correspond l'adresse 198.5.203.3?
4. Pourquoi une interface porte-t-elle les Flags BLRU et l'autre BRU ?
5. Quelle est la taille des paquets utilisée par la passerelle par défaut ?

Section 4. Eléments de cours sur ARP

Table of Contents

Le protocole ARP
Processus de recherche de l'adresse physique

Abstract

Résumé sur ARP - RFC 826

4.1 Le protocole ARP

Chaque carte réseau possède une adresse physique MAC unique sur 48 bits (6 octets) . La communication entre machines ne peut donc avoir lieu que lorsque celles-ci connaissent leurs adresses MAC. Pour envoyer les paquets IP vers les autres noeuds du réseau, les noeuds qui utilisent les protocoles TCP/IP **traduisent les adresses IP de destination en adresses MAC**. Ainsi, lorsqu'un noeud `N1` du réseau TCP/IP X1.X1.X1.X1 veut émettre un paquet TCP/IP (dans une trame Ethernet) vers une machine `N2` d'adresse IP (X2.X2.X2.X2), il faut qu'il connaisse l'adresse Ethernet de `N2` (E2.E2.E2.E2.E2.E2). L'application émettrice ajoute son adresse IP au paquet et l'application réceptrice peut utiliser cette adresse IP pour répondre.
Sur les réseaux à diffusion, tels qu'Ethernet et Token-Ring, le **protocole** IP nommé **ARP** (*Address Resolution Protocol*) fait le lien entre les adresses IP et les adresses physiques (ou MAC).
Pour réaliser l'association *@ip / @ Ethernet* l'émetteur `N1` utilise le protocole ARP dont le principe est le suivant :
L'émetteur envoie une trame Ethernet de diffusion (broadcast), c'est-à-dire où @destinataire : tous à 1 soit **FFFFFFFFFFFF** contenant un message ARP demandant
qui est X2.X2.X2.X2 ?

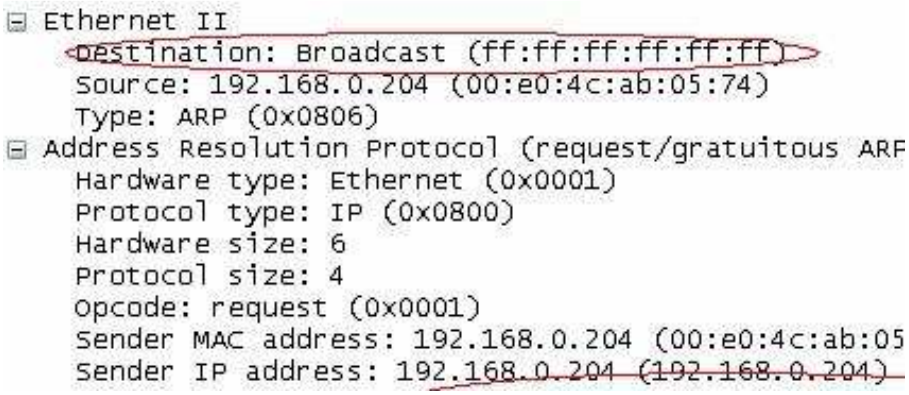


Figure 4.1. Adresses d'une trame Ethernet contenant une requête ARP Request

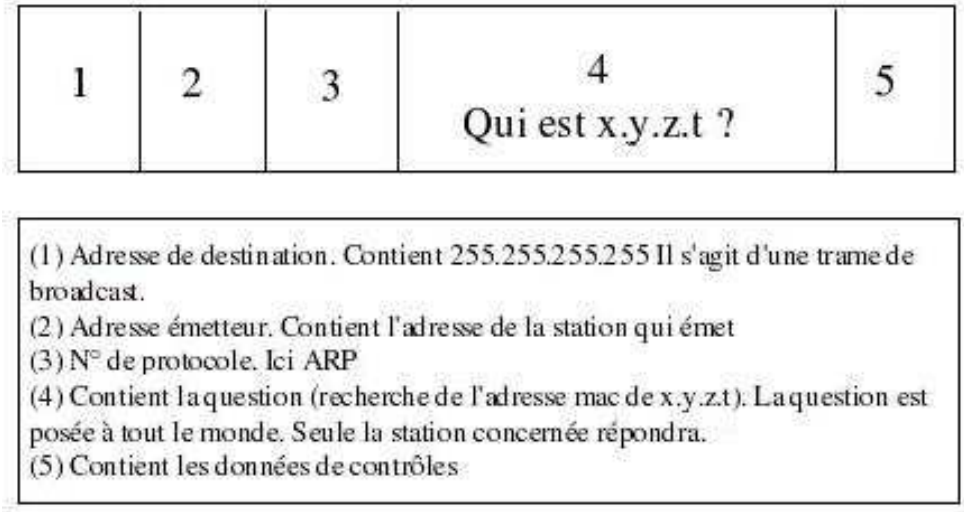


Figure 4.2. Trame Ethernet contenant une requête ARP

Toutes les machines IP du réseau local reçoivent la requête. `N2` qui a l'adresse X2.X2.X2.X2 se reconnaît, et elle répond à `N1` ie X1.X1.X1.X1 (dans une trame destinée à E1.E1.E1.E1.E1.E1)

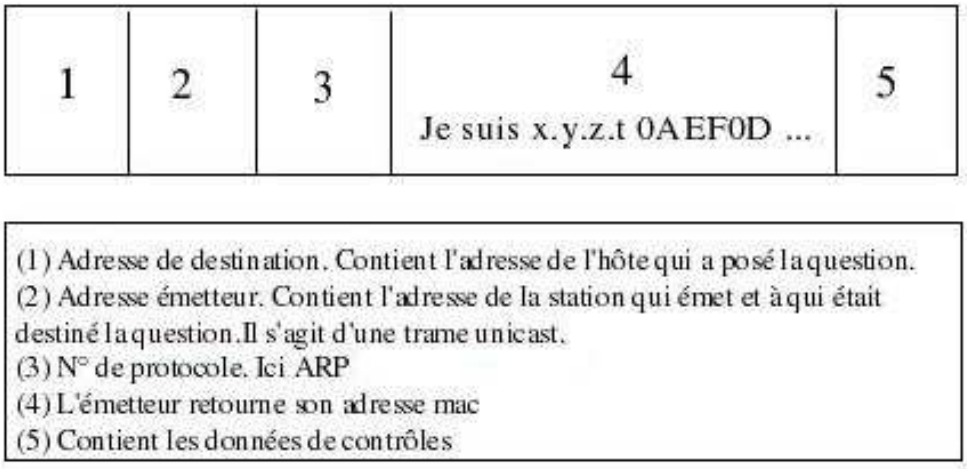


Figure 4.3. Trame Ethernet contenant une réponse ARP

4.2 Processus de recherche de l'adresse physique

Pour accélérer la transmission des paquets et réduire le nombre de requêtes de diffusion ARP, chaque noeud dispose d'un **cache ARP de résolution d'adresse**. Chaque fois que le noeud diffuse une requête ARP et reçoit une réponse, il crée une entrée dans une table de correspondance stockée en mémoire cache (@ip / @ Ethernet).

Lorsque le noeud envoie un autre paquet IP, il cherche d'abord l'adresse IP dans son cache. S'il la trouve, il utilise alors l'adresse physique correspondante pour son paquet.

Quand un poste **cherche l'adresse physique** correspondant à une adresse IP qu'il connaît, le protocole **ARP** se met en oeuvre et réalise donc les tâches suivantes :

1. **recherche dans le cache ARP** de la machine si l'adresse IP du destinataire y figure, s'il la trouve, il ajoute alors l'adresse physique correspondante dans l'en-tête de la trame à émettre.
2. réalisation d'un broadcast (**ARP request**) sur le réseau en demandant à qui correspond l'adresse IP à résoudre : il diffuse un paquet ARP qui contient l'adresse IP du destinataire
3. les machines du réseau comparent l'adresse demandée à leur adresse et le noeud correspondant renvoie son adresse physique au noeud qui a émis la requête (**ARP reply**) .
4. **stockage de l'adresse physique** lorsque le destinataire répond **dans le cache ARP** de la machine

Remarque : chaque entrée de la table à une durée de vie limitée (2 minutes mini sous Windows).

Voici pour exemple ce que donne le programme tcpdump avec la commande **ping *`192.168.1.2`*** à partir de la machine `uranus` alors que la table ARP de l'hôte `uranus` est vide :

```
13:17:14.490500 arp who-has 192.168.1.2 tell uranus.planete.net
13:17:14.490500 arp reply 192.168.1.2 is-at 0:40:33:2d:b5:dd
13:17:14.490500 uranus.planete.net > 192.168.1.2: icmp: echo request
13:17:14.490500 192.168.1.2 > uranus.planete.net: icmp: echo reply
13:17:15.500500 uranus.planete.net > 192.168.1.2: icmp: echo request
13:17:15.500500 192.168.1.2 > uranus.planete.net: icmp: echo reply
```

Explications :

Ligne 1, `uranus` demande qui est 192.168.1.2 (requête ARP) Le paquet est diffusé à tous les hôtes du réseau.

Ligne 2 réponse ARP : je suis à l'adresse Ethernet 00:40:33:2d:b5:dd

Lignes 3 à 6 : échanges de paquets ICMP entre les 2 hôtes.

Travaux Pratiques : La résolution d'adresses physiques

Table of Contents

Tester la présence d'un poste sur le réseau avec la commande ping
Interface couche 3 / couche 2 , IP / ethernet : utilisation du protocole ARP
- Le cache ARP
- La commande ARP
- Tromper ARP avec une adresse inexistante
- Tromper ARP avec une adresse Ethernet existante mais mal associée

Utilisation de l'analyseur de trames `Ethereal`
examen de paquets ARP

Conditions de réalisation

Ce TP nécessite des machines Windows ou Linux reliées par un réseau Ethernet et TCP/IP, et disposant d'un routeur connecté à internet. L'adresse du réseau utilisé par ce TP est 10.69.0.0, mais peut être modifiée sans problème. Il faut disposer d'un analyseur de trames, les exemples utilisent **Ethereal** (http://www.ethereal.com/).

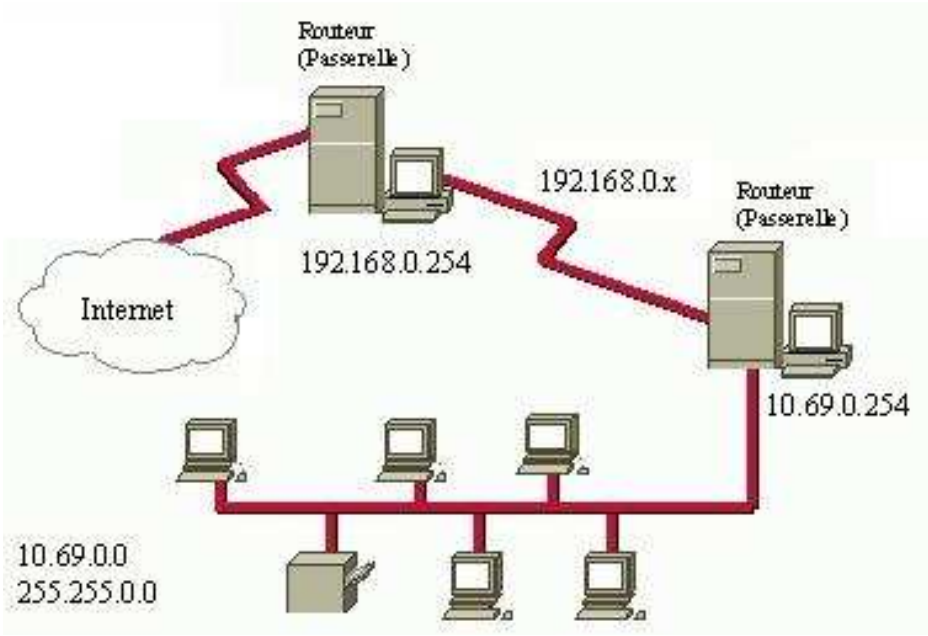


Figure 1. Schéma du réseau utilisé

Tester la présence d'un poste sur le réseau avec la commande ping

Ces tests peuvent être réalisés à partir d'un poste de travail Linux ou Windows

- à partir de votre poste, ouvrez une fenêtre d'invite de commandes
- quelle commande utilisez-vous pour afficher les informations concernant votre carte réseau ?
- testez l'adresse IP de loopback pour vérifier que les liaisons définies par TCP/IP sont correctes : **ping 127.0.0.1**
- testez avec l'adresse IP de votre poste
- testez avec l'adresse du poste d'un voisin
- testez avec l'adresse du réseau : **10.69.0.0** (notez le résultat)

Que signifient les réponses ?

- testez avec l'adresse de broadcast sur ce réseau : 10.69.**255.255** (notez le résultat)

Que signifient les réponses ?

- testez une adresse IP non utilisée sur le réseau par exemple **10.69.0.99** (idem)

Que peut-on déduire de la réponse ? Le poste existe t-il ? Le poste n'est pas actif ?

- testez l'adresse de la passerelle : **10.69.0.254**
- testez une adresse existante au delà de votre passerelle : **192.168.0.n** (idem)

Comment expliquez-vous votre résultat ? Pourquoi ce poste répond-il alors que son adresse réseau est différente de la vôtre ?

- testez une adresse au hasard ne faisant pas partie du réseau : **200.110.20.37** (idem)

Que pouvez-vous déduire du résultat ?

Interface couche 3 / couche 2 , IP / ethernet : utilisation du protocole ARP

2 stations pour dialoguer doivent utiliser les adresses MAC, il faut donc un moyen pour passer de l'adresse IP à l'adresse MAC et vice versa. IP fournit pour cela le protocole **ARP** (Adress Resolution Protocol) et **RARP** (Reverse Adress Resolution Protocol)

Le cache ARP

- Affichez le cache ARP : tapez la commande **arp -a**àl'invite de commandes
- Notez le résultat
- faites un ping sur une autre station
- Affichez de nouveau le cache arp (notez le résultat)

Que contient le cache ARP à l'issue du PING ?

- faites un ping sur l'adresse interne de votre passerelle internet (192.168.0.254)
- affichez le cache arp (notez le résultat)

Que contient le cache ? Expliquez le résultat. Cela confirme-t-il vos hypothèses précédentes (question 1) ?

- Décrivez les entrées du cache ARP

La commande ARP

Utilisez les commandes pour obtenir de l'aide sur la commande **arp** : **arp /?** ou **arp --help** et **man arp** sous Liniux

- videz le cache arp (notez l'instruction qu'il faut utiliser)
- Sous Linux reprenez les opérations du paragraphe précédent, afin de mettre des informations en cache. Pour vider le cache vous utiliserez les instructions suivantes :

```
ifconfig eth0 down
arp -a /* vous constaterez qu'il n'y a plus rien en cache */
ifconfig eth0 up
```

- A l'aide de la commande « **man ifconfig** », expliquez ce que font les commandes **ifconfig eth0 down** et **ifconfig eth0 up**.
  Reprenons les manipulations une fois le cache vidé
- notez l'adresse ethernet du poste voisin
- ajoutez manuellement cette entrée arp dans la cache (donnez l'instruction qu'il faut passer)
- affichez votre cache arp pour vérifier.

Quel est le type de l'entrée ?

- testez cette entrée avec ping

Quelles sont les différentes possibilités de la commande ARP ?

Tromper ARP avec une adresse inexistante

- utiliser la commande **arp** permettant de modifier l'adresse matérielle de la passerelle par défaut (donnez l'instruction qu'il faut utiliser)

(donnez lui par exemple 08-00-02-22-22-20 qui est une fausse adresse)

- faites un ping sur votre passerelle (notez le résultat)

expliquez-le

- faites un ping au delà de votre passerelle (notez le résultat)

expliquez-le

- supprimez cette entrée incorrecte

quelle est l'instruction ?

Tromper ARP avec une adresse Ethernet existante mais mal associée

- prenez l'adresse ethernet d'un poste
- prenez l'adresse ip d'un autre poste
- affectez avec arp l'adresse ip à l'adresse ethernet
- testez l'adresse ip avec ping

Notez le résultat et expliquez-le

Utilisation de l'analyseur de trames `Ethereal`

- Lancer `ethereal` et au besoin sélectionnez la carte réseau sur laquelle vous souhaitez capturer le trafic (vérifiez son adresse Mac)
- Démarrer la capture : **Capture/Start**,activez le rafraîchissement temps réel et **OK**
- générer du trafic réseau

- ouvrez une fenêtre d'invite de commande xterm
- faites un ping sur l'adresse ip de votre voisin

- arrêter la capture des données réseau. Revenez au moniteur réseau « **stop** »
- Analyser les données capturées

Le moniteur réseau vous affiche trois fenêtres.

Décrivez les types d'informations que vous obtenez dans chacune de ces fenêtres.

- Mettre en évidence des données capturées
  - Menu Display/Colorize Display.
  - Ajouter une option name : NomName, filter : icmp
  - Choisissez les couleurs de fond et de texte et validez.
- Afficher le détail des trames ICMP
- dans la première fenêtre sélectionnez une trame ICMP dont la colonne de description contient l'entrée « **REQUEST** ».
- dans la fenêtre de détails, cliquez sur le signe **+** devant **ICMP** (le contenu du paquet ICMP est mis en évidence et affiché selon la notation hexadécimale dans la fenêtre du bas)
- dans la fenêtre détail, cliquez sur **ICMP: Packet type = request**

Quel nombre hexadécimal correspond à ICMP: packet type =request ?

- cliquez sur **identifier** (notez sa valeur)

- cliquez sur **sequence number** (notez sa valeur)
- cliquez sur **data**

Les données reçues dans le message d'écho (**echo-request**)doivent être renvoyées dans le message de réponse d'écho (**echo-reply**).

Vérifiez-le.

- dans le menu affichage, cliquez sur cherchez la trame suivante qui doit être une trame **reply**
- sur la trame suivante notez le **packet type**, l'identifier et le**sequence number** comparez avec les valeurs précédentes, qu'en déduisez-vous ?
- enregistrer la capture à des fins d'analyse ultérieure (File / Save As)
- pour imprimer (File / Print)

Examen de paquets ARP

- démarrez une capture réseau
- générez un flux réseau (avec ping)
- arrêtez la capture
- mettez en évidence les trames comportant le protocole ARP_RARP
- Affichez le détail des trames **ARP:request**

- détail de **frame** : taille de la trame de base
- détail **Ethernet** : adresse destination
- *l'adresse de destination est-elle l'adresse physique(MAC d'une machine ?*
- *quelle est la partie constructeur de l'adresse physique?*
- *quelle est l'adresse source ?*
- *quel est le type de trame ethernet ?*
- *combien d'octets contient la trame ?*
- *A quoi servent les 14 premiers octets* ?

- détail **ARP_RARP**

- *adresse matérielle de l'expéditeur ?*
- *adresse IP de l'expéditeur ?*
- *adresse matérielle de la cible ?*
- *adresse IP de la cible ?*
- *pourquoi l'adresse matérielle de la cible contient-elle des FF* ?

- Affichez le détail de la trame **ARP:Reply**

- *répétez les opérations précédentes et comparez*
- *expliquez le fonctionnement du protocole ARP*.

Section 5. Le routage

Table of Contents

Principe
Acheminement des paquets TCP-IP
Les tables de routage
Acheminement Internet
    Domaine d'acheminement
    Principe du choix d'une voie d'acheminement
Routage dynamique

Abstract

Le document présente le routage IP sur un réseau local et en inter-réseau

Mots clés : routage, table de routage

5.1 Principe

Le routage dans Internet est similaire au **mécanisme d'adressage du courrier**.

Si vous adressez une lettre à un destinataire aux USA, à Los Angeles, dans l'état de Californie. Le bureau de poste de Belfort reconnaîtra que cette adresse n'est pas locale et transmettra le courrier au bureau français des PTT qui le remettra au service du mail US. Celui-ci s'en remettra à son bureau de la Californie, qui le transmettra au bureau de Los Angeles, qui connaît la localisation qui correspond à l'adresse dans la ville.

Avantages du système :

1. le bureau de poste local n'a pas à connaître toutes les adresses du monde
2. le chemin suivi peut être variable : chaque opérateur sait juste à qui remettre le courrier.

Le routage dans un réseau est identique :

Internet en entier est composé de réseaux autonomes qui s'occupent en interne de l'adressage entre leurs hôtes. Ainsi, tout datagramme arrivant sur un hôte quelconque du réseau destination sera acheminé à bon port par ce réseau seul.

Quand tous les hôtes participent au même réseau, chacun d'eux peut adresser des paquets aux autres sans difficulté. Par contre, si le destinataire est situé sur un autre réseau, le problème est de savoir où et à qui adresser le paquet puisque l'hôte expéditeur ne « voit » pas le destinataire.

On appelle **passerelle** (dans la terminologie TCP/IP) ou **routeur** un équipement qui fait le lien entre différents réseaux ou entre sous-réseaux. Ex de passerelle: **un ordinateur équipé**

de plusieurs adaptateurs réseau peut être relié avec chacune d'elle à un réseau physiquement séparé.

Les paquets d'un réseau qui sont adressés à l'autre réseau doivent passer par la passerelle. D'où la nécessité pour chaque hôte de connaître, sur son réseau, l'adresse IP d'un ou de plusieurs routeurs qui servent de passage vers le ou les réseaux qu'ils ne connaît pas.

Mettre en place le routage consiste à configurer chaque hôte du réseau de façon à ce qu'il sache vers quelle adresse de son propre réseau il doit adresser un paquet qui concerne un autre réseau (ou sous-réseau). Ces destinataires intermédiaires sont des routeurs qui prennent en charge le paquet.

Les hôtes pouvant être nombreux, bien souvent chacun ne connaît que l'adresse d'une passerelle (routeur) par défaut et ce sera cette passerelle qui « connaîtra » les adresses des autres routeurs.

5.2 Acheminement des paquets TCP-IP

Comment faire transiter des paquets entre 2 machines séparées par plusieurs routeurs?

Simplement chaque routeur doit connaître l'adresse du routeur suivant que doit emprunter le paquet pour arriver à destination. Ainsi le paquet arrive en sautant de routeur en routeur jusqu'à destination.

Mais concrètement comment ça se passe ?

Voici comment un hôte expéditeur se comporte pour adresser un paquet à un destinataire :

1. Il extrait l'adresse de réseau, voire de sous réseau de l'adresse du destinataire et la compare à sa propre adresse de réseau ou de sous réseau. S'il s'agit du même réseau, le paquet est expédié directement au destinataire en mettant en oeuvre ARP.
2. S'il ne s'agit pas du même réseau, l'expéditeur cherche dans sa table de routage une correspondance destinataire final / destinataire intermédiaire (routeur). Il cherche, en quelque sorte, sur son réseau, un hôte capable de servir de facteur vers un autre réseau.
3. L'expéditeur cherche d'abord à trouver dans sa table de routage locale l'adresse IP complète du destinataire,
4. s'il ne la trouve pas il cherche l'adresse du sous réseau du destinataire,
5. s'il ne la trouve pas, il cherche enfin l'adresse du réseau,
6. s'il ne trouve aucune correspondance, l'expéditeur cherche dans sa table l'adresse d'une passerelle à utiliser par défaut, (route 0.0.0.0)
7. s'il échoue là encore, le paquet, décidément bien encombrant, est supprimé.

Si l'une de ces recherches aboutit, la machine émettrice construit le paquet avec l'**adresse IP du destinataire** hors réseau. Elle l'encapsule dans une trame ayant comme **adresse MAC de destination** l'**adresse MAC du routeur**. La couche 2 du routeur lit la trame qui lui est adressée et la transmet à la couche 3 IP. Celle-ci récupère le paquet et s'aperçoit que le paquet ne lui est pas adressé, **elle consulte sa table de routage**, décide sur quelle nouvelle interface réseau le paquet doit être transmis, encapsule le paquet dans une nouvelle trame, et ainsi de suite de passerelle en passerelle jusqu'à destination.

5.3 Les tables de routage

Les réseaux IP sont interconnectés par des routeurs IP de niveau 3 (appelés abusivement en terminologie IP des gateways ou passerelles).

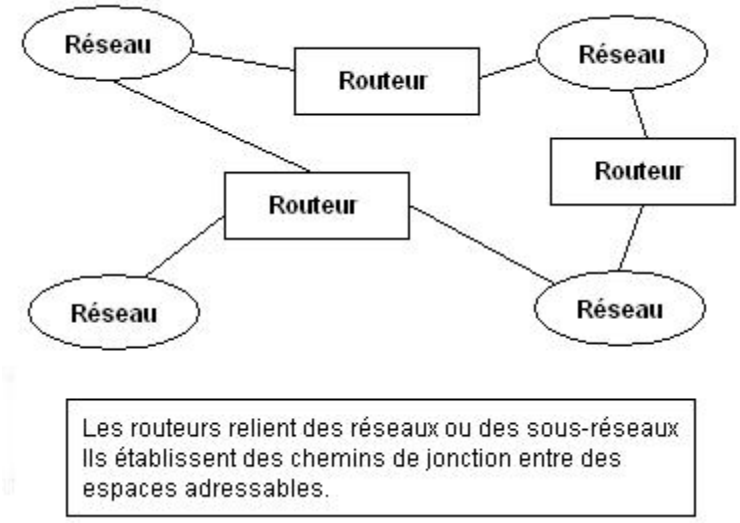


Figure 5.1. Routeurs interconnectés

Chaque hôte IP doit connaître le routeur par lequel il faut sortir pour pouvoir atteindre un réseau extérieur, c'est-à-dire avoir en mémoire une table des réseaux et des routeurs. Pour cela il contient une table de routage locale.

Dans une configuration de **routage statique**, une table de correspondance entre adresses de destination et adresses de routeurs intermédiaires est complétée « à la main » par l'administrateur, on parle de **table de routage**.

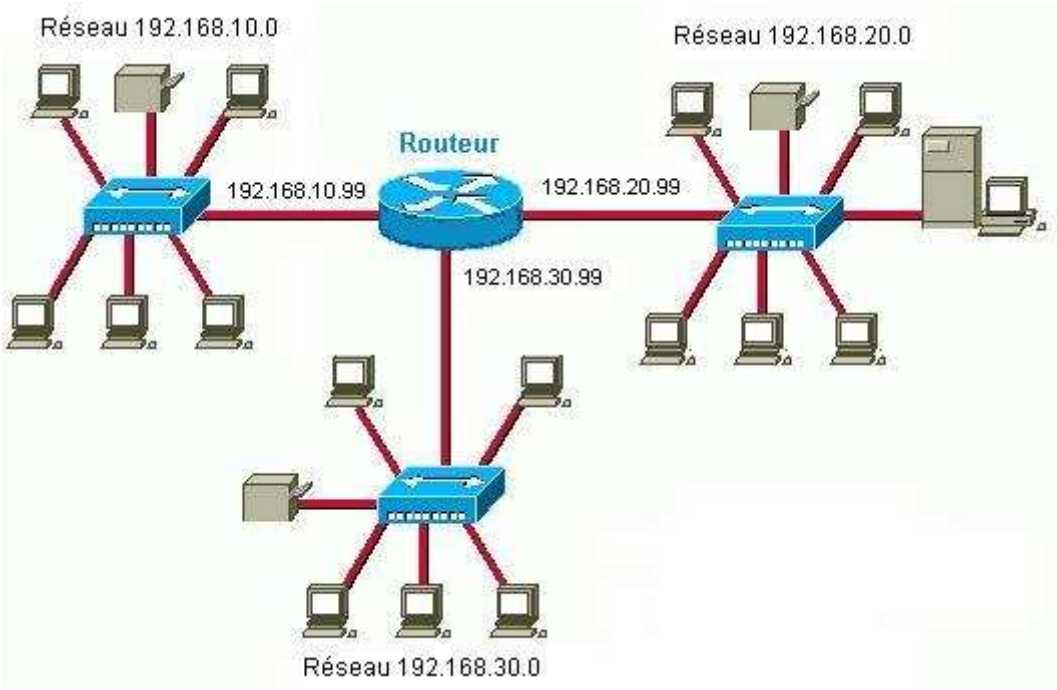


Figure 5.2. Schéma de routage

La table de routage d'un routeur comporte les adresses des réseaux de destination, le masque, les adresses des passerelles (routeurs intermédiaires) permettant de les atteindre, l'adresse de la carte réseau (interface) par laquelle le paquet doit sortir du routeur.

La commande **Route** permet d'afficher et de manipuler le contenu de la table de routage.

Considérons le schéma de réseau suivant :

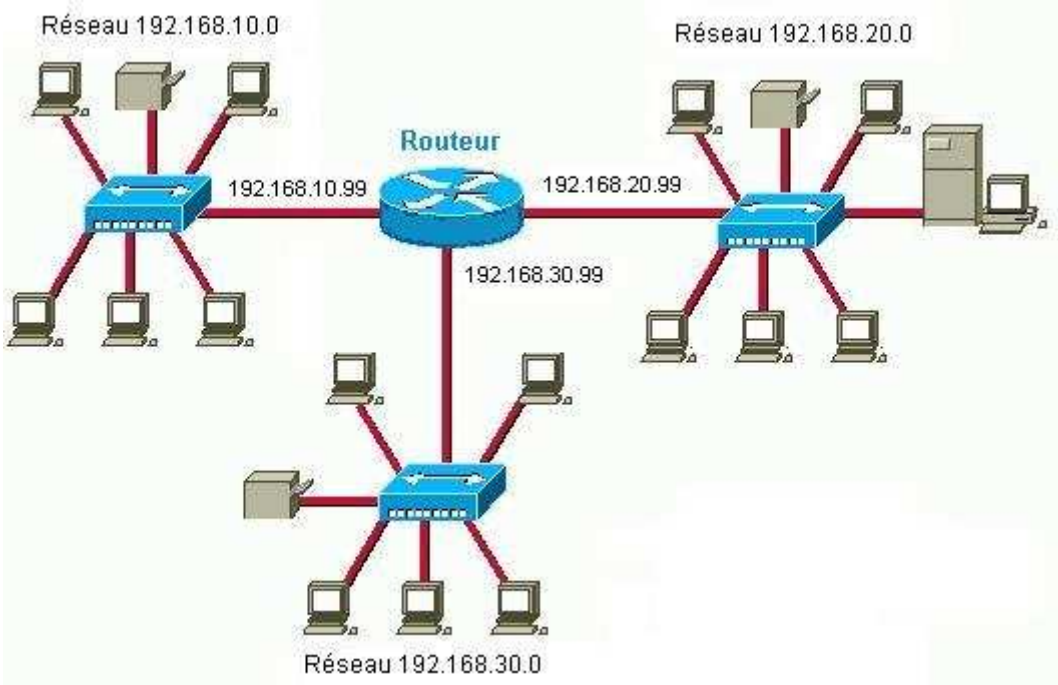


Figure 5.3. Schéma de réseau 1

La table de routage du routeur sera :

Destination	*Masque de Sous réseau*	*Passerelle*	*Interface*	
192.168.10.0	255.255.255.0	192.168.10.99	192.168.10.99	sortie de la passerelle vers le sous-réseau 10
192.168.20.0	255.255.255.0	192.168.20.99	192.168.20.99	sortie de la passerelle vers le sous-réseau 20
192.168.30.0	255.255.255.0	192.168.30.99	192.168.30.99	sortie de la passerelle vers le sous-réseau 30

Tableau 5.1 Table de routage

Ce réseau local est maintenant relié via un autre routeur à un 4ème réseau, le schéma devient :

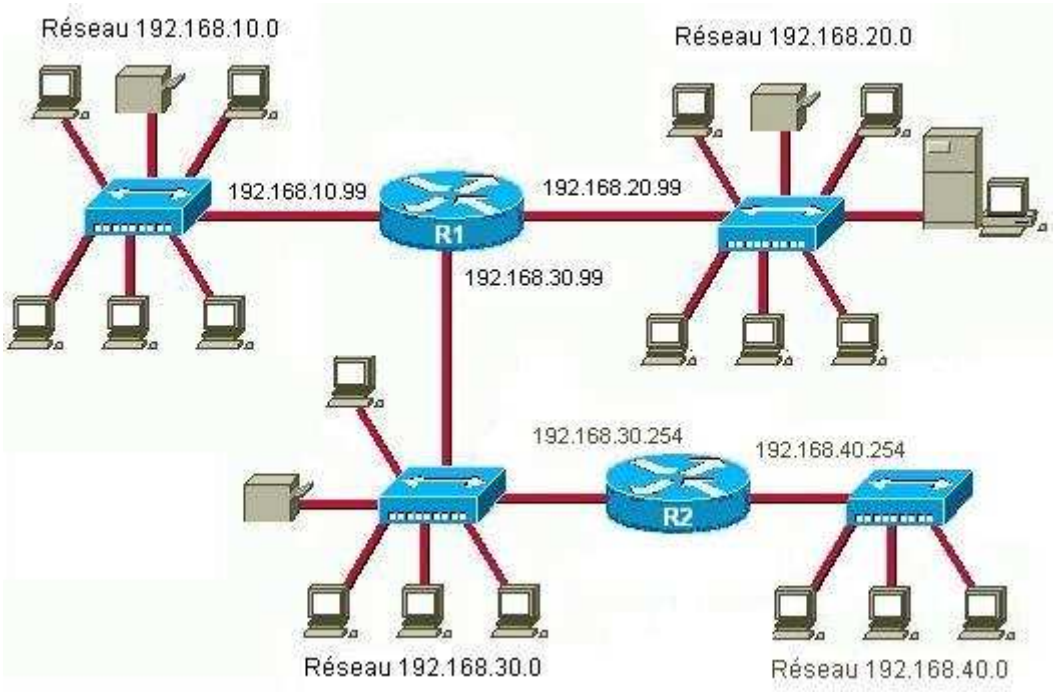


Figure 5.4. Schéma de réseau 2

La nouvelle entrée à ajouter dans **la table de routage du routeur R1** sera :

Destination	*Masque de Sous réseau*	*Passerelle*	*Interface*	
192.168.40.0	255.255.255.0	192.168.30.254	192.168.30.99	sortie de la passerelle vers le sous-réseau 40 via le routeur 192.168.30.254

Tableau 5.2. Nouvelle entrée de la table de routage

5.4 Acheminement Internet

5.4.1 Domaine d'acheminement

Les échanges entre passerelles de chaque domaine de routage font l'objet de protocoles particuliers : EGP (Exterior Gateway Protocol) et BGP (Border Gateway Protocol) plus récent. Ces protocoles envoient les paquets vers des destinations en dehors du réseau local vers des réseaux externes (Internet, Extranet...).

5.4.2 Principe du choix d'une voie d'acheminement

1. Si l'hôte de destination se trouve sur le réseau local, les données sont transmises à l'hôte destination
2. Si l'hôte destination se trouve sur un réseau à distance, les données sont expédiées vers une passerelle locale qui route le paquet vers une autre passerelle et ainsi de suite de passerelle en passerelle jusqu'à destination.

La commande **Tracert** permet de suivre à la trace le passage de routeur en routeur pour atteindre un hôte sur le réseau. La commande **Ping** permet de vérifier la fiabilité d'une route donnée.

5.5 Routage dynamique

Les **protocoles d'échange dynamique des tables de routage IP** sur un réseau local sont **RIP** (*Routing Information Protocol*) et le protocole **OSPF (Open Shortest Path First).** Dans une configuration de **routage dynamique**, un protocole (RIP ou OSPF) est mis en oeuvre pour construire dynamiquement les chemins entre routeurs.

Le protocole RIP permet à un routeur d'échanger des informations de routage avec les routeurs avoisinants. Dès qu'un routeur est informé d'une modification quelconque de la configuration sur les réseaux (telle que l'arrêt d'un routeur), il transmet ces informations aux routeurs avoisinants. Les routeurs envoient également des paquets de diffusion générale RIP périodiques contenant toutes les informations de routage dont ils disposent. Ces diffusions générales assurent la synchronisation entre tous les routeurs.

Avec un protocole comme RIP, on peut considérer que les tables de routages des routeurs et passerelles sont constituées et mises à jour automatiquement.

Section 6. Installation d'un serveur NFS

Le partage de fichiers pour les clients Unix/Linux

Table of Contents

Résumé

Installation des produits clients et serveurs

Les fichiers de configuration du serveur NFS

Les fichiers de configuration du client NFS

Exemple Unix de montage NFS

Configuration du serveur

Configuration et utilisation du client Unix/Linux

6.1 Résumé

Pourquoi un service NFS alors que celui-ci est très peu utilisé sur les environnements Windows et qu'il n'existe à ma connaissance pas de produits libres client ou serveur pour Windows. Pour deux raisons :

La première est que le service NFS est très largement employé dans les environnements Unix/Linux. Si vous avez des machines sous Linux vous utiliserez NFS. Il est donc nécessaire de connaître les procédures de configuration et d'utilisation de ce service.

La deuxième concerne Windows. Vous aurez sans doute un jour envie ou besoin d'installer le produit Windows Services For Unix (WSFU) de Microsoft. Ce produit disponible déjà sous Windows NT4 Server et mis à jour pour Windows 2000, offre de nombreux outils d'administration de type Unix pour Windows, dont un service NFS.

Nous allons voir, dans un environnement Linux, comment utiliser le service NFS.

Comment ça marche ?

À l'heure actuelle, il existe trois versions de NFS. La version 2 de NFS (NFSv2), une version plus ancienne qui est largement prise en charge. La version 3 de NFS (NFSv3) qui a davantage de fonctionnalités, y compris le traitement de fichiers de tailles variables et un meilleur rapportage d'erreurs, mais qui n'est pas entièrement compatible avec les clients NFSv2. La version 4 de NFS (NFSv4) qui inclut la sécurité Kerberos, fonctionne à travers des pare-feu et qui sur Internet, n'a plus besoin de portmapper, prend en charge les ACL et utilise des opérations avec état (ou qualifiées de stateful). Red Hat Enterprise Linux supporte les clients NFSv2, NFSv3 et NFSv4 et lors du montage d'un système de fichiers via NFS, Red Hat Enterprise Linux utilise NFSv4 par défaut, si le serveur le prend en charge.

Toutes les versions de NFS peuvent utiliser le protocole *TCP* (de l'anglais *Transmission Control Protocol*) exécuté sur un réseau IP, sachant qu'il est nécessaire pour NFSv4. NFSv2 et NFSv3 peuvent utiliser le protocole *UDP* (de l'anglais *User Datagram Protocol*) exécuté sur un réseau IP pour fournir une connexion réseau sans état (aussi qualifiée de stateless) entre le client et le serveur.

Lors de l'utilisation de NFSv2 ou NFSv3 avec UDP, la connexion UDP stateless dans des conditions normales minimise le trafic réseau, car le serveur NFS envoie un cookie au client une fois que ce dernier est autorisé à accéder au volume partagé. Ce cookie, qui représente une valeur aléatoire stockée côté serveur, est transmis en même temps que les requêtes RPC en provenance du client. Le serveur NFS peut être redémarré sans affecter le client et le cookie reste intact. Ceci étant, le protocole UDP étant sans état (ou stateless), si le serveur s'arrête inopinément, les clients UDP continuent à saturer le réseau de requêtes pour le serveur. Telle est la raison pour laquelle TCP est le protocole préféré lors de la connexion à un serveur NFS.

Lors de l'utilisation de NFSv4, une connexion dite stateful est effectuée et l'authentification Kerberos des utilisateurs et groupes avec des niveaux de sécurité variés est disponible de manière optionnelle. NFSv4 n'a pas d'interaction avec portmapper, `rpc.mountd`, `rpc.lockd` et `rpc.statd` étant donné qu'ils ont été incorporés au noyau. NFSv4 est en écoute sur le port bien connu 2049.

6.2 Installation des produits clients et serveurs

Sur une debian, vous pouvez installer les outils nécessaires sur un serveur grâce à **apt-get**.

Sur le serveur, il faut installer portmap, nfs-common, et nfs-kernel-server.

```
apt-get install portmap nfs-common nfs-kernel-server
```

Sur le client, il est faut installer nfs-common et portmat.

```
apt-get install portmap nfs-common
```

En ce qui concerne les sécurités, sachez que NFS utilise le wrapper tcp (tcpd). Il est possible de configurer la sécurité via les fichiers `/etc/hosts.allow` et `/etc/hosts.deny`. Les protocoles à ouvrir sur le serveur sont statd, nfsd, lockd, rquotad et mountd. Sur le client, il faut permettre à statd d'accèder à localhost.

Vous pouvez activer NFS par la commande **`/etc/init.d/nfs-kernel-server start`**. Il vous faudra au préalable avoir défini les ressources à partager (exporter).

Les programmes sur lequel s'appuie le service NFS utilisent les RPC (Remote Procedure Call). Ils s'inscrivent donc auprès du service portmap qui met à jour sa table de service rpc. Voici un extrait de ce que donne la commande **`rpcinfo -p`**

```
 program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
```

```
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100021    1   udp  33065  nlockmgr
    100021    3   udp  33065  nlockmgr
    100021    4   udp  33065  nlockmgr
    100021    1   tcp  38399  nlockmgr
    100021    3   tcp  38399  nlockmgr
    100021    4   tcp  38399  nlockmgr
    100005    1   udp    967  mountd
    100005    1   tcp    970  mountd
    100005    2   udp    967  mountd
    100005    2   tcp    970  mountd
    100005    3   udp    967  mountd
    100005    3   tcp    970  mountd
```

Voici maintenant les processus qui doivent être actifs sur le serveur NFS.

portmap gére le catalogue des programmes RPC,

mountd est chargé des opérations de montage/démontage d'arborescence,

nfsd exécute les primitives d'accès aux fichiers - requêtes émanant des clients.

6.3 Lancement et arrêt de NFS

Afin d'exécuter un serveur NFS, le service `portmap` doit être en cours d'exécution. Pour vérifier que `portmap` est activé, tapez la commande suivante en tant que super-utilisateur :

```
/sbin/service portmap status
```

Si le service `portmap` est en cours d'exécution, le service `fs` peut alors être lancé. Pour démarrer un serveur NFS, tapez la commande suivante en tant que super-utilisateur :

```
/sbin/service nfs start
```

Pour arrêter le serveur, tapez la commande suivante en étant connecté comme super-utilisateur :

```
/sbin/service nfs stop
```

L'option `restart` est un raccourci pour arrêter, puis redémarrer NFS. Cette option est la manière la plus efficace pour que les changements de configuration prennent effet après la modification du fichier de configuration pour NFS.

Pour redémarrer le serveur, tapez la commande suivante en tant que super-utilisateur :

```
/sbin/service nfs restart
```

L'option `condrestart` (*conditional restart*) ne lance `nfs` que s'il est actuellement en cours d'exécution. Cette option est utile pour les scripts, vu qu'elle ne lance pas le démon s'il n'est pas en cours d'exécution.

Pour redémarrer le serveur sous certaines conditions, tapez la commande suivante en tant que super-utilisateur :

```
/sbin/service nfs condrestart
```

Pour recharger le fichier de configuration du serveur NFS sans redémarrer le service, tapez la commande suivante en tant que super-utilisateur :

```
/sbin/service nfs reload
```

Par défaut, le service `nfs` ne se lance *pas* automatiquement au démarrage. Pour configurer NFS pour une exécution au démarrage, utilisez un utilitaire initscript tel que `/sbin/chkconfig`, **/sbin/ntsysv** ou l'**Outil de configuration des services**.

6.4 Configuration Client / Serveur

6.4.1 Les fichiers de configuration du serveur NFS

Il existe trois manières de configurer un serveur NFS sous Red Hat Enterprise Linux : en utilisant l'**Outil de configuration du serveur NFS** (`system-config-nfs`), en modifiant manuellement son fichier de configuration (`/etc/exports`) ou en exécutant la commande `/usr/sbin/exportfs`.

Pour obtenir des instructions sur l'utilisation de l'**Outil de configuration du serveur NFS**, reportez-vous au chapitre intitulé *Système de fichiers réseau (NFS)* du *Guide d'administration système de Red Hat Enterprise Linux*. Le reste de cette section examine la modification manuelle de `/etc/exports` et l'utilisation de la commande `/usr/sbin/exportfs` pour exporter les systèmes de fichiers NFS.

Le fichier `/etc/exports` permet non seulement de contrôler les systèmes de fichiers spécifiques qui sont exportés vers des hôtes distants, mais il permet également de spécifier des options. Les lignes blanches ne sont pas prises en compte, des commentaires peuvent être mentionnés en ajoutant un symbole dièse (#) en début de ligne et un retour à la ligne peut être introduit grâce à une barre oblique inverse (\). Chaque système de fichiers exporté doit avoir sa propre ligne et toutes les listes d'hôtes autorisés placées après un système de fichiers exporté doivent être séparées par des espaces. Les options pour chacun des hôtes doivent être placées entre parenthèses directement après l'identifieur d'hôte, sans espace entre l'hôte et la première parenthèse.

La ligne pour un système de fichiers exporté a la structure suivante :

```
<export> <host1>(<options>) <hostN>(<options>)...
```

Dans cette structure, remplacez *<export>* par le répertoire devant être exporté, remplacez *<host1>* par l'hôte ou le réseau vers lequel l'export est partagé et remplacez *<options>* par les options pour cet hôte ou ce réseau. Des hôtes supplémentaires peuvent être spécifiés dans une liste délimitée par des espaces.

Les méthodes suivantes peuvent être utilisées pour spécifier des noms d'hôtes :

- *hôte simple* — Où un hôte particulier est spécifié avec un nom de domaine pleinement qualifiée, un nom d'hôte ou une adresse IP.
- *caractères génériques* — Où les caractères `*` ou `?` sont utilisés pour prendre en compte un groupement de noms de domaines pleinement qualifiés qui correspondent à une chaîne de lettres donnée. Les caractères génériques (aussi appelés wildcards) ne devraient pas être utilisés avec les adresses IP ; il est toutefois possible qu'ils fonctionnent par accident, si les recherches inverses de DNS échouent.

  Soyez toutefois prudent lors de l'utilisation de caractères génériques avec des noms de domaines pleinement qualifiés, car ils sont souvent plus exacts que ce que vous escomptez. Par exemple, si vous utilisez `*.example.com` comme caractère générique, sales.example.com sera autorisé à accéder au système de fichiers exporté, mais pas bob.sales.example.com. Pour une correspondance incluant les deux noms de domaine, vous devrez spécifier `*.example.com` et `*.*.example.com`.

- *réseaux IP* — Autorisent la mise en correspondance d'hôtes en fonction de leur adresse IP dans un réseau plus grand. Par exemple, `192.168.0.0/28` autorisera les 16 premières adresses IP, de 192.168.0.0 à 192.168.0.15, à accéder au système de fichiers exporté, mais pas 192.168.0.16 ou une adresse IP supérieure.
- *groupes réseau* — Attribuent un nom de groupe réseau NIS, écrit ainsi : `@<group-name>`. Cette option attribue au serveur NIS la charge du contrôle d'accès pour ce système de fichier exporté, où les utilisateurs peuvent être ajoutés et supprimés dans un groupe NIS sans affecter `/etc/exports`.

Dans sa forme la plus simple, le fichier `/etc/exports` précise seulement le répertoire exporté et les hôtes autorisés à y accéder, comme dans l'exemple suivant :

```
/exported/directory bob.example.com
```

Dans cet exemple, `bob.example.com` peut monter `/exported/directory/`. Étant donné qu'aucune option n'est spécifiée dans cet exemple, les options NFS par défaut prennent effet :

- `ro` — Les montages du système de fichiers exporté sont en lecture-seule. Les hôtes distants ne peuvent pas modifier les données partagées sur le système de fichiers. Pour autoriser les hôtes à apporter des modifications au système de fichiers, l'option `rw` (lecture-écriture) doit être spécifiée.
- `wdelay` — Cette option entraîne un retard des opérations d'écriture sur le disque par NFS, s'il suspecte qu'une autre requête d'écriture est imminente. Ce faisant, les performances peuvent être améliorées grâce à une réduction du nombre d'accès au disque par des commandes d'écriture séparées, réduisant ainsi le temps d'écriture. L'option `no_wdelay` quant à elle, désactive cette fonction mais n'est disponible que lors de l'utilisation de l'option `sync`.
- `root_squash` — Cette option retire au super-utilisateur en connexion distante tous les privilèges de son statut en lui assignant l'ID d'utilisateur `nfsnobody` (personne). Ce faisant, le pouvoir du super-utilisateur distant est réduit au niveau d'utilisateur le plus bas, l'empêchant d'apporter des modifications non autorisées dans des fichiers sur le serveur distant. Sinon, l'option `no_root_squash` annule cette fonction de réduction des privilèges du super-utilisateur. Afin de limiter le champ d'action de chaque utilisateur distant, y compris le super-utilisateur, utilisez l'option `all_squash`. Pour

spécifier les ID d'utilisateur et de groupe à utiliser avec des utilisateurs distants d'un hôte particulier, utilisez respectivement les options `anonuid` et `anongid`. Dans ce cas, un compte utilisateur spécial peut être créé pour que les utilisateurs NFS distants le partagent et spécifient (`anonuid=<uid-value>,anongid=<gid-value>`), où `<uid-value>` correspond au numéro de l'ID d'utilisateur et `<gid-value>` représente le numéro de l'ID de groupe.

- **`Secure`** — requiert une authentification
- `Insecure` — ne requiert pas d'authentification
- `Noaccess` — permet d'exclure une partie de l'arborescence pour des clients donnés

Toutes les valeurs par défaut de chaque système de fichiers exporté doivent être explicitement écrasées. Par exemple, si l'option `rw` n'est pas spécifiée, le système de fichiers exporté est partagé en lecture-seule. L'exemple suivant est une ligne de `/etc/exports` qui écrase les deux options par défaut :

```
/another/exported/directory 192.168.0.3(rw,sync)
```

Dans cet exemple, `192.168.0.3` peut monter `/another/exported/directory/` en lecture/écriture et tous les transferts vers le disque sont validés avant que la requête d'écriture par le client ne soit achevée.

De plus, d'autres options sont disponibles là où il n'existe pas de valeur par défaut. Elles permettent d'annuler la vérification de la sous-arborescence, l'accès à des ports non-sûrs et les verrouillages non-sûrs de fichiers (nécessaires pour certaines implémentations anciennes de client NFS). Consultez la page de manuel de `exports` pour obtenir deplus amples informations sur ces options moins souvent utilisées.

Avertissement

Le format du fichier `/etc/exports` est très précis, particulièrement en ce qui concerne l'utilisation des caractères d'espacement. Rappelez-vous bien de toujours séparer les systèmes de fichiers exportés des hôtes, et les hôtes entre eux à l'aide d'un caractère d'espacement. Toutefois, aucun autre caractère d'espacement ne doit figurer dans le fichier, sauf sur des lignes de commentaire.

Par exemple, les deux lignes suivantes n'ont pas la même signification :

```
/home bob.example.com(rw)
/home bob.example.com (rw)
```

La première ligne autorise seulement les utilisateurs de `bob.example.com` à avoir un accès en lecture/écriture au répertoire `/home/`. La deuxième ligne elle autorise les utilisateurs de `bob.example.com` à monter le répertoire en lecture-seule (la valeur par défaut), alors que tout autre utilisateur peut le monter en lecture/écriture.

Exemple 1 de fichier `/etc/exports` :

```
# Ressource  Options   Liste_de_Clients
# Exporte /tmp vers la machine "cli" avec possibilité Read Write (rw)
# rw est l'option par défaut
```

```
/tmp  cli(rw)
#Exporte "/tmp" en lecture seule vers toutes les machines du réseau
/tmp  *(ro)
```

Exemple 2

```
# fichier /etc/exports d'exemple
/               maître(rw) confiance(rw,no_root_squash)
/projects       proj*.local.domain(rw)
/usr            *.local.domain(ro) @trusted(rw)
/home/joe       pc001(rw,all_squash,anonuid=150,anongid=100)
/pub            (ro,insecure,all_squash)
```

Commentaire :

La première ligne exporte l'ensemble du système de fichiers vers les machines maître et confiance. En plus des droits d'écriture, toute conversion d'UID est abandonnée pour l'hôte confiance.

La deuxième et la troisième ligne montrent des exemples de noms d'hôtes génériques, et de sous-réseaux (`@trusted').

La quatrième ligne montre une entrée pour le client PC/NFS présenté plus haut.

La dernière ligne exporte un répertoire public de FTP, à tous les hôtes dans le monde, en effectuant les requêtes sous le compte anonyme. L'option `insecure` permet l'accès aux clients dont l'implémentation NFS n'utilise pas un port réservé.

6.4.2 La commande `exportfs`

Chaque fichier exporté vers les utilisateurs distants via NFS ainsi que les droits d'accès liés à ces systèmes de fichiers sont énumérés dans le fichier `/etc/exports`. Lorsque le service `nfs` démarre, la commande `/usr/sbin/exportfs` se lance et lit ce fichier, passe le contrôle à `rpc.mountd` (si NFSv2 ou NFSv3 sont utilisés) pour le processus de montage proprement dit et ensuite à `rpc.nfsd` où les systèmes de fichiers sont alors disponibles pour les utilisateurs distants.

Lorsqu'elle est exécutée manuellement, la commande `/usr/sbin/exportfs` permet au super-utilisateur d'exporter ou de désexporter sélectivement des répertoires sans redémarrer le service NFS. Avec les options appropriées, la commande `/usr/sbin/exportfs` écrit les systèmes de fichiers exportés dans `/var/lib/nfs/xtab`. Puisque `rpc.mountd` se réfère au fichier `xtab` lorsqu'il décide des privilèges d'accès à un système de fichiers, les changements apportés à la liste des systèmes de fichiers exportés prennent effet immédiatement.

Ci-dessous figure une liste des options couramment utilisées pour `/usr/sbin/exportfs` :

- `-r` — Provoque l'export de tous les répertoires listés dans `/etc/exports` en dressant une nouvelle liste d'exports dans `/etc/lib/nfs/xtab`. Cette option rafraîchit effectivement la liste des exports avec les changements quelconques apportés à `/etc/exports`.

- `-a` — Provoque l'export ou le désexport de tous les répertoires, selon les autres options de la commande `/usr/sbin/exportfs`. Si aucune autre option n'est spécifiée, `/usr/sbin/exportfs` exporte tous les systèmes de fichiers spécifiés dans `/etc/exports`.
- `-o` *`file-systems`* — Spécifie les répertoires à exporter qui ne sont pas énumérés dans `/etc/exports`. Remplacez *`file-systems`* par les systèmes de fichiers supplémentaires à exporter. Ces derniers doivent être formatés selon le type spécifié dans `/etc/exports`. Cette option est souvent utilisée pour tester un système de fichiers exporté avant de l'ajouter de façon permanente à la liste des systèmes de fichiers à exporter.
- `-i` — Ne prend pas en compte `/etc/exports` ; seules les options données par la ligne de commande sont utilisées pour définir les systèmes de fichiers exportés.
- `-u` — Désexporte tous les répertoires partagés. La commande `/usr/sbin/exportfs -ua` suspend le partage de fichiers NFS alors que tous les démons NFS restent actifs. Pour activer à nouveau le partage NFS, tapez `exportfs -r`.
- `-v` — Représente une opération prolixe où les systèmes de fichiers à exporter ou à désexporter sont affichés avec beaucoup de détails lorsque la commande `exportfs` est exécutée.

Si aucune option n'est transmise à la commande `/usr/sbin/exportfs`, elle affiche une liste des systèmes de fichiers actuellement exportés.

Pour obtenir davantage d'informations sur la commande `/usr/sbin/exportfs`, reportez-vous à la page de manuel d'`exportfs`.

6.4.3 Utilisation de la commande exportfs avec NFSv4

Étant donné que NFSv4 n'utilise plus le protocole `rpc.mountd` comme dans NFSv2 et NFSv3, le montage de systèmes de fichiers a changé.

Un client NFSv4 a désormais la possibilité de voir l'ensemble des exports effectués par le serveur NFSv4 en tant qu'un seul système de fichiers qui porte le nom de pseudo-système de fichiers NFSv4. Sous Red Hat Enterprise Linux, le pseudo-système de fichiers est identifié comme un seul système de fichiers réel, identifié à l'export avec l'option `fsid=0`.

Par exemple, les commandes suivantes pourraient être exécutées sur un serveur NFSv4 :

```
mkdir /exports
mkdir /exports/opt
mkdir /exports/etc
mount --bind /usr/local/opt /exports/opt
mount --bind /usr/local/etc /exports/etc
exportfs -o fsid=0,insecure,no_subtree_check gss/krb5p:/exports
exportfs -o rw,nohide,insecure,no_subtree_check gss/krb5p:/exports/opt
exportfs -o rw,nohide,insecure,no_subtree_check gss/krb5p:/exports/etc
```

Dans cet exemple, on fournit aux clients de multiples systèmes de fichiers à monter, en utilisant l'option `--bind`.

6.4.4 Autres commandes d'administration

rpcinfo : (par exemple **rpcinfo `-p`** consulte le catalogue des applications RPC (nfsd, mountd sont des applicatifs RPC parmi d'autres).

nfsstat : fournit des statistiques d'utilisation de NFS.

6.4.5 L'identité des utilisateurs

Un des problèmes de NFS va être la gestion des utilisateurs et de leurs droits. Le serveur NFS va tenter d'identifier les users de la machine cliente par rapport au système habituel d'authentification de la machine. Tant que le client et le serveur ne se mettent pas d'accord sur une base commune d'identification, les confusions, voire les attaques, sont possibles. En effet, comment gérer les droits de l'utilisateur Pierre qui vient de la machine client Pluton si notre serveur ne connait aucun utilisateur Pierre ? Et d'abord, comment sait-on quel utilisateur est connecté ?

La réponse est simple : par son id. Or, si on a deux systèmes d'authentification différents et autonomes (par exemple, par `/etc/passswd`, sur chacune des machines), qui dit que l'id de Pierre sur la machine cliente ne sera pas celle de Paul sur la machine serveur ?

En fait, c'est encore pire : Si on ne met pas un système commun d'identification (et pour être sur, d'authentification), on est certain de confondre les utilisateurs. DE MANIERE GENERAL, IL VAUT MIEUX UTILISER NFS DANS UN ENVIRONNEMENT DE CONFIANCE

NFS offre de commandes qui permettent de manipuler les identifiants des utilisateurs afin de plier ceux-ci à notre système de sécurité (contrôle et modification des identifiants donnés par les clients).

Une fois que le système de fichiers NFS est monté en lecture-écriture par un hôte distant, la seule protection dont dispose chacun des fichiers partagés se situe au niveau de ses permissions. Si deux utilisateurs partageant la même valeur d'ID d'utilisateur montent le même système de fichiers NFS, ils pourront modifier les fichiers mutuellement. De plus, toute personne connectée en tant que super-utilisateur sur le système client peut utiliser la commande `su -` pour devenir un utilisateur ayant accès à des fichiers particuliers via le partage NFS.

Le comportement par défaut lors de l'export d'un système de fichiers via NFS consiste à utiliser la fonction de *réduction du super-utilisateur* (ou root squashing). Cette dernière permet de définir l'ID d'utilisateur d'une personne quelconque accédant au partage NFS en tant que super-utilisateur sur son ordinateur local, une valeur du compte personne (nobody) du serveur. Il est vivement conseillé de ne jamais désactiver cette fonction.

Si vous exportez un partage NFS en lecture-seule, songez à utiliser l'option `all_squash` qui donne à tout utilisateur accédant au système de fichiers exporté, l'ID d'utilisateur de `nfsnobody` (personne).

6.4.6 Configuration et utilisation du client Unix/Linux

6.4.6.1 Programmes client

Comme vu plus haut, n'oubiez pas d'avoir **portmap** actif sur votre client. Les programmes clients à utiliser sont : **mount** et **showmount**

6.4.6.2 Le fichier /etc/fstab

Ce fichier contient une table des volumes montés sur le système. Il est utilisé par les daemons mount, umount, fsck. Les volumes déclarés sont montés au démarrage du système.

Le fichier `/etc/fstab` est référencé par le service `netfs` au démarrage donc les lignes faisant référence aux partages NFS jouent le même rôle que la saisie manuelle de la commande `mount` durant le processus de démarrage.

Un exemple de ligne présente dans `/etc/fstab` permettant le montage d'un export NFS ressemble à l'exemple suivant :

```
<server>:</remote/export> </local/directory> <nfs-type> <options> 0 0
```

Remplacez *<server>* par le nom d'hôte, l'adresse IP ou le nom de domaine pleinement qualifié du serveur exportant le système de fichiers.

Remplacez *</remote/export>* par le chemin vers le répertoire exporté.

Remplacez *</local/directory>* par le système de fichiers local sur lequel le répertoire exporté est monté. Ce point de montage doit exister avant que `/etc/fstab` ne soit lu, sinon le montage échouera.

Remplacez *<nfs-type>* par `nfs` pour des serveurs NFSv2 ou NFSv3 ou par `nfs4` pour des serveurs NFSv4.

Remplacez *<options>* par une liste d'options séparées par des virgules pour le système de fichiers NFS. Reportez-vous à la page de manuel de `fstab` pour obtenir de plus amples informations.

Voici un extrait de fichier :

Exemple

```
/dev/hda1       /               ext2       defaults    1 1
/dev/hda2       swap            swap       defaults    0 0
/dev/fd0        /mnt/floppy     ext2       noauto      0 0
/dev/cdrom      /mnt/cdrom      iso9660    user,noauto,ro 0 0
ns1:/usr/local/man  /doc        nfs        rsize=8192,wsize=8192,timeo=14,intr
```

La dernière ligne indique que le volume `/usr/local/man`, situé sur le serveur `ns1`, et qui contient les pages du manuel est un volume nfs, monté sous le nom de local de `/doc`.

Ce fichier évite d'avoir à " monter " manuellement des systèmes de fichiers ou d'avoir à indiquer les points de montage, bien que cela puisse s'avérer parfois nécessaire (utilisation ponctuelle d'une ressource...). L'option `auto` permet de préciser si le montage doit etre fait automatiquement au démarrage de la machine. L'option `noauto` permet d'indiquer le montage tel qu'il doit être fait, lors d'une demande manuelle de montage (pratique pour les disquettes, CD et autres lecteurs amovibles).

6.4.6.3 autofs

Un inconvénient lors de l'utilisation de `/etc/fstab` est que, indépendamment de la fréquence d'utilisation de ce système de fichiers monté via NFS, le système doit allouer des ressources pour que ce montage demeure. Ceci n'est pas un problème pour un ou deux montages, mais si votre système maintient le montage de douzaines de systèmes à un moment donné, les performances générales du système peuvent en pâtir. Une alternative à `/etc/fstab` consiste à utiliser l'utilitaire basé sur le noyau nommé `automount` qui montera et démontera automatiquement les systèmes de fichiers NFS, économisant ainsi des ressources.

Le service `autofs` sert à contrôler la commande `automount` par le biais du fichier de configuration primaire `/etc/auto.master`. Alors que la commande `automount` peut être spécifiée dans une ligne de commande, il est plus commode de spécifier les points de montage, nom d'hôte, répertoire exporté et autres options dans un ensemble de fichiers, plutôt que de tous les taper manuellement.

Les fichiers de configuration `autofs` sont organisés selon une relation parent-enfant. Le fichier de configuration principal (`/etc/auto.master`) énumère des points de montage sur le système qui sont liés à un *type de correspondance* (map type) particulier, prenant la forme d'autres fichiers de configuration, programmes, chemins NIS et autres méthodes de montage moins courantes. Le fichier `auto.master` contient des lignes se référant à chacun de ces points de montage, organisées de la manière suivante :

```
<mount-point> <map-type>
```

L'élément *<mount-point>* de cette ligne indique l'emplacement du montage sur le système de fichiers local. L'option *<map-type>* fait référence à la manière dont le point de montage sera monté. La méthode la plus courante pour monter automatiquement des exports NFS consiste à utiliser un fichier en tant que type de chemin d'accès pour un point de montage particulier. Le fichier de chemin d'accès est généralement nommé `auto.`*<mount-point>*, où *<mount-point>* est le point de montage désigné dans `auto.master`. Les fichiers de chemin d'accès contiennent une ligne similaire à celle reproduite ci-dessous pour monter un export NFS :

```
</local/directory>  -<options>  <server>:</remote/export>
```

Remplacez *</local/directory;>* par le système de fichiers local où le répertoire exporté doit être monté. Ce point de montage doit exister avant que le fichier de chemin d'accès ne soit lu, sinon le montage échouera.

Remplacez *<options>* par une liste d'options séparées par des virgules pour le système de fichiers NFS. Assurez-vous de bien inclure un tiret (`-`) immédiatement avant la liste d'options.

Remplacez *<server>* par le nom d'hôte, l'adresse IP ou le nom de domaine pleinement qualifié du serveur exportant le système de fichiers.

Remplacez *</remote/export>* par le chemin vers le répertoire exporté.

Remplacez *<options>* par une liste d'options séparées par des virgules pour le système de fichiers NFS.

Bien que les fichiers de configuration `autofs` puissent être utilisés pour une variété de montages applicables à de nombreux types de périphériques et de systèmes de fichiers, ils sont particulièrement utiles lors de la création de montages NFS. Par exemple, des organisations stockent le répertoire `/home/` d'un utilisateur sur un serveur central via le partage NFS. Ensuite, elles configurent le fichier `auto.master` sur chacun des postes de travail pour qu'il renvoie à un fichier `auto.home` contenant les spécifications du montage du répertoire `/home/` via NFS. Ce faisant, l'utilisateur peut alors accéder à ses données personnelles et aux fichiers de configuration dans son répertoire `/home/` en se connectant sur un ordinateur quelconque du réseau. Le fichier `auto.master` correspondant à une telle situation ressemblerait à l'extrait reproduit ci-dessous :

```
/home   /etc/auto.home
```

Cette ligne instruit le système d'installer le point de montage `/home/` sur le système local qui doit être configuré selon le fichier `/etc/auto.home` ressemblant à l'extrait suivant :

```
*                       -fstype=nfs4,soft,intr,rsize=32768,wsize=32768,nosuid
server.example.com:/home
```

Cette ligne stipule que tout répertoire auquel un utilisateur tente d'accéder dans le répertoire `/home/` local (en raison de l'astérisque) devrait entraîner un montage NFS sur le système `server.example.com` au point de montage `/home/`. Les options de montage spécifient que chaque montage NFS du répertoire `/home/` devrait utiliser une série particulière de paramètres.

Pour davantage d'informations sur les fichiers de configuration de `autofs`, reportez-vous à la page de manuel de `auto.master`.

6.4.6.4 Montage manuel de système de fichiers

La commande souvent utilisée est de la forme

mount -t TypeDeSGF *NomDeMontage VolumeMonté*.

Vous pourrez avoir toutes les options avec la commande **man *mount*** ou une aide plus brève avec **mount --help**.

Exemple de montage : **mount -t nfs *ns1:/usr/local/man /doc*.**

La forme standard de la commande **mount** est **mount -t type *périphérique répertoire*** avec :

Type : type de sgf (fat, vfat, nfs, ext2, minix....) pour nous c'est nfs

Périphérique : nom du fichier exporté sous la forme NomServeur:NomDossierExporté

Répertoire : nom du répertoire local de montage/

Le type de fichier que vous montez est de type nfs, vous utiliserez l'exemple de la commande ci-dessous :

mount `-t nfs` ***`serveurNFS:/usr/share/doc /mnt/doc`***

Commentaire : la ligne de commande monte le répertoire exporté `/usr/share/doc` du serveur serveurNFS, sur le répertoire local du client `/mnt/doc`.

Le `mtab` est modifié chaque fois que l'utilisateur " monte " ou " démonte " un système de fichiers. Le système tient à jour une table des volumes montés.

Attention, l'accès à la commande **mount** n'est, par défaut, autorisée que pour root.

Il faut rajouter l'option `user` dans le fichier `/etc/fstab`, afin qu'un autre utilisateur puisse accéder à cette commande.

Exemple : /dev/cdrom /mnt/cdrom iso9660 noauto,ro

devient /dev/cdrom /mnt/cdrom iso9660 user,noauto,ro

La prise en compte des modifications est dynamique.

La commande **mount** sans paramètres donne la liste des volumes montés. La commande consulte la table maintenue à jour dans le fichier `mtab`.

6.4.6.5 Options courantes de montage NFS

Outre le montage d'un système de fichiers sur un hôte distant via NFS, d'autres options peuvent être spécifiées au moment du montage afin de le rendre plus commode à utiliser. Ces options peuvent être utilisées avec des commandes `mount` exécutées manuellement, à l'aide de paramètres spécifiés dans `/etc/fstab` et grâce à `autofs`.

Ci-dessous figurent les options couramment utilisées pour les montages NFS :

- `fsid=num` — Force les paramètres des descritpeurs de fichiers et des attributs de fichiers sur le réseau à prendre la valeur *num*, au lieu d'un nombre dérivé du nombre majeur et mineur du périphérique bloc présent sur le système de fichiers monté. La valeur `0` prend une signification particulière lorsqu'elle est utilisée avec NFSv4. NFSv4 a une notion de racine (aussi appelée root) pour l'ensemble du système de fichiers exporté. Le point d'export exporté avec `fsid=0` est utilisé comme cette racine (ou root).
- `hard` ou `soft` — Spécifie si le programme utilisant un fichier via une connexion NFS doit s'arrêter et attendre (`hard`) que le serveur revienne en ligne, si l'hôte fournissant le

système de fichiers exporté n'est pas disponible ou s'il doit au contraire émettre un message d'erreur (`soft`).

Si l'option `hard` est spécifiée, l'utilisateur ne peut pas mettre fin au processus attendant le rétablissement de la communication NFS à moins que l'option `intr` ne soit également spécifiée.

Si l'option `soft` est spécifiée, l'utilisateur peut ajouter une option `timeo=<value>` supplémentaire, où `<value>` spécifie le nombre de secondes devant s'écouler avant que le message d'erreur ne soit émis.

- `intr` — Autorise l'interruption des requêtes NFS si le serveur est en panne ou ne peut pas être contacté.
- `nfsvers=2` ou `nfsvers=3` — Spécifie la version spécifique du protocole NFS devant être utilisée. Cette option est utile pour des hôtes qui exécutent de multiples serveurs NFS. Si aucune version n'est précisée, NFS utilise le numéro de version le plus élevé pris en charge par le noyau et la commande `mount`. Cette option n'est pas prise en charge avec NFSv4 et ne devrait donc pas être utilisée.
- `noacl` — Désactive le traitement de toutes les LCA (ou ACL selon l'acronyme anglais). Une telle option sera peut-être nécessaire lors de l'interfaçage avec des versions plus anciennes de Red Hat Enterprise Linux, Red Hat Linux, ou de Solaris, étant donné que la technologie la plus récente en matière de listes de contrôle d'accès n'est pas compatible avec des systèmes anciens.
- `nolock` — Désactive le verrouillage de fichiers. Cette option peut être nécessaire afin de pouvoir se connecter à d'anciens serveurs NFS.
- `noexec` — Interdit l'exécution de binaires sur les systèmes de fichiers montés. Cette option est utile si votre système est en train de monter un système de fichiers non-Linux via NFS, contenant des binaires incompatibles.
- `nosuid` — Désactive les bits setuid et setgid. Cette option empêche que les utilisateurs puissent obtenir des privilèges plus étendus en exécutant un programme setuid.
- `port=num` — Spécifie la valeur numérique du port du serveur NFS. Si *`num`* équivaut à `0` (la valeur par défaut), `mount` interroge le gestionnaire de ports (ou portmapper) de l'hôte distant pour connaître le numéro de port à utiliser. Si le démon NFS de l'hôte distant n'est pas répertorié dans le gestionnaire de port, le numéro de port NFS standard utilisé est alors TCP 2049.
- `rsize=num` et `wsize=num` — Ces paramètres accélèrent la communication NFS pour les opérations de lecture (`rsize`) et d'écriture (`wsize`) en déterminant une taille de bloc de données supérieure, exprimée en octets, devant être transférée à un moment donné. Une extrême prudence est recommandée lors de la modification de ces valeurs ; certaines cartes réseau et certains noyaux Linux relativement anciens ne fonctionnent pas très bien avec des blocs d'une taille plus grande. Pour NFSv2 ou NFSv3, la valeur par défaut pour les deux paramètres est 8192. Pour NFSv4, la valeur par défaut pour les deux paramètres est 32768.
- `sec=mode` — Spécifie le type de sécurité à utiliser lors de l'authentification d'une connexion NFS.

  `sec=sys` représente le paramétrage par défaut, qui utilise des UID et GID UNIX locaux au moyen de AUTH_SYS pour authentifier des opérations NFS.

`sec=krb5` utilise Kerberos V5 au lieu des UID et GID UNIX locaux pour authentifier les utilisateurs.

`sec=krb5i` utilise Kerberos V5 pour l'authentification des utilisateurs et effectue la vérification de l'intégrité des opérations NFS à l'aide des contrôles de sommes sécurisés pour éviter la falsification de données.

`sec=krb5p` utilise Kerberos V5 pour l'authentification des utilisateurs, le contrôle de l'intégrité et le cryptage du trafic NFS afin d'empêcher le reniflage de paquets.Ce paramétrage est certes le plus sûr mais il a également le temps de gestion système le plus élevé au niveau de la performance.

- `tcp` — Indique au montage NFS d'utiliser le protocole TCP.
- `udp` — Indique au montage NFS d'utiliser le protocole UDP.

Davantage d'options sont énumérées dans les pages de manuel de `mount` et `nfs`.

6.4.6.6 La commande showmount

Cette commande permet d'interroger un hôte distant sur les services NFS qu'il offre, et notamment les volumes qu'il exporte.

showmount `-e AdresseIP_ou_NomIP` lancée à partir d'un client nous affichera la liste des ressources offertes par *sAdresseIP_ou_NomIP* (=serveur).

Sur le serveur, **showmount `-a`** nous affichera la liste des clients connectés sur chacune de nos ressources.

De même, sur le serveur, la command **showmount `-e`** affiche le liste des partages en cours.

6.5 Ressources supplémentaires

L'administration d'un serveur NFS peut se transformer en un véritable défi. Maintes options, y compris un certain nombre qui ne sont pas mentionnées dans ce chapitre, sont disponibles pour l'export ou le montage de partages NFS. Pour obtenir de plus amples informations, consultez les sources d'information mentionnées ci-dessous.

6.5.1 Documentation installée

- `/usr/share/doc/nfs-utils-<version-number>/` — Remplacez `<version-number>` par le numéro de version du paquetage NFS installé. Ce répertoire contient de nombreuses informations sur l'implémentation de NFS sous Linux, y compris diverses configurations NFS et leur impact sur les performances au niveau du transfert de fichiers.
- `man mount` — Contient une vue complète des options de montage aussi bien pour les configurations de serveur que celles de client NFS.
- `man fstab` — Donne des informations détaillées sur le format du fichier `/etc/fstab` utilisé pour monter les systèmes de fichiers au démarrage.

- `man nfs` — Fournit des informations détaillées sur l'export de systèmes de fichiers spécifiques à NFS et sur les options de montage.
- `man exports` — Montre les options couramment utilisées dans le fichier `/etc/exports` lors de l'export de systèmes de fichiers NFS.

6.5.2 Sites Web utiles

- http://nfs.sourceforge.net/ — La page d'accueil du projet NFS Linux et un bon endroit pour les mises à jour du statut du projet.
- http://www.citi.umich.edu/projects/nfsv4/linux/ — Une ressource pour NFSv4avec le noyau Linux 2.6.
- http://www.nfsv4.org — La page d'accueil du projet NFS version 4 et tous les standards associés.
- http://www.vanemery.com/Linux/NFSv4/NFSv4-no-rpcsec.html — Cette page décrit en détail NFSv4 utilisé avec Fedora Core 2 et le noyau 2.6 qu'il contient.
- http://www.nluug.nl/events/sane2000/papers/pawlowski.pdf — Un excellent document technique sur les fonctionnalités et améliorations du protocole NFS Version 4.

6.5.3 Livres sur le sujet

- *Managing NFS and NIS* de Hal Stern, Mike Eisler, et Ricardo Labiaga ; O'Reilly & Associates — Ce livre constitue un excellent guide de référence pour les nombreux exports NFS et options de montage disponibles.
- *NFS Illustrated* de Brent Callaghan ; Addison-Wesley Publishing Company — Ce livre fournit des comparaisons de NFS avec d'autres systèmes de fichiers réseau et montre, en détail, comment se déroule une communication NFS.
- *Guide d'administration système de Red Hat Enterprise Linux* ; Red Hat, Inc. — Le chapitre *Système de fichiers réseau (NFS)* explique de manière concise comment configurer les clients et serveurs NFS.
- *Guide de sécurité de Red Hat Enterprise Linux*; Red Hat, Inc. — Le chapitre *Sécurité serveur* explique les différentes manières de sécuriser NFS et autres services.

Travaux pratiques : partages NFS

Table of Contents

Première partie
Deuxième partie
Troisième partie

Première partie

Vous allez configurer un service de partage de disque pour un client Unix. Vous serez, au cours du TP, serveur pour un autre binôme puis client du serveur d'un autre binôme. Vous allez créer deux répertoires partagés qui seront accessibles par le client :

- `/tmp` sur le serveur sera accessible en lecture/écriture
- `/usr/share/doc` sur le serveur sera accessible en lecture pour le client.

Ces répertoires seront montés respectivement sur les répertoires locaux `/mnt/tempo` et `/mnt/doc`

Vous pourrez utiliser les commandes **man *`exports`***, **man *`mount`***, **man *`showmount`***, **man *`fstab`***, **man *`rpcinfo`***.

1. Créez sur le serveur le fichier `/etc/exports`, et déclarez les répertoires exportés.

   Activez le service `portmap`. Vérifiez qu'il est bien actif.

   Voici un exemple de ce que vous pouvez obtenir avec **rpcinfo `-p`** :

```
program no_version protocole  no_port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100011    1   udp    725  rquotad
    100011    2   udp    725  rquotad
    100003    2   udp   2049  nfs
    100005    1   udp   1026  mountd
    100005    1   tcp   1047  mountd
    100005    2   udp   1026  mountd
    100005    2   tcp   1047  mountd
```

2. Vérifiez sur le serveur les fichiers exportés avec la commande **showmount `-e`**

   Attention, si vous montez une arborescence sur un répertoire local, et que ce répertoire contenait des fichiers, ces derniers seront masqués le temps du montage.

3. Créez sur le client les points de montage, montez les dossiers exportés du serveur et testez les accès à partir du client.
4. Vérifiez les permissions d'accès lecture et lecture/écriture.
5. A partir du client, créez un fichier sur le `fs` (*file system*) accessible en écriture.

6. Ouvrez une autre session sur le serveur dans un autre terminal et essayez de démonter les répertoire montés. Que se passe t-il, pourquoi ?
7. Vérifiez sur le serveur les fichiers exportés avec la commande **showmount -a**.
8. Démontez les systèmes de fichiers.

Deuxième partie

Le fichier `/etc/fstab` permet de déclarer tous les points de montage.

1. Editez et modifiez le fichier sur le client afin d'inclure les systèmes de fichiers nfs exportés par le serveur. Utilisez l'exemple que vous avez dans `/etc/fstab`.
2. Rajoutez les lignes nécessaires en vous servant de l'exemple ci-dessous.

```
serveurNFS:/usr/share/doc /mnt/doc nfs  user
```

3. Vérifiez que les modifications que vous avez apportées dans le fichier `fstab` fonctionnent.
4. Supprimez l'option `user` sur les lignes que vous avez mises dans le fichier `fstab`, enregistrez. Essayez ensuite de monter l'arborescence en utilisant un compte autre que " root ". Que se passe-t-il ?
5. Restaurez l'environnement.

Troisième partie

1. Créez un utilisateur Linus sur la machine client, et donnez lui l'UID 1100. Créez un utilisateur Larry sur le serveur (UID 1100).
2. Réalisez un partage de `/tmp` sur le serveur, que le client montera dans /mnt.
3. Sur le client, Linus crée un fichier dans `/mnt`. Vérifiez son propriétaire
4. Sur le serveur, vérifiez le contenu du répertoire `/tmp`. A qui appartient le fichier crée ?
5. De façon théorique, pour quel utilisateur cela est il encore plus dangereux ? Quels sont les UIDs que vous connaissez tous à l'avance ?

   Consultez la page de man de exports (c'est un fichier de configuration, donc dans le chapitre 5 : man 5 exports, et plus largement man man).

6. Dans un environnement de grande confiance, vous voulez donner le droit à un root distant d'être reconnu comme root sur le partage. Comment faire ? Testez.
7. Interdisez ensuite seulement à root d'être confondu avec le root local, mais laissez les autres utilisateurs être eux mêmes. Testezxi. (Ceci est le comportement par défaut).
8. Sachant que l'UID de l'utilisateur nobody, qui existe certainement sur votre machine, est certainement 65534, trouvez une solution pour donner à toute personne se connectant l'identité de nobody. Créez un nouvel utilisateur BillG sur votre client, en vérifiant bien que son UID n'existe pas dans la sécurité locale du serveur.
9. Montez le répertoire partagé à partir du client, et testez la création et l'accès aux fichiers avec les utilisateurs BillG,Linus, et root. Que remarquez vous ?
10. Comment faire pour que tous les utilisateurs soient transformés en nobody ? Testez.
11. Vous devez maintenant offrir un partage de `/tmp`, mais chaque machine qui se connectera sur ce partage doit être différenciée (peu importe l'utilisateur de cette machine). On vous propose de créer sur le serveur des utilisateurs client1, client2 et

client3. Chaque machine qui se connectera se verra affublée de chacun de ces comptes (tout utilisateur de la machine client PC1 sera l'utilisateur client1 sur le serveur, tout utilisateur du pc client PC2 sera reconnu comme utilisateur client2 sur le serveur, et ainsi de suite. Créez le fichier `/etc/exports` correspondant à ces contraintes. Testez, et validez votre solution.

Section 7. Le service SAMBA

Table of Contents

Introduction
Eléments d'installation et de configuration de SAMBA
Le fichier de configuration sous Linux
Les étapes de la configuration du serveur
Première étape - Configuration du fichier smb.conf
Deuxième étape - Déclarer les ressources partagées
Fichier de configuration d'un serveur SAMBA :
Création d'utilisateurs Samba
Accès depuis un poste client Linux
Accès depuis un poste client Windows
Serveur Samba en tant que contrôleur de domaine

Abstract

Partage de ressources sous GNU/Linux pour les clients GNU/Linux ou Windows avec le protocole SMB.

7.1 Introduction

Avec SAMBA vous allez mettre en place un service de partage de disque pour des clients réseau. Ceux-ci peuvent être sous Linux ou sous Windows. Nous verrons surtout la configuration du service serveur sous Linux, et la configuration des clients sous Windows.

Samba est un produit assez populaire. Il existe de plus en plus d'outils de configuration en environnement graphique qui simplifient les tâches sur un serveur en exploitation (partage de ressources, création de comptes utilisateurs). Comme nous n'en sommes pas là, nous allons réaliser les opérations manuellement.

Vous devez savoir ce qu'est un domaine Microsoft, un groupe de travail, comment fonctionne la résolution de nom NetBIOS avec le protocole NetBIOS, ce qu'est un serveur WINS, un serveur d'authentification (contrôleur de domaine).

7.2 Eléments d'installation et de configuration de SAMBA

SAMBA est installé avec le paquet fds-network sur Kubuntu Dapper. Si vous n'utilisez pas le paquet fds-network, installez les paquets manuellement. Il ne devrait normalement pas y avoir de problèmes de dépendances.

Le paquet installe principalement samba et samba-common :

- le programme **nmbd** qui assure la résolution de nom NetBIOS et **smbd** qui assure le partage de ressource SMB/CIFS dans `/usr/sbin`,
- le script d'initialisation dans `/etc/init.d`,
- un fichier de configuration `/etc/samba/smb.conf`,

- une documentation complète dans `/usr/share/doc`,
- le service de journalisation (log) dans `/var/log/samba`,
- des outils comme smbpasswd pour la création des comptes `samba` et `nmblookup` pour vérifier le fonctionnement de la résolution de noms NetBIOS.

La commande **`dpkg-reconfigure samba`** vous demande si samba doit être lancé en mode autonome, choisissez « oui », si un fichier `/etc/samba/smbpasswd` doit être créé, choisissez également « oui ». La dernière option vous permet d'avoir une base de données de compte créée automatiquement à partir de la base de compte du fichier `/etc/passwd`.

Faites tout de suite une sauvegarde du fichier `/etc/smb.conf`.

Dans la suite du document, vous remplacerez $NOM_HOTE par le nom d'hôte de votre machine qui servira également de nom NetBIOS.

7.3 Le fichier de configuration sous Linux

Voici le fichier de configuration qui nous servira de base de travail. Il va permettre de :

- définir NomDuServeur comme serveur Samba,
- mettre en place l'authentification des utilisateurs,
- partager des disques et une imprimante pour un client Windows,
- partager le dossier personnel d'un utilisateur sous Linux comme étant son répertoire personnel sous Windows.

Le fichier de configuration comprend essentiellement deux parties :

- une partie “ générale ” qui définit le comportement général du serveur et la stratégie adoptée pour les services communs (CPD, mode d'authentification, service WINS)...
- une partie `share`, qui définit les ressources partagées et les permissions d'accès.

7.4 Les étapes de la configuration du serveur

Nous allons réaliser les opérations suivantes :

- Vérifier et valider le fichier de configuration,
- Déclarer les ressources partagées,
- Créer des comptes utilisateurs pour SAMBA.

Il n'y aura plus qu'à tester la configuration à partir d'un client.

Attention, un compte système n'est pas un compte SAMBA. Faites bien la distinction entre les deux.

7.5.1 Première étape - Configuration du fichier smb.conf

Configurer l'environnement de samba par le fichier `/etc/samba/smb.conf` et demarrez le service avec la commande **`/etc/init.d/samba start ou restart`**. Cette opération doit

être réalisée chaque fois que le fichier de configuration est modifié. Vérifiez la configuration à l'aide de la commande **`testparm | more`**.

Corrigez les erreurs éventuelles de configuration.

7.5.2 Deuxième étape - Déclarer les ressources partagées

Cette opération est réalisée dans la partie `Share Definitions` du fichier `smb.conf`. Chaque fois que vous ajoutez ou modifiez une ressource, relancez le service serveur.

7.6 Fichier de configuration d'un serveur SAMBA :

```
Exemple de fichier de configuration de Samba (pour un serveur simple) :
Vous trouverez de nombreuses autres options dans la documentation.

#======================================================================
[global]

## Browsing/Identification ###

   workgroup = workgroup   //à remplacer par le nom de votre groupe de
travail
   netbios name = NomDuServeur  //à remplacer par le nom du serveur Samba
   server string = %h server (Samba %v)
;   wins support = no
;   wins server = w.x.y.z
   dns proxy = no
;   name resolve order = lmhosts host wins bcast


#### Debugging/Accounting ####
   log file = /var/log/samba/log.%m

# Put a capping on the size of the log files (in Kb).
   max log size = 1000
   syslog = 0

# Do something sensible when Samba crashes: mail the admin a backtrace
   panic action = /usr/share/samba/panic-action %d

####### Authentication #######
   security = user   //nécessite une authentification pour accéder aux
ressources

  encrypt passwords = true
  passdb backend = tdbsam guest

   obey pam restrictions = yes
   invalid users = root
   passwd program = /usr/bin/passwd %u
   passwd       chat       =       *Enter\snew\sUNIX\spassword:*       %n\n
*Retype\snew\sUNIX\spassword:* %n\n .
   socket options = TCP_NODELAY

#========================= Share Definitions =======================

[homes]
```

```
#permet de partager le répertoire personnel de chaque utilisateur
   comment = Home Directories
   browseable = no
   writable = yes
   create mask = 0755
   directory mask = 0755

[partage]
   comment = Ressource partagée
#le répertoire /home/partage doit exister dans l'arborescence linux
   path=/home/partage
   browseable = yes
   writable = yes
   create mask = 0777
   directory mask = 0777

[printers]
   comment = All Printers
   browseable = no
   path = /tmp
   printable = yes
   public = no
   writable = no
   create mode = 0700

# Windows clients look for this share name as a source of downloadable
# printer drivers
[print$]
   comment = Printer Drivers
   path = /var/lib/samba/printers
   browseable = yes
   read only = yes
   guest ok = no
=============================================================================
```

7.7 Création d'utilisateurs Samba

Rappel: les comptes doivent déjà être créés sous linux avec la commande adduser .

Pour créer les comptes Samba, il faut utiliser la commande:

```
smbpasswd -a MonCompte MonMotdePasse
```

Cette commande ajoute le compte SAMBA `MonCompte` avec le mot de passe `MonMotDePasse`.

Il est possible ensuite dans la section "Share définitions" d'ajouter des partages accessibles seulement à certains utilisateurs par exemple pour le répertoire /home/administration :

```
[administration]

path=/home/administration
public = no
valid users = pierre  @admin
writable = yes
```

```
create mask = 0770
```

Le paramètre @admin permet de donner des droits aux membres du groupe système admin.

Le répertoire /home/administration doit être créé sous linux avec les droits adéquats , par exemple:

```
mkdir /home/administration
        chown pierre:admin /home/administration
        chmod 770 /home/administration
```

Un problème à éviter:

Le compte utilisateur SAMBA dispose de moins de privilèges que le compte root. Si vous partagez un répertoire et que vous faites les manipulations sous le compte root, faites attention aux droits, car si root est propriétaire (chmod 700), le client SAMBA ne pourra pas accéder au disque.Les droits SAMBA ne peuvent pas outrepasse les droits Linux, cf exemple ci-dessus pour donner des droits.

Voir **man *smbpasswd*** ou **smbpasswd --help** pour le mode d'utilisation de la commande.

Remarques :

- Les manipulations peuvent paraître fastidieuses si vous avez un grand nombre de comptes utilisateurs à créer.
- Si vous disposez de nombreux comptes d'utilisateurs sur votre système Linux, il est possible de créer sans difficulté un script qui, a partir d'un fichier texte crée les comptes systèmes et les comptes SAMBA (voir à la fin du TP Samba).

Toutes les indications sont dans la documentation de SAMBA.

7.8 Accès depuis un poste client Linux

Il est possible d'y accéder graphiquement avec le navigateur konqueror et le protocole smb en utilisant l'url suivante : smb://@ip-du-serveur/nom-partage ou smb://@ip-du-serveur/

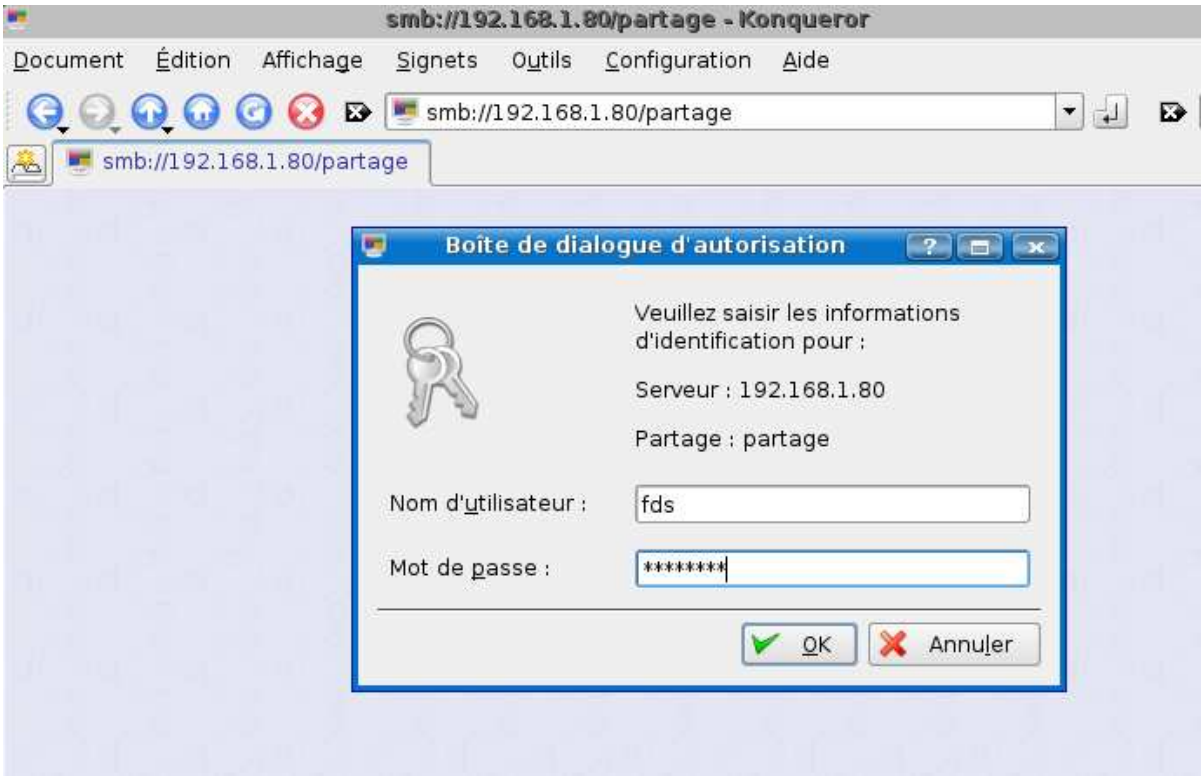


Figure 7.1. Accès à un serveur SAMBA à partir d'un client Linux par méthode graphique.

Accès en mode commande en utilisant la commande smbclient:

Pour visualiser tous les partages d'un serveur.

```
smbclient -L //@ip-du-serveur -U NomdeCompte
entrez ensuite le mot de passe Samba du compte
```

Vous pouvez également mettre le nom du serveur à la place de l'adresse IP du serveur

```
smbclient -L //NomDuServeur -U NomdeCompte
entrez ensuite le mot de passe Samba du compte
```

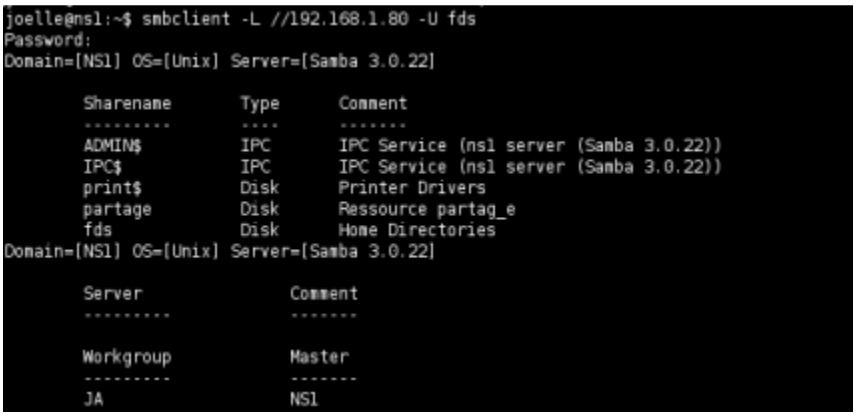


Figure 7.2. Accès à un serveur SAMBA à partir d'un client Linux par commande manuelle.

Cette autre commande vous permettra une fois connecté sur votre répertoire partagé d'envoyer ou de recevoir des fichiers avec un shell du même type que ftp. Vous pouvez afficher la liste des commandes disponibles en tapant help au prompt smb: .

```
#smbclient //@ipduserveur/partage -U NomdeCompte
Password:
Domain=[NS1] OS={Unix] Server=[Samba 3.0.22 ]
smb: \>help
```

D'autres possibilités existent pour accéder à des ressources d'un serveur Samba à partir d'un poste linux avec des logiciels comme Smb4k, Komba2pass ou swat.

7.9 Accès depuis un poste client Windows

La configuration du client Windows ne doit pas poser de difficulté.

Le compte sur lequel vous êtes connecté sous Windows XP n'a pas d'importance.

Vérifiez la configuration du protocole TCP/IP, la machine cliente Windows doit se trouver dans le même réseau IP que le serveur Samba.

Cliquez sur "favoris réseaux", puis sur "Voir les ordinateurs de votre groupe de travail" ensuite sur "Réseaux Microsoft Windows", normalement vous allez trouver votre groupe de travail (ici "ja"):

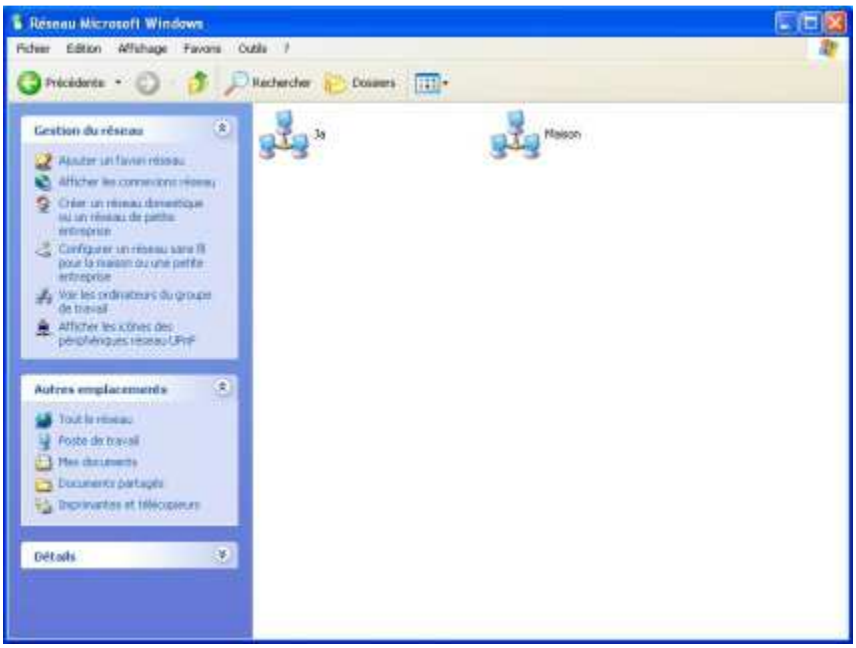


Figure 7.3. Accès à votre groupe de travail à partir d'un client Windows.

En cliquant sur votre groupe de travail, vous voyez votre serveur Samba (ici Ns1):

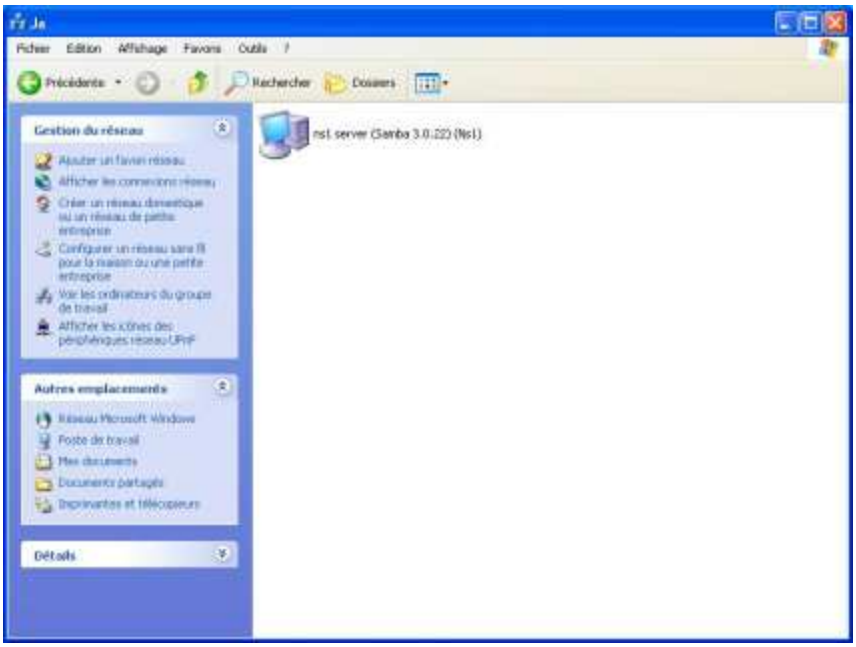


Figure 7.4. Accès à votre serveur Samba à partir d'un client Windows.

En cliquant sur le serveur, vous devrez entrer un login (ici fds) et un mot de passe valides pour accéder aux partages du serveur SAMBA :

Figure 7.5. Accès à votre groupe de travail à partir d'un client Windows.

Vous avez ensuite accès aux partages définis dans le fichier smb.conf, dans l'exemple le répertoire personnel de l'utilisateur et le répertoire partage:

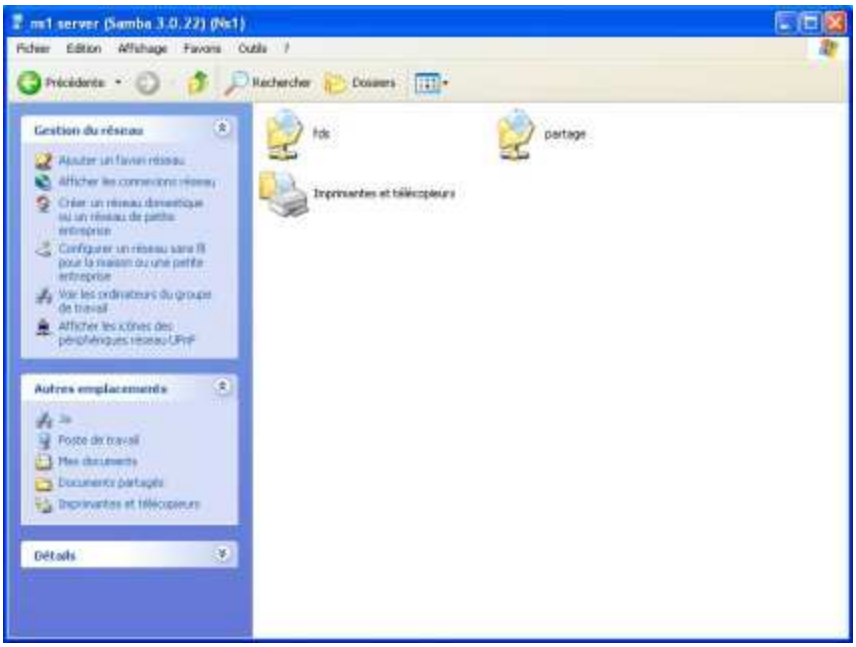


Figure 7.6. Accès à votre groupe de travail à partir d'un client Windows.

Toutes les commandes manuelles **`net use`**, **`net share`** vous permettent d'accéder aux ressources du serveur (disques partagés, imprimantes, disque personnels).

A noter: pour vous connecter sous un autre login sur votre serveur Samba à partir du poste Windows XP, vous devrez fermer et réouvrir votre session sous XP.

7.10 Serveur Samba en tant que contrôleur de domaine

Jusqu'à présent le serveur Samba ne gérait que ses propres partages mais aucune authentification de machines.

- Pour devenir contrôleur de domaine, c'est à dire l'équivalent d'un serveur Windows 2000/2003, il faut tout d'abord modifier la section globale du fichier smb.conf:

```
[global]

   workgroup = mondomaine       //à remplacer par votre nom de domaine
   netbios name = nom_serveur   // à remplacer par votre nom de serveur

# paramètres samba
   logon drive = U:
   logon home = \\serveur\%U
   logon path = \\serveur\profiles\%U
   logon script = logon.cmd  //ce script devra être créé et stocké dans le
répertoire Netlogon


   # Controle de domaine
   os level = 65
   domain logons = yes
   domain master = yes
   local master = yes
```

```
   preferred master = yes
   wins support = yes

# Base de comptes
   passdb backend = tdbsam:/var/lib/samba/mypassdb.tdb, guest
# server string is the equivalent of the NT Description field
   server string = %h server (Samba %v)

   wins support = yes

# la suite de cette section est la même que dans le premier exemple
```

- La section "Share definitions"

Il faut y créer des partages spécifiques "homes" qui contiendra les répertoires spécifiques des utilisateurs, "netlogon" qui contiendra les scripts de connexion exécutés par les machines clientes à chaque connexion d'un utilisateur, "profiles" qui permet de stocker de manière centralisée la configuration du bureau etc ... de chaque utilisateur du domaine.

```
#====================== Share Definitions ======================

[homes]
   path = /data/samba/home/%u
   comment = Home Directories
   valid users = %S
   guest ok = no
   browseable = no
   writable = yes
   create mask = 0700
   directory mask = 2700
[netlogon]
    comment = Partage Netlogon
    path = /data/samba/netlogon
    guestok = no
    readonly = yes
    browseable = no
    writable = no
    valid users = @sambausers
    create mask = 0777
    directory mask = 0777
[profiles]
    path = /data/samba/profiles
    comment = Repertoires Profiles
    guest ok = no
    writable = yes
    create mode = 0700
    browsable = no
    valid users = @sambausers
[banc2]
    path = /home/banc2
    public = no
    valid users = @banc2
    writable = yes
    create mask = 0640

[printers]
   comment = All Printers
   browseable = no
   path = /tmp
   printable = yes
```

```
   public = no
   writable = no
   create mode = 0700

[print$]
   comment = Printer Drivers
   path = /var/lib/samba/printers
   browseable = yes
   read only = yes
   guest ok = no
```

- Créer un groupe sambausers:

Ce groupe doit être créé avec le RID 513 pour être en conformité avec la terminologie Windows.

```
#groupadd -g 513 sambausers
```

Ajouter dans ce groupe les utilisateurs linux du domaine. Il faut également créér des comptes Samba pour chacun de ces utilisateurs grâce à la commande smbpasswd vue précédemment.

Ne pas oublier d'ajouter le compte "root" au groupe "sambausers", celui-ci sera le super-utilisateur samba.

```
#smbpasswd -a root MotDePasse
```

- Créer le groupe sambamachines:

Ce groupe doit être créé avec le RID 515.

```
#groupadd -g 515 sambamachines
```

Les comptes machines du domaine doivent être créés avec un $ à la fin du nom netbios,voici un extrait du fichier "/etc/passwd" avec comme exemple 2 machines labo et postexp :

```
labo$:x:1003:515::/dev/null:/dev/null
postexp$:x:1004:515::/dev/null:/dev/null
```

Créez également un compte Samba pour chaque machine:

```
#smbpasswd -a -m NomMachine  (sans le $)
```

- Création des 3 nouveaux répertoires partagés

```
#mkdir -p /data/samba/home
#mkdir -p /data/samba/netlogon
#mkdir -p /data/samba/profiles
```

Attribution des droits:

```
#chown root:sambausers /data/samba/home
#chown root:sambausers /data/samba/netlogon
#chown root:sambausers /data/samba/profiles
```

- Extrait du fichier /etc/group

```
sambausers:x:513:root,bird
sambamachines:x:515:labo$,postexp$
```

- Exemple de script exécuté au démarrage logon.cmd

```
@echo off
NET USE H: \\NomServeur\Nompartage
@echo on
```

Travaux pratiques : installation d'un serveur SAMBA

Table of Contents

Déroulement des opérations
Configuration du fichier smb.conf et démarrage des services
Création de comptes utilisateurs
Vérification de la configuration sur le serveur SAMBA
Procédure de test à partir d'un client Linux
Procédure de test à partir d'un client Windows
Configuration d'un contrôleur de domaine.
Automatisation de création de comptes.
Administration graphique

Abstract

Partage de ressources sous GNU/Linux pour les clients GNU/Linux ou Windows avec le protocole SMB.

Ce document décrit comment utiliser un serveur samba d'abord comme serveur simple puis comme serveur d'identification et d'authentification pour des clients Windows (le serveur simule un contrôleur de domaine NT4, 2000 ou 2003).

Vous utiliserez 2 postes en réseau. Le premier est sous Linux, le second sous Windows. On désire installer et configurer le service de partage de fichiers SAMBA sous Linux. Le client Windows doit permettre l'identification des utilisateurs sur le serveur en utilisant les mots de passe cryptés.

Cet atelier permet la mise en oeuvre du protocole SMB. Il permet également d'envisager la mise en place du partage de fichiers et d'imprimantes.

Déroulement des opérations

Les opérations vont se dérouler en 6 étapes :

- Configuration du fichier `/etc/smb.conf` et démarrage des services.
- Création de comptes utilisateurs.
- Création de ressources disques partagées en lecture et en lecture/écriture.
- Configuration du client Windows XP.
- Procédure de test.
- Configuration d'un contrôleur de domaine Samba.
- Procédure de test.

Configuration du fichier smb.conf et démarrage des services

Le fichier de configuration smb.conf est dans le répertoire `/etc/samba`. Faites une copie de sauvegarde de ce fichier puis ouvrez l'original avec un éditeur. Modifiez-le afin que les utilisateurs puissent accéder au répertoire `/tmp` du serveur en `rw` et à leur répertoire personnel en `rw` également.

Vous utiliserez et adapterez l'exemple donné dans la fiche de cours.

Création de comptes utilisateurs

Vous allez tout d'abord :

- créer le compte linux
- créer le compte samba.

Créez un compte système à l'aide de la commande **`adduser`**.

Pour ce compte système vous créerez un compte samba à l'aide de la commande **`smbpasswd`**.

Vérification de la configuration sur le serveur SAMBA

Démarrez le service. Vous pouvez utiliser la commande **`testparm`** pour valider la configuration du serveur. Vérifiez également la table des processus et les traces dans le fichier log.

Le fichier `diagnosis.html` de la documentation de samba, donne une procédure en 10 points pour vérifier que tout fonctionne. Localisez ce fichier, (en général dans `/usr/share/doc/samba`) ouvrez-le avec un navigateur et réalisez la procédure de test qui y est décrite.

Procédure de test à partir d'un client Linux

Si le serveur fonctionne correctement et que vous utilisez une Freeduc-Sup , vous pouvez utiliser le plugin `smb` directement à partir de konqueror.

Lancez konqueror à partir d'un autre client linux et utilisez l'url suivante : **`smb://@IP_Du_Serveur/un_partage_Samba_public`** ou **`smb://loginname@Nom_Du_Serveur/`**. Vous devriez avoir une fenêtre équivalente à celle donnée ci-dessous.

Figure 1. Accès à un serveur SAMBA à partir d'un client Linux

En ligne de commande (man smbmount smbclient mount), il est fortement déconseillé de mettre le mot de passe en clair, car outre le fait qu'il soit visible lors de sa saisie sur la console, il apparaitra en clair dans la liste des processus avec un simple ps. Pour un montage automatique du système de fichiers partagés dans fstab par exemple, on utilisera un fichier avec l'option du même nom et on veillera à positionner les droits minimum.

```
#mkdir -p /mnt/samba
# smbmount //@ipduserveur/partage /mnt/smbmnt -o username=fds
Password:
```

Concernant la dernière commande, après le prompt smb, vous pouvez ensuite taper help pour obtenir l'aide.

Exemple d'utilisation de tar avec les partages de fichiers samba :

```
smbclient //monpc/monpartage "" -N -Tc sauvegarde.tar users/docs
```

Procédure de test à partir d'un client Windows

La procédure de test sera réalisée à partir d'un client XP. Pour d'autres types de clients, il sera peut être nécessaire de créer des comptes d'ordinateurs, ou adapter la configuration du serveur SAMBA. Il faudra se référer à la documentation située en général dans `/usr/share/doc/samba`.

Configurez votre client Windows pour qu'il puisse faire partie de votre domaine NT (mettre une adresse IP dans le même réseau que le serveur Samba) déclaré sur votre serveur Linux.

Au démarrage du PC, vous pouvez vous authentifier avec un compte propre au poste XP.

Une fois la session ouverte vous utilisez la procédure décrite dans le cours:

- Favoris réseaux,
- Voir les ordinateur du groupe de travail,
- Réseaux Microsoft Windows
- Sélectionnez le nom de votre domaine Samba, puis cliquez sur votre serveur
- Vous devrez alors vous identifier avec un des utilisateurs définis sur votre serveur Samba.
- Vérifiez les accès en lecture / écriture sur les espaces disques partagés.

Modification de l'environnement serveur

Créez sur le serveur les espaces supplémentaires `/mnt/apps` et `/mnt/partage`. Le premier est en lecture uniquement, le deuxième en lecture / écriture. Donnez des droits à certains utilisateurs seulement. Modifiez `smb.conf`, relancez le service serveur, testez les accès.

Configuration d'un contrôleur de domaine.

Modifiez le fichier smb.conf suivant l'exemple donné dans le cours.

L'authentification sur un poste client Windows XP sera différente, vous pourrez dès l'ouverture de session entrer un compte utilisateur Samba.

Automatisation de création de comptes.

On donne, dans un fichier texte “ personnes ”, une liste de personnes. Le fichier a la structure suivante :

```
login motdepasse
```

par exemple :

```
tux1 tux1
tux2 tux2
...
```

En général un fichier d'importation n'est pas aussi simple car on peut avoir des noms comprenant des “ espaces ”. Les champs sont distingués par des séparateurs comme un point-virgule par exemple. Il faudra dans ce cas traiter différement le fichier.

Le principe est simple :

- On crée un script qui va créer automatiquement les comptes systèmes,
- Le compte est représenté par la 1ère colonne du fichier "persons"
- Le mot de passe est représenté par la 2ème colonne du fichier
- Les groupes sont préalablement créés.

Testez et vérifiez le fonctionnement du script. Modifiez le script pour qu'il crée également les comptes SAMBA.

Version pour Kubuntu

```
cat persons | while true
do
read ligne
if [ "$ligne" == "" ]
then
    echo "fin du fichier"
    break
fi
set -- $ligne
login=$1
pass=$2

if [ -d "/home/$1" ]
then
    echo "le compte $1 existe déjà"
else
    echo "création du compte $login"
    useradd -m $login -G $1 -s /bin/bash
    echo $login:$pass | chpasswd
    (echo $pass ; echo $pass) | smbpasswd -s -a $login
    chown $login:$groupe /home/$login
    chmod 711 /home/$login
    chown -R $login:$login /home/$login

done
echo "fin du script"
```

Administration graphique

Vous pouvez utiliser des outils graphiques d'administration de SAMBA comme swatSmb4k . Pour utilisez swat, décommentez la ligne dans `/etc/inetd.conf` :

```
swat stream tcp nowait root /usr/sbin/tcpd /usr/sbin/swat
```

Activez les services apache et inetd. Ouvrez le navigateur et saisissez :

```
http://localhost:901
```

Ouvrez une session avec le compte root.

Section 8. FTP

FTP (de l'anglais File Transfer Protocol) est l'un des protocoles les plus anciens et les plus utilisés que l'on trouve sur Internet de nos jours. Son but est de transférer de manière sécurisée des fichiers entre les ordinateurs hôtes d'un réseau sans que l'utilisateur ne doive se connecter directement à l'hôte distant ou ne doive savoir comment utiliser le système distant. Ce protocole permet aux utilisateurs d'accéder à des fichiers sur des systèmes distants en utilisant un ensemble standard de commandes simples.

Ce chapitre présente les éléments de base du protocole FTP ainsi que les options de configuration pour le serveur FTP primaire inclus dans Red Hat Enterprise Linux, `vsftpd`.

8.1. Protocole FTP (File Transport Protocol)

FTP utilise une architecture de serveur client pour transférer des fichiers à l'aide du protocole réseau TCP. Étant donné que FTP est un protocole relativement ancien, il utilise une authentification basée sur un nom d'utilisateur et un mot de passe non-cryptés. Telle est la raison pour laquelle ce protocole est considéré comme vulnérable au niveau de la sécurité et que son utilisation est déconseillée à moins qu'elle ne soit essentielle. Une bonne alternative à FT existe avec `sftp`, de la suite d'outils OpenSSH. Pour obtenir des informations sur la configuration d'OpenSSH, reportez-vous au chapitre intitulé *OpenSSH* du *Guide d'administration système de Red Hat Enterprise Linux*.

Toutefois, en raison de l'utilisation très répandue de FTP sur Internet, il est souvent nécessaire de partager des fichiers avec le public. Les administrateurs système devraient donc être conscients des caractéristiques uniques du protocole FTP.

8.1.1. Ports multiples, Modes multiples

Contrairement ŕ la plupart des protocoles utilisés sur Internet, FTP a besoin de multiples ports réseau afin de pouvoir fonctionner correctement. Lorsqu'une application FTP client établit une connexion avec un serveur FTP, elle ouvre le port 21 sur le serveur — appelé *port de commande*. Ce port est utilisé pour exécuter toutes les commandes destinées au serveur. Toute donnée requise du serveur est renvoyée au client via un *port de données*. Le numéro de port pour les connexions aux données et la manière selon laquelle les connexions aux données sont effectuées varient selon que le client demande les données en mode *actif* ou en mode *passif*.

Ces modes sont définis de la manière suivante :

- Mode actif

  Le mode actif représente la méthode utilisée ŕ l'origine par le protocole FTP pour transférer des données ŕ l'application cliente. Lorsqu'un transfert de données en mode actif est engendré par le client FTP, le serveur établit une connexion depuis le port 20 sur le serveur vers l'adresse IP et un port aléatoire, non-privilégié (supérieur ŕ 1024) spécifié par le client. Dans une telle situation, l'ordinateur client doit être autorisé à accepter des connexions sur tout port supérieur ŕ 1024. Avec le nombre croissant de réseaux non-sécurisés, tels que l'Internet, l'utilisation de pare-feu pour protéger les ordinateurs clients est désormais très répandue. Étant donné que ces pare-feu côté

client refusent souvent les connexions entrantes originaires de serveurs FTP en mode actif, il est recommandé d'utiliser le mode passif.

- Mode passif

  Le mode passif, tout comme le mode actif, est engendré par l'application client FTP. Lors d'une demande de données auprès du serveur, le client FTP indique qu'il souhaite accéder aux données en mode passif et le serveur fournit une adresse IP et un port aléatoire, non-privilégié (supérieur ŕ 1024) sur le serveur. Le client se connecte alors ŕ ce port sur le serveur afin de télécharger les informations demandées.

  Alors que le mode passif résout les problèmes d'interférence du pare-feu côté client avec des connexions aux données, il peut rendre plus complexe l'administration du pare-feu côté serveur. En limitant dans le fichier de configuration du serveur FTP, l'éventail des ports non-privilégiés disponibles pour des connexions passives, il est possible de restreindre le nombre de ports ouverts sur un serveur, ce qui permet également de simplifier la création de règles de pare-feu pour le serveur.

8.2. Serveurs FTP

Red Hat Enterprise Linux est fourni avec deux serveurs FTP différents :

- **Accélérateur de contenu Red Hat** — Un serveur Web basé sur le noyau qui fournit un serveur Web haute performance et des services FTP. Étant donné que le but de sa conception est à l'origine la vitesse, ses fonctionnalités sont limitées et son fonctionnement n'est possible que comme serveur FTP anonyme. Pour obtenir de plus amples informations sur la configuration et l'administration de l'**Accélérateur de contenu Red Hat**, consultez la documentation disponible en ligne à l'adresse suivante : http://www.redhat.com/docs/manuals/tux/.
- `vsftpd` — Un démon FTP sécurisé et rapide qui est le serveur FTP préféré pour Red Hat Enterprise Linux. Le reste de ce chapitre se concentre sur `vsftpd`.

8.2.1. vsftpd

Le démon `vsftpd` (acronyme de Very Secure FTP Daemon) est conçu pour être rapide, stable et surtout sécurisé du point de branchement à l'utilisateur final. Sa capacité à traiter un grand nombre de connexions de manière efficace et sécurisée explique la raison pour laquelle `vsftpd` est le seul FTP autonome distribué avec Red Hat Enterprise Linux.

Le modèle de sécurité utilisé par `vsftpd` possède trois caractéristiques essentielles, à savoir :

- *Séparation claire entre les processus privilégiés et les processus non-privilégiés* — Des processus différents traitent différentes tâches et chacun de ces derniers fonctionne avec le minimum de privilèges nécessaires pour la tâche à exécuter.
- *Les tâches demandant un degré élevé de privilèges sont traitées par des processus dotés du minimum de privilèges nécessaires* — En contrôlant les compatibilités, contenues dans la bibliothèque `libcap`, des tâches qui nécessitent généralement l'ensemble des privilèges root peuvent être exécutées de manière plus sûre depuis un processus doté de privilèges moins étendus.

- *La plupart des processus sont exécutés dans une prison `chroot`* — Autant que possible, le répertoire root des processus devient le répertoire partagé ; ce répertoire est alors considéré comme une prison `chroot`. Par exemple, si le répertoire `/var/ftp/` est le répertoire partagé primaire, `vsftpd` réassigne alors `/var/ftp/` au nouveau répertoire root, `/`. Cette situation empêche toute activité de la part de pirates malintentionnés sur les répertoires qui ne sont pas présents sous le nouveau répertoire root.

L'utilisation de ces pratiques de sécurité entraîne les conséquences suivantes sur la manière dont `vsftpd` traite les requêtes :

- *Le processus parent tourne avec le moins de privilèges requis.* — Le processus parent calcule dynamiquement le degré de privilèges dont il a besoin pour minimiser le degré de risque. Les processus enfants traitent l'interaction directe avec les clients FTP et tournent avec aussi peu de privilèges que possible.
- *Toutes les opérations nécessitant un degré élevé de privilèges sont traitées par un petit processus parent* — D'une manière semblable à Serveur HTTP Apache, `vsftpd` lance des processus enfants non-privilégiés pour le traitement des connexions entrantes. Ce faisant, le processus parent privilégié peut être aussi petit que possible et traiter relativement peu de tâches.
- *Le processus parent se méfie de toutes les requêtes provenant de processus enfants non-privilégiés* — Toute communication avec des processus enfants est reçue sur un socket et la validité de toute information provenant de processus enfants est vérifiée avant que toute opération ne soit exécutée.
- *La plupart de l'interaction avec les clients FTP est traitée par des processus enfants non-privilégiés dans une prison `chroot`* — Étant donné que ces processus enfants ne sont pas privilégiés et n'ont accès qu'au répertoire partagé, tout processus planté ne permet à un agresseur d'accéder qu'aux fichiers partagés.

8.3. Fichiers installés avec `vsftpd`

Le RPM `vsftpd` installe sur le système le démon (`/usr/sbin/vsftpd`), son fichier de configuration et tout fichier connexe, ainsi que les répertoires FTP. Ci-après figure une liste des fichiers et répertoires le plus souvent utilisés pour la configuration de `vsftpd` :

- `/etc/rc.d/init.d/vsftpd` — Le *script d'initialisation* (*initscript*) utilisé par la commande `/sbin/service` pour utiliser, arrêter ou recharger `vsftpd`.
- `/etc/pam.d/vsftpd` — Le fichier de configuration du module d'authentification enfichable ou PAM (de l'anglais Pluggable Authentication Modules) de `vsftpd`. Ce fichier définit les conditions qu'un utilisateur doit satisfaire pour pouvoir se connecter au serveur FTP.
- `/etc/vsftpd/vsftpd.conf` — Le fichier de configuration de `vsftpd`.
- `/etc/vsftpd.ftpusers` — Une liste des utilisateurs qui ne sont pas autorisés à se connecter à `vsftpd`. Par défaut, cette liste inclut entre autres les utilisateurs `root`, `bin` et `daemon`.
- `/etc/vsftpd.user_list` — Ce fichier peut être configuré de manière à refuser ou permettre l'accès aux utilisateurs faisant partie de la liste, selon que la directive `userlist_deny` a pour valeur `YES` (par défaut) ou `NO` dans `/etc/vsftpd/vsftpd.conf`. Si `/etc/vsftpd.user_list` est utilisé pour accorder

l'accès aux utilisateurs, les noms d'utilisateur *ne doivent pas* apparaître dans `/etc/vsftpd.ftpusers`.

- `/var/ftp/` — Le répertoire contenant les fichiers fournis par `vsftpd`. Il contient également le répertoire `/var/ftp/pub/` pour les utilisateurs anonymes. Les deux répertoires sont lisibles par tout un chacun, mais sont uniquement modifiables par l'utilisateur root.

8.4. Démarrage et arrêt de `vsftpd`

Le RPM `vsftpd` installe le script `/etc/rc.d/init.d/vsftpd` qui est accessible à l'aide de la commande `/sbin/service`.

Pour démarrer le serveur, connectez-vous en tant que super-utilisateur et tapez la commande suivante :

```
/sbin/service vsftpd start
```

Pour arrêter le serveur, connectez-vous en tant que super-utilisateur et tapez la commande suivante :

```
/sbin/service vsftpd stop
```

L'option `restart` représente une manière raccourcie d'arrêter et de démarrer ensuite `vsftpd`. Cette méthode représente la manière la plus efficace d'appliquer des changements apportés au niveau de la configuration, suite à la modification du fichier de configuration de `vsftpd`.

Pour redémarrer le serveur, connectez-vous en tant que super-utilisateur et tapez la commande suivante :

```
/sbin/service vsftpd restart
```

L'option `condrestart` (ou redémarrage conditionnel de l'anglais *conditional restart*) ne démarre `vsftpd` que s'il est actuellement en cours d'exécution. Cette option est utile pour les scripts car elle ne démarre pas le démon s'il n'est pas en cours d'exécution.

Pour redémarrer conditionnellement le serveur, connectez-vous en tant que super-utilisateur et tapez la commande suivante :

```
/sbin/service vsftpd condrestart
```

Par défaut, le service `vsftpd` *n'est pas lancé* automatiquement au démarrage. Pour configurer le service `vsftpd` de sorte qu'il soit amorcé lors du démarrage, utilisez un utilitaire initscript tel que `/sbin/chkconfig`, **/sbin/ntsysv** ou le programme de l'**Outil de configuration des services**. Reportez-vous au chapitre intitulé *Contrôle de l'accès aux services* du *Guide d'administration système de Red Hat Enterprise Linux* pour obtenir de plus amples informations sur ces outils.

8.4.1. Démarrage de multiples copies de `vsftpd`

Parfois, un ordinateur est utilisé pour fournir de multiples domaines FTP. Cette technique est appelée *multihoming* (aussi appelée hébergement multidomaines). Une possibilité d'effectuer du multihoming à l'aide de `vsftpd` consiste à exécuter de multiples copies du démon, chacune disposant de son propre fichier de configuration.

Pour ce faire, assignez d'abord les adresses IP appropriées aux périphériques réseau ou aux alias des périphériques réseau du système. Reportez-vous au chapitre intitulé *Configuration réseau* du *Guide d'administration système de Red Hat Enterprise Linux* pour obtenir de plus amples informations sur la configurations des périphériques réseau et des alias de périphériques réseau.

Ensuite, assurez-vous que le serveur DNS pour les domaines FTP est bien configuré pour référencer le bon ordinateur. Si le serveur DNS tourne sur Red Hat Enterprise Linux, reportez-vous au chapitre intitulé *Configuration de BIND* du *Guide d'administration système de Red Hat Enterprise Linux* afin d'obtenir des instructions sur l'utilisation de l'**Outil de configuration du service de noms de domaines** (`system-config-bind`).

Pour que `vsftpd` réponde à des requêtes sur des adresses IP, il est nécessaire que de multiples copies du démon tournent. La première copie doit être exécutée à l'aide des initscripts de `vsftpd`. Cette copie utilise le fichier de configuration standard, `/etc/vsftpd/vsftpd.conf`.

Chaque site FTP supplémentaire doit avoir un fichier de configuration portant un nom unique dans le répertoire `/etc/vsftpd/`, comme `/etc/vsftpd/vsftpd-site-2.conf`. Chaque fichier de configuration ne doit être lisible et modifiable que par le super-utilisateur. Au sein de chaque fichier de configuration relatif à chaque serveur FTP écoutant sur un réseau IPv4, la directive suivante doit être unique :

```
listen_address=N.N.N.N
```

Remplacez *`N.N.N.N`* par l'adresse IP *unique* du site FTP fourni. Si le site utilise IPv6, employez plutôt la directive `listen_address6`.

Une fois que chaque serveur supplémentaire est doté d'un fichier de configuration, le démon `vsftpd` doit être exécuté depuis une invite du shell root à l'aide de la commande suivante :

```
vsftpd /etc/vsftpd/<configuration-file> &
```

Dans la commande ci-dessus, remplacez *`<configuration-file>`* par le nom unique du fichier de configuration du serveur, tel que `/etc/vsftpd/vsftpd-site-2.conf`.

Parmi d'autres directives pouvant faire l'objet de modifications sur une base individuelle pour chaque serveur figurent :

- `anon_root`
- `local_root`
- `vsftpd_log_file`
- `xferlog_file`

Pour configurer tout serveur supplémentaire afin qu'il s'exécute automatiquement au démarrage, ajoutez la commande ci-dessus à la fin du fichier de configuration `/etc/rc.local`.

8.5. Options de configuration de `vsftpd`

Bien que `vsftpd` n'offre pas forcément le même degré de personnalisation que d'autres serveurs FTP courants, il fournit suffisamment d'options pour répondre aux besoins de la plupart des administrateurs. Étant donné que sa gamme de fonctionnalités n'est pas excessivement vaste, les erreurs pragmatiques et de configuration sont plus restreintes.

Toute configuration de `vsftpd` est traitée par son fichier de configuration, `/etc/vsftpd/vsftpd.conf`. Chaque directive apparaît sur sa propre ligne au sein du fichier et suit le format suivant :

```
<directive>=<value>
```

Pour chaque directive, remplacez d'une part *<directive>* par une directive valide et d'autre part *<value>* par une valeur valide.

Important

Dans une directive, aucun espace ne doit figurer entre la *<directive>*, le signe égal et l'élément *<value>*.

Les lignes de commentaire doivent être précédées par un signe dièse (#) et ne sont pas prises en compte par le démon.

Pour obtenir une liste complète de toutes les directives disponibles, reportez-vous à la page de manuel de `vsftpd.conf`.

Ci-dessous figure une liste des directives les plus importantes présentes dans `/etc/vsftpd/vsftpd.conf`. Toute directive ne se trouvant pas explicitement dans le fichier de configuration de `vsftpd` se voit attribuer la valeur par défaut.

8.5.1. Options pour le démon

Ci-dessous figure une liste des directives contrôlant le comportement général du démon `vsftpd`.

- `listen` — Lorsque cette option est activée, `vsftpd` est exécuté en mode autonome. Red Hat Enterprise Linux lui attribue la valeur `YES`. Cette directive ne peut pas être utilisée de concert avec la directive `listen_ipv6`.

  La valeur par défaut est `NO`.

- `listen_ipv6` — Lorsque cette option est activée, `vsftpd` est exécuté en mode autonome, mais n'écoute que l'interface de connexion (ou socket) IPv6. Cette directive ne peut pas être utilisée de concert avec la directive `listen`.

La valeur par défaut est `NO`.

- `session_support` — Lorsque cette option est activée, `vsftpd` tente de maintenir les sessions de connexion pour chaque utilisateur par le biais de modules d'authentification enfichables (ou PAM). Si l'ouverture de sessions n'est pas nécessaire, la désactivation de cette option permet à `vsftpd` de tourner avec moins de processus et avec des privilèges moindres.

  La valeur par défaut est `YES`.

8.5.2. Options de connexion et contrôles d'accès

Ci-dessous figure une liste des directives contrôlant le comportement de connexion et les mécanismes de contrôle d'accès.

- `anonymous_enable` — Lorsque cette option est activée, des utilisateurs anonymes sont autorisés à se connecter. Les noms d'utilisateurs anonymes (dits `anonymous`) et `ftp` sont acceptés.

  La valeur par défaut est `YES`.

- `banned_email_file` — Si la directive `deny_email_enable` a pour valeur `YES`, elle spécifie le fichier contenant une liste de mots de passe de messagerie anonymes pour lesquels l'accès au serveur est refusé.

  La valeur par défaut est `/etc/vsftpd.banned_emails`.

- `banner_file` — Spécifie le fichier contenant le texte affiché lorsqu'une connexion est établie avec le serveur . Cette option écrase tout texte spécifié dans la directive `ftpd_banner`.

  Il n'existe pas de valeur par défaut pour cette directive.

- `cmds_allowed` — Spécifie une liste de commandes FTP, séparées les unes des autres par des virgules, qui permises par le serveur. Toutes les autres commandes sont refusées.

  Il n'existe pas de valeur par défaut pour cette directive.

- `deny_email_enable` — Lorsque cette option est activée, tout utilisateur anonyme employant des mots de passe de messagerie spécifiés dans `/etc/vsftpd.banned_emails` se voit refuser l'accès au serveur. Le nom du fichier référencé par cette directive peut être spécifié à l'aide de la directive `banned_email_file`.

  La valeur par défaut est `NO`.

- `ftpd_banner` — Lorsque cette option est activée, la chaîne spécifiée dans cette directive est affichée lorsque qu'une connexion au serveur est établie. Cette option peut être annulé par la directive `banner_file`.

  Par défaut, `vsftpd` affiche sa bannière standard.

- `local_enable` — Lorsque cette option est activée, les utilisateurs locaux sont autorisés à se connecter au système.

  La valeur par défaut est `YES`.

- `pam_service_name` — Spécifie le nom du service PAM pour `vsftpd`.

  La valeur par défaut est `ftp`. Notez que sous Red Hat Enterprise Linux, cette valeur est `vsftpd`.

- `tcp_wrappers` — Lorsque cette option est activée, les enveloppeurs TCP sont utilisés pour accorder l'accès au serveur. De plus, si le serveur FTP est configuré sur de multiples adresses IP, l'option `VSFTPD_LOAD_CONF` peut être utilisée pour charger des fichiers de configuration différents en fonction de l'adresse IP demandée par le client.

  La valeur par défaut est `NO`. Notez que, sous Red Hat Enterprise Linux, la valeur est `YES`.

- `userlist_deny` — Lorsque cette option est utilisée de concert avec la directive `userlist_enable` et que sa valeur est `NO`, tous les utilisateurs locaux se voient refuser l'accès à moins que le nom d'utilisateur ne figure dans le fichier spécifié par la directive `userlist_file`. Étant donné que l'accès est refusé avant même que le client ne puisse saisir son mot de passe, le choix de la valeur `NO` pour cette directive empêche les utilisateurs de soumettre des mots de passe non-cryptés sur le réseau.

  La valeur par défaut est `YES`.

- `userlist_enable` — Lorsque cette option est activée, les utilisateurs mentionnés dans le fichier spécifiés par la directive `userlist_file` se voient refuser l'accès. Étant donné que l'accès est refusé avant même que le client ne puisse saisir son mot de passe, les utilisateurs n'ont pas la possibilité de soumettre des mots de passe non-cryptés sur le réseau.

  La valeur par défaut est `NO`, cependant, sous Red Hat Enterprise Linux la valeur donnée est `YES`.

- `userlist_file` — Spécifie le fichier référencé par `vsftpd` lorsque la directive `userlist_enable` est activée.

  La valeur par défaut est `/etc/vsftpd.user_list` ; cette dernière est créée durant l'installation.

- `cmds_allowed` — Spécifie une liste de commandes FTP, séparées les unes des autres par des virgules, que le serveur autorise. Toutes les autres commandes sont refusées.

  Il n'existe pas de valeur par défaut pour cette directive.

8.5.3. Options pour les utilisateurs anonymes

Ci-dessous figure une liste des directives qui contrôlent l'accès des utilisateurs anonymes au serveur. Pour utiliser ces options, la valeur de la directive `anonymous_enable` doit être `YES`.

- `anon_mkdir_write_enable` — Lorsque cette option est activée de concert avec la directive `write_enable`, des utilisateurs anonymes sont autorisés à créer de nouveaux répertoires au sein du répertoire parent qui a des permissions en écriture.

  La valeur par défaut est `NO`.

- `anon_root` — Spécifie le répertoire que `vsftpd` utilise après la connexion d'un utilisateur anonyme.

  Il n'existe pas de valeur par défaut pour cette directive.

- `anon_upload_enable` — Lorsque cette option est activée de concert avec la directive `write_enable`, des utilisateurs anonymes sont autorisés à télécharger vers le serveur des fichiers dans un répertoire parent doté de permissions en écriture.

  La valeur par défaut est `NO`.

- `anon_world_readable_only` — Lorsque cette option est activée, des utilisateurs anonymes sont autorisés à télécharger des fichiers lisibles par tout un chacun.

  La valeur par défaut est `YES`.

- `ftp_username` — Spécifie le compte de l'utilisateur local (énoncé dans `/etc/passwd`) employé pour l'utilisateur FTP anonyme. Le répertoire personnel spécifié dans `/etc/passwd` pour l'utilisateur est le répertoire root de l'utilisateur FTP anonyme.

  La valeur par défaut est `ftp`.

- `no_anon_password` — Lorsque cette option est activée, l'utilisateur anonyme ne doit pas saisir de mot de passe.

  La valeur par défaut est `NO`.

- `secure_email_list_enable` — Lorsque cette option est activée, seule une liste de mots de passe électroniques spécifiée pour les connexions anonymes est acceptée. Ce faisant, il est d'offrir une certaine sécurité à un contenu public sans avoir besoin d'utilisateurs virtuels.

Les connexions anonymes sont refusées à moins que le mot de passe fourni soit contenu dans `/etc/vsftpd.email_passwords`. Le format du fichier est un mot de passe par ligne, sans espace à la fin.

La valeur par défaut est `NO`.

8.5.4. Options pour les utilisateurs locaux

Ci-dessous figure une liste des directives caractérisant la manière selon laquelle les utilisateurs locaux ont accès au serveur. Pour utiliser ces options, la directive `local_enable` doit avoir la valeur `YES`.

- `chmod_enable` — Lorsque cette option est activée, la commande FTP `SITE CHMOD` est autorisée pour les utilisateurs locaux. Cette commande permet aux utilisateurs de changer les permissions s'appliquant aux fichiers.

  La valeur par défaut est `YES`.

- `chroot_list_enable` — Lorsque cette option est activée, les utilisateurs locaux énumérés dans le fichier qui est spécifié dans la directive `chroot_list_file`, sont placés dans une prison `chroot` dès qu'ils se connectent.

  Si cette option est activée de concert avec la directive `chroot_local_user`, les utilisateurs locaux énumérés dans le fichier qui est spécifié dans la directive `chroot_list_file` *ne sont pas* placés dans une prison `chroot` lors de la connexion.

  La valeur par défaut est `NO`.

- `chroot_list_file` — Spécifie le fichier contenant une liste des utilisateurs locaux référencés lorsque la valeur de la directive `chroot_list_enable` est `YES`.

  La valeur par défaut est `/etc/vsftpd.chroot_list`.

- `chroot_local_user` — Lorsque cette option est activée, les utilisateurs locaux opèrent dans l'environnement chrooté de leur répertoire personnel après leur connexion.

  La valeur par défaut est `NO`.

Avertissement

L'activation de l'option `chroot_local_user` crée un certain nombre de problèmes de sécurité, particulièrement pour les utilisateurs possédant les privilèges nécessaire pour télécharger sur le serveur. Elle *n'est* par conséquent pas recommandée.

- `guest_enable` — Lorsque cette option est activée, tous les utilisateurs autres que les utilisateurs anonymes sont connectés en tant que l'utilisateur invité (`guest`) qui est l'utilisateur local spécifié dans la directive `guest_username`.

La valeur par défaut est `NO`.

- `guest_username` — Spécifie le nom d'utilisateur vers lequel l'utilisateur invité (`guest`) est mappé.

  La valeur par défaut est `ftp`.

- `local_root` — Spécifie le répertoire que `vsftpd` utilise après la connexion d'un utilisateur local.

  Il n'existe pas de valeur par défaut pour cette directive.

- `local_umask` — Spécifie la valeur donnée à umask pour la création de fichiers. Notez que la valeur par défaut se présente sous la forme octale (un système numérique en base huit), qui inclut un préfixe "0". Sinon la valeur est traitée comme un entier à base 10.

  La valeur par défaut est `022`.

- `passwd_chroot_enable` — Lorsque cette option est activée de concert avec la directive `chroot_local_user`, `vsftpd` chroote les utilisateurs locaux si l'élément `/./` figure dans le champ du répertoire personnel au sein de `/etc/passwd`.

  La valeur par défaut est `NO`.

- `user_config_dir` — Spécifie le chemin vers un répertoire contenant les fichiers de configuration portant le nom des utilisateurs du système local qui renferment des paramètres spécifiques pour ces utilisateurs. Toute directive figurant dans le fichier de configuration d'un utilisateur annule celles figurant dans `/etc/vsftpd/vsftpd.conf`.

  Il n'existe pas de valeur par défaut pour cette directive.

8.5.5. Options pour les répertoires

Ci-dessous figure la liste des directives ayant un impact sur les répertoires.

- `dirlist_enable` — Lorsque cette option est activée, les utilisateurs sont autorisés à visionner les listes de répertoires.

  La valeur par défaut est `YES`.

- `dirmessage_enable` — Lorsque cette option est activée, un message apparaît chaque fois qu'un utilisateur ouvre un répertoire avec un fichier message. Ce message se trouve dans le répertoire qui est ouvert. Le nom de ce fichier est spécifié dans la directive `message_file` et part défaut prend la valeur `.message`.

  La valeur par défaut est `NO`. Notez que, sous Red Hat Enterprise Linux, la valeur est `YES`.

- `force_dot_files` — Lorsque cette option est activée, les fichiers commençant par un point (`.`) sont inclus dans les listes de répertoires, à l'exception des fichiers `.` et `..`.

  La valeur par défaut est `NO`.

- `hide_ids` — Lorsque cette option est activée, toutes les listes de répertoires font apparaître `ftp` comme l'utilisateur et le groupe de chaque fichier.

  La valeur par défaut est `NO`.

- `message_file` — Spécifie le nom du fichier message lorsque la directive `dirmessage_enable` est utilisée.

  La valeur par défaut est `.message`.

- `text_userdb_names` — Lorsque cette option est activée, des noms d'utilisateurs et noms de groupes test sont utilisés au lieu des entrées UID et GID. L'activation de cette option peut entraîner un ralentissement des performances du serveur.

  La valeur par défaut est `NO`.

- `use_localtime` — Lorsque cette option est activée, les listes de répertoires révèlent l'heure locale de l'ordinateur au lieu de l'heure GMT.

  La valeur par défaut est `NO`.

8.5.6. Options pour le transfert de fichiers

Ci-dessous figure la liste des directives ayant un impact sur les répertoires.

- `download_enable` — Lorsque cette option est activée, le téléchargement de fichiers est autorisé.

  La valeur par défaut est `YES`.

- `chown_uploads` — Lorsque cette option est activée, tous les fichiers téléléchargés vers les serveur par des utilisateurs anonymes deviennent la propriété de l'utilisateur spécifié dans la directive `chown_username`.

  La valeur par défaut est `NO`.

- `chown_username` — Spécifie la propriété de fichiers téléchargés anonymement vers le serveur si la directive `chown_uploads` est activée.

  La valeur par défaut est `root`.

- `write_enable` — Lorsque cette option est activée, les commandes FTP permettant de modifier le système de fichiers sont permises, telles que `DELE`, `RNFR` et `STOR`.

La valeur par défaut est YES.

8.5.7. Options de journalisation

Ci-dessous figure une liste des directives ayant un impact sur le comportement de journalisation de vsftpd.

- dual_log_enable — Lorsque cette option est activée de concert avec xferlog_enable, vsftpd enregistre deux fichiers simultanément : un journal compatible avec wu-ftpd dans le fichier spécifiée dans la directive xferlog_file (par défaut /var/log/xferlog) et un fichier journal vsftpd standard spécifié dans la directive vsftpd_log_file (par défaut /var/log/vsftpd.log).

  La valeur par défaut est NO.

- log_ftp_protocol — Lorsque cette option est activée de concert avec xferlog_enable et lorsque xferlog_std_format a pour valeur NO, toutes les commandes et réponses FTP sont journalisées. Cette directive est très utilise lors d'opérations de débogage.

  La valeur par défaut est NO.

- syslog_enable — Lorsque cette option est activée de concert avec xferlog_enable, toute journalisation normalement enregistrée dans le fichier journal standard vsftpd spécifié dans la directive vsftpd_log_file (par défaut /var/log/vsftpd.log) est envoyée à l'enregistreur du système sous le service FTPD.

  La valeur par défaut est NO.

- vsftpd_log_file — Spécifie le fichier journal vsftpd. Pour que ce fichier soit utilisé, xferlog_enable doit être activée et xferlog_std_format doit avoir pour valeur NO ou, si la valeur de xferlog_std_format estYES, l'activation dedual_log_enable est nécessaire. Il est important de noter ici que si syslog_enable a pour valeur YES, le journal du système est utilisé à la place du fichier spécifié dans cette directive.

  La valeur par défaut est /var/log/vsftpd.log.

- xferlog_enable — Lorsque cette commande est activée, vsftpd journalise les connexions (seulement au format vsftpd) et les informations de transfert de fichiers dans le fichier journal spécifié dans la directive vsftpd_log_file (par défaut /var/log/vsftpd.log). Si xferlog_std_format a pour valeur YES, les informations de transfert de fichiers sont journalisées mais les connexions elles ne le sont pas et le fichier spécifié dans xferlog_file (par défaut /var/log/xferlog) est utilisé à la place. Il est important de noter ici que les fichiers journaux aussi bien que les formats de journaux sont utilisés si la valeur de dual_log_enable est YES.

  La valeur par défaut est NO. Notez que, sous Red Hat Enterprise Linux, la valeur est YES.

- `xferlog_file` — Spécifie le fichier journal compatible avec `wu-ftpd`. Pour que ce fichier soit utilisé, `xferlog_enable` doit être activé et la valeur de `xferlog_std_format` doit être `YES`. Elle est également utilisée si la valeur de `dual_log_enable` est `YES`.

  La valeur par défaut est `/var/log/xferlog`.

- `xferlog_std_format` — Lorsque cette option est activée de concert avec `xferlog_enable`, seul un journal de transfert de fichiers compatible avec `wu-ftpd` est enregistré dans le fichier spécifié dans la directive `xferlog_file` (par défaut `/var/log/xferlog`). Il est important de noter ici que ce fichier journalise seulement les transferts de fichiers et n'enregistre pas les connexions au serveur.

  La valeur par défaut est `NO`. Notez que, sous Red Hat Enterprise Linux, la valeur est `YES`.

Important

Pour maintenir la compatibilité avec les fichiers journaux enregistrés par l'ancien serveur FTP `wu-ftpd`, la directive `xferlog_std_format` prend la valeur `YES` sous Red Hat Enterprise Linux. Toutefois, ce paramètre signifie que les connexions au serveur ne sont pas journalisées.

Pour journaliser les connexions au format `vsftpd` et maintenir un journal des transferts de fichiers qui est compatible avec `wu-ftpd`, donnez à `dual_log_enable` la valeur `YES`.

S'il n'est pas important de maintenir un journal des transferts de fichiers qui est compatible avec `wu-ftpd`, vous pouvez donner à `xferlog_std_format` la valeur `NO`, commenter la ligne à l'aide d'un signe dièse (`#`) ou supprimer la ligne complètement.

8.5.8. Options réseau

Ci-dessous figure une liste des directives ayant un impact sur la manière dont `vsftpd` interagit avec le réseau.

- `accept_timeout` — Spécifie la durée donnée à un client utilisant une connexion passive pour se connecter.

  La valeur par défaut est `60`.

- `anon_max_rate` — Spécifie le taux de transfert de données maximal, exprimé en octets par seconde, pour les utilisateurs anonymes.

  La valeur par défaut est `0`, ce qui ne limite pas le taux de transfert.

- `connect_from_port_20` Lorsque cette option est activée, `vsftpd` tourne avec suffisamment de privilèges pour ouvrir le port 20 sur le serveur lors des transferts de données en mode actif. La désactivation de cette option permet à `vsftpd` de tourner

avec moins de privilèges, mais cette option peut-être incompatible avec certains clients FTP.

La valeur par défaut est `NO`. Notez que, sous Red Hat Enterprise Linux, la valeur est `YES`.

- `connect_timeout` — Spécifie la durée maximale exprimée en secondes, donnée à un client utilisant un mode actif pour répondre à une connexion de données.

  La valeur par défaut est `60`.

- `data_connection_timeout` — Spécifie la durée maximale exprimée en secondes, pendant laquelle les transferts de données peuvent s'arrêter. Une fois cette durée écoulée, la connexion au client distant est fermée.

  La valeur par défaut est `300`.

- `ftp_data_port` — Spécifie le port utilisé pour les connexions actives aux données lorsque `connect_from_port_20` a pour valeur `YES`.

  La valeur par défaut est `20`.

- `idle_session_timeout` — Spécifie la durée maximale pouvant s'écouler entre des commandes depuis un client distant. Une fois cette durée écoulée, la connexion au client distant est fermée.

  La valeur par défaut est `300`.

- `listen_address` — Spécifie l'adresse IP sur laquelle `vsftpd` doit être à l'écoute de connexions réseau.

  Il n'existe pas de valeur par défaut pour cette directive.

Astuce

Si plusieurs copies de `vsftpd` tournent et servent différentes adresses IP, le fichier de configuration de chaque copie du démon `vsftpd` doit avoir une valeur différente pour cette directive.

- listen_address6 – Spécifie l'adresse IPv6 sur laquelle vsftpd doit être à l'écoute de connexions réseau lorsque listen_ipv6 a pour valeur YES.

Il n'existe pas de valeur par défaut pour cette directive.

Astuce

Si plusieurs copies de `vsftpd` tournent et servent différentes adresses IP, le fichier de configuration de chaque copie du démon `vsftpd` doit avoir une valeur différente pour cette directive.

- `listen_port` — Spécifie le port sur lequel `vsftpd` doit être à l'écoute de connexions réseau.

La valeur par défaut est `21`.

- `local_max_rate` — Spécifie le taux maximal (exprimé en octets par seconde) auquel les données sont transférées, pour les utilisateurs locaux connectés au serveur.

La valeur par défaut est `0`, ce qui ne limite pas le taux de transfert.

- `max_clients` — Spécifie le nombre maximal de clients autorisés à se connecter simultanément au serveur lorsqu'il tourne en mode autonome. Toute connexion client supplémentaire provoquerait un message d'erreur.

La valeur par défaut est `0`, ce qui ne limite pas les connexions.

- `max_per_ip` — Spécifie le nombre maximal de clients autorisés à se connecter depuis l'adresse IP source.

La valeur par défaut est `0`, ce qui ne limite pas les connexions.

- `pasv_address` — Spécifie l'adresse IP utilisée pour l'adresse IP publique du serveur aux serveurs se trouvant derrière des pare-feu NAT (Network Address Translation). Cette option permet à `vsftpd` de fournir la bonne adresse de retour pour des connexions en mode passif.

Il n'existe pas de valeur par défaut pour cette directive.

- `pasv_enable` — Lorsque cette option est activée, les connexions en mode passif ne sont pas permises.

La valeur par défaut est `YES`.

- `pasv_max_port` — Spécifie le port le plus élevé possible qui est envoyé aux clients FTP pour des connexions en mode passif. Ce paramètre est utilisé pour limiter la plage de ports afin que les règles de pare-feu soient faciles à créer.

La valeur par défaut est `0`, ce qui ne limite pas la plage des ports passifs les plus élevés. La valeur ne doit pas dépasser `65535`.

- `pasv_min_port` — Spécifie le port le plus bas possible qui est envoyé au client FTP pour des connexions en mode passif. Ce paramètre est utilisé pour limiter la plage de ports afin que les règles de pare-feu soient faciles à créer.

La valeur par défaut est `0`, ce qui ne limite pas la plage des ports passifs les plus bas. La valeur ne doit pas être inférieure à `1024`.

- `pasv_promiscuous` — Lorsque cette option est activée, les connexions aux données ne sont pas analysées pour vérifier qu'elles proviennent bien de la même adresse IP. Ce paramètre est seulement utile pour certains types de tunnellisation.

Attention

N'activez pas cette option à moins qu'elle ne soit absolument nécessaire. En effet, elle désactive une fonctionnalité de sécurité importante permettant de vérifier que les connexions en mode passif proviennent bien de la même adresse IP que la connexion de contrôle qui lance le transfert de données.

La valeur par défaut est `NO`.

- `port_enable` — Lorsque cette option est activée, les connexions en mode actif ne sont pas permises.

  La valeur par défaut est `YES`.

8.6. Ressources supplémentaires

Pour obtenir de plus amples informations sur `vsftpd`, reportez-vous aux ressources mentionnées ci-dessous.

8.6.1. Documentation installée

- Le répertoire `/usr/share/doc/vsftpd-<version-number>/` — Remplacez `<version-number>` par le numéro de la version du paquetage `vsftpd` installée sur le système. Ce répertoire contient un document `README` fournissant des informations élémentaires sur le logiciel. Le fichier `TUNING` contient des astuces de base pour régler la performance alors que le répertoire `SECURITY/` contient lui des informations sur le modèle de sécurité employé par `vsftpd`.
- Pages de manuels de `vsftpd` — Il existe un certain nombre de pages de manuel pour le démon et les fichiers de configuration. Ci-après figure une liste des pages de manuel les plus importantes.

  Application serveur

  - `man vsftpd` — Examine les options de ligne de commande disponibles pour `vsftpd`.

  Fichiers de configuration

  - `man vsftpd.conf` — Contient une liste détaillée des options disponibles au sein du fichier de configuration de `vsftpd`.
  - `man 5 hosts_access` — Examine le format et les options disponibles au sein des fichiers de configuration des enveloppeurs TCP : `hosts.allow` et `hosts.deny`.

8.6.2. Sites Web utiles

- http://vsftpd.beasts.org/ — La page du projet `vsftpd` est très utile pour trouver la documentation la plus récente et pour contacter l'auteur du logiciel.
- http://slacksite.com/other/ftp.html — Ce site Web fournit une explication concise des différences existant entre FTP en mode passif et en mode actif.
- http://war.jgaa.com/ftp/?cmd=rfc — Une liste complète de documents *RFC* (de l'anglais *Request for Comments*) en relation avec le protocole FTP.

Section 9. Eléments de cours sur le service DHCP

Table of Contents

Résumé
Rôle d'un service DHCP
    Pourquoi mettre en place un réseau TCP/IP avec des adresses IP dynamiques
    Protocole DHCP(Dynamic Host Configuration Protocol)
Fonctionnement de DHCP
    Attribution d'une adresse DHCP
    Renouvellement de bail IP
Configuration d'un serveur DHCP
Mise en oeuvre d'un client DHCP
Rôle de l'agent de relais DHCP

Abstract

Eléments de cours sur le service DHCP

9.1 Résumé

Présentation, installation et configuration d'un serveur DHCP

Mots clés : “ Serveur DHCP ”, “ Agent relais DHCP ”

Description et objectifs de la séquence

L'atelier propose

- d'installer un serveur DHCP sous Linux,
- d'installer un client DHCP sous Linux
- d'installer un client DHCP sous Windows
- de réaliser une phase de test avec les commandes winipcfg et ipconfig de Windows

Les éléments sur l'analyse de trame, notamment les trames bootp, seront retraités lors des TP sur la métrologie.

9.2 Rôle d'un service DHCP

Un serveur DHCP (*Dynamic Host Configuration Protocol*) a pour rôle de distribuer des adresses IP à des clients pour une durée déterminée.

Au lieu d'affecter manuellement à chaque hôte une adresse statique, ainsi que tous les paramètres tels que (serveur de noms, passerelle par défaut, nom du réseau), un serveur DHCP alloue à un client, un bail d'accès au réseau, pour une durée déterminée (durée du bail). Le serveur passe en paramètres au client toutes les informations dont il a besoin.

Tous les noeuds critiques du réseau (serveur de nom primaire et secondaire, passerelle par défaut) ont une adresse IP statique ; en effet, si celle-ci variait, ce processus ne serait plus réalisable.

Ce processus est mis en oeuvre quand vous ouvrez une session chez un fournisseur d'accès Internet par modem. Le fournisseur d'accès, vous alloue une adresse IP de son réseau le temps de la liaison. Cette adresse est libérée, donc de nouveau disponible, lors de la fermeture de la session.

9.2.1 Pourquoi mettre en place un réseau TCP/IP avec des adresses IP dynamiques

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

Avantages de DHCP dans l'administration d'un réseau ?

1. Le protocole DHCP offre une configuration de réseau TCP/IP fiable et simple, empêche les conflits d'adresses et permet de **contrôler** l'utilisation des adresses IP de façon **centralisée**. Ainsi, si un paramètre change au niveau du réseau, comme, par exemple l'adresse de la passerelle par défaut, il suffit de changer la valeur du paramètre au niveau du serveur DHCP, pour que toutes les stations aient une prise en compte du nouveau paramètre dès que le bail sera renouvelé. Dans le cas de l'adressage statique, il faudrait manuellement reconfigurer toutes les machines.
2. **économie d'adresse** : ce protocole est presque toujours utilisé par les fournisseurs d'accès Internet qui disposent d'un nombre d'adresses limité. Ainsi grâce à DHCP, seules les machines connectées en ligne ont une adresse IP. En effet, imaginons un fournisseur d'accès qui a plus de 1000 clients. Il lui faudrait 5 réseaux de classe C, s'il voulait donner à chaque client une adresse IP particulière. S'il se dit que chaque client utilise en moyenne un temps de connexion de 10 mn par jour, il peut s'en sortir avec une seule classe C, en attribuant, ce que l'on pourrait appeler des "jetons d'accès" en fonction des besoins des clients.
3. Les postes itinérants sont plus faciles à gérer
4. Le changement de plan d'adressage se trouve facilité par le dynamisme d'attribution.

Avec DHCP, il suffit d'attribuer une adresse au serveur. Lorsqu'un ordinateur client DHCPdemande l'accès au réseau en TCP-IP son adresse est allouée dynamiquement à l'intérieur d'une plage d'adresses définie sur le serveur .

L'administrateur de réseau contrôle le mode d'attribution des adresses IP en spécifiant une **durée de bail** qui indique combien de temps l'hôte peut utiliser une configuration IP attribuée, avant de devoir solliciter le renouvellement du bail auprès du serveur DHCP.

L'adresse IP est libérée automatiquement, à l'expiration du bail, pour un ordinateur client DHCP retiré d'un sous-réseau, et une nouvelle adresse est automatiquement définie pour ce dernier, lorsque cet ordinateur est reconnecté à un autre sous-réseau. Ni l'utilisateur ni l'administrateur de réseau n'ont besoin de fournir de nouvelles informations relatives à la

configuration. Cette fonctionnalité est non négligeable, tant pour les utilisateurs de portables fixés ou non à différentes stations d'accueil que pour les ordinateurs fréquemment déplacés.

L'inconvénient :

Le client utilise des trames de **broadcast** pour rechercher un serveur DHCP sur le réseau, cela charge le réseau. Si vous avez une entreprise avec plusieurs centaines de personnes qui ouvrent leur session le matin à 8 h ou l'après midi à 14 h, il peut s'en suivre de graves goulets d'étranglement sur le réseau. L'administrateur devra donc réfléchir sérieusement à l'organisation de son réseau.

9.2.2 Protocole DHCP(Dynamic Host Configuration Protocol)

Le protocole **DHCP** (*Dynamic Host Configuration Protocol*) (RFC 1533 1534) est une extension de BOOTP (RFC 1532), il fournit une **configuration dynamique** des adresses IP et des informations associées aux ordinateurs configurés pour l'utiliser (clients DHCP). Ainsi chaque hôte du réseau obtient une configuration IP dynamiquement au moment du démarrage, auprès du **serveur DHCP**. Le serveur DHCP lui attribuera notamment une adresse IP, un masque et éventuellement l'adresse d'une passerelle par défaut. Il peut attribuer beaucoup d'autres paramètres IP notamment en matière de noms (l'adresse des serveurs DNS, l'adresse des serveurs WINS)

9.3 Fonctionnement de DHCP

Un client DHCP est un ordinateur qui demande une adresse IP à un serveur DHCP.

Comment, alors, un client DHCP, qui utilise le protocole TCP/IP mais qui n'a pas encore obtenu d'adresse IP par le serveur, peut-il communiquer sur le réseau ?

9.3.1 Attribution d'une adresse DHCP

Lorsqu'un client DHCP initialise un accès à un réseau TCP/IP, le processus d'obtention du bail IP se déroule en 4 phases :

1 - Le client émet un message de demande de bail IP (DHCPDISCOVER) qui est envoyé sous forme d'une diffusion sur le réseau avec adresse IP source 0.0.0.0 et adresse IP destination 255.255.255.255 et adresse MAC.

2 - Les serveurs DHCP répondent en proposant une adresse IP avec une durée de bail et l'adresse IP du serveur DHCP (DHCOFFER)

3 - Le client sélectionne la première adresse IP (s'il y a plusieurs serveurs DHCP) reçue et envoie une demande d'utilisation de cette adresse au serveur DHCP (DHCPREQUEST). Son message envoyé par diffusion comporte l'identification du serveur sélectionné qui est informé que son offre a été retenue ; tous les autres serveurs DHCP retirent leur offre et les adresses proposée redeviennent disponibles.

4 - Le serveur DHCP accuse réception de la demande et accorde l'adresse en bail (DHCPACK), les autres serveurs retirent leur proposition.

Enfin le client utilise l'adresse pour se connecter au réseau.

Vous trouverez des éléments très précis sur le protocole DHCP dans les pages du manuel de Linux. (dhcp3d, `dhcpd.conf` et `dhclient.conf`).

9.3.2 Renouvellement de bail IP

Lorsqu'un client redémarre, il tente d'obtenir un bail pour la même adresse avec le serveur DHCP d'origine, en émettant un DHCPREQUEST. Si la tentative se solde par un échec, le client continue à utiliser la même adresse IP s'il lui reste du temps sur son bail.

Les clients DHCP d'un serveur DHCP Windows (NT/2000) tentent de renouveler leur bail lorsqu'ils ont atteint 50% de sa durée par un DHCPREQUEST. Si le serveur DHCP est disponible il envoie un DHCPACK avec la nouvelle durée et éventuellement les mises à jour des paramètres de configuration.

Si à 50% le bail n'a pu être renouvelé, le client tente de contacter l'ensemble des serveurs DHCP (diffusion) lorsqu'il atteint 87,5% de son bail, avec un DHCPREQUEST, les serveurs répondent soit par DHCPACK soit par DHCPNACK (adresse inutilisable, étendue désactivée...).

Lorsque le bail expire ou qu'un message DHCPNACK est reçu le client doit cesser d'utiliser l'adresse IP et et demander un nouveau bail (retour au processus de souscription). Lorsque le bail expire et que le client n'obtient pas d'autre adresse la communication TCP/IP s'interrompt.

Remarque: Si la demande n'aboutit pas et que le bail n'est pas expiré, le client continue à utiliser ses paramètres IP.

9.4 Configuration d'un serveur DHCP

Définir une plage d'adresses qui peuvent être louées à des hôtes qui en font la demande.

En général, on donne :

- Une adresse de début (la première qui sera attribuée)
- Une adresse de fin (la dernière)
- Une ou plusieurs plages d'adresses à exclure de la location (ceci permet de faire cohabiter un modèle de configuration IP dynamique avec un modèle statique)
- Un masque de sous-réseau

Tous ces éléments sont attribués pour une durée de bail à fixer. Si, au bout de cette durée, l'hôte ne sollicite pas à nouveau une adresse au serveur, cette adresse est jugée disponible pour un autre hôte.

Il est possible de connaître les baux actifs (les locations en cours), on voit alors à quelle adresse MAC est attribuée une adresse IP.

9.5 Mise en oeuvre d'un client DHCP

Les clients DHCP doivent être configurés seulement après la configuration du serveur. Etant donné qu'un ordinateur ne peut fonctionner simultanément comme client et serveur DHCP, l'ordinateur fonctionnant comme **serveur DHCP doit être configuré avec une adresse IP fixe**.

Lors de la configuration du client DHCP, il faut cocher la case « Obtenir une adresse IP depuis un serveur DHCP » dans la fenêtre des propriétés de Microsoft TCP/IP. Il n'est pas nécessaire alors de préciser une adresse IP ou un masque de sous-réseau.

Voici, par exemple, la configuration TCP/IP d'un ordinateur Windows XP qui sollicite une configuration IP auprès d'un serveur DHCP :

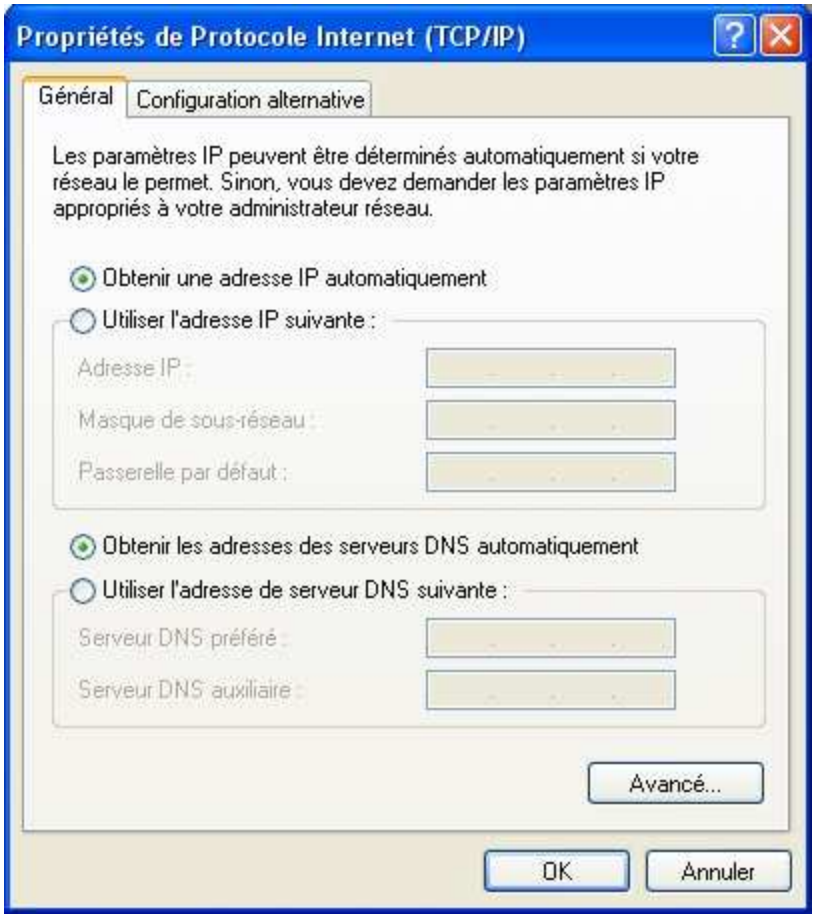


Figure 9.1. Client DHCP sous Windows XP

Les commandes IPCONFIG (Windows NT/200*x*) et **Winipcfg** (Windows 9*x*) permettent de visualiser la configuration IP complète du poste de travail :

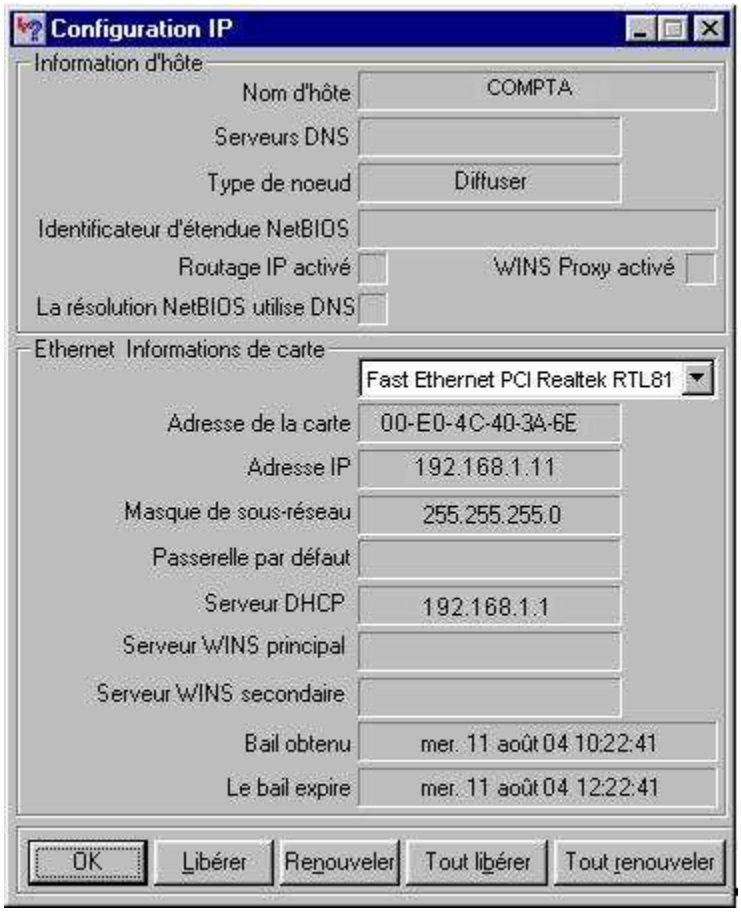


Figure 9.2. WinIPCFG sous Windows 9x

IPCONFIG

ipconfig /all : affiche les paramètres IP complets, cela va nous permettre de vérifier la bonne affectation d'adresse.

ipconfig /renew : déclenche l'envoi d'un message DHCPREQUEST vers le serveur DHCP pour obtenir des options de mise à jour

ipconfig /release : déclenche l'envoi d'un message DHCPRELEASE pour abandonner le bail. Commande utile lorsque le client change de réseau.

9.6 Rôle de l'agent de relais DHCP

Comme les clients contactent les serveurs DHCP à l'aide d'une diffusion, dans un inter-réseau, vous devrez théoriquement installer un serveur DHCP par sous-réseau. Si votre routeur prend en charge la RFC 1542, il peut faire office d'agent de relais DHCP, et ainsi relayer les diffusions de demande d'adresse IP des clients DHCP dans chaque sous-réseau.

Si votre routeur ne prend pas en charge la RFC 1542, une machine serveur peut être configurée comme agent de relais DHCP, il suffira de lui spécifier l'adresse du serveur DHCP. Les demandes des clients DHCP seront relayées vers le serveur DHCP par l'agent de relais DHCP qui transmettra les offres aux clients.

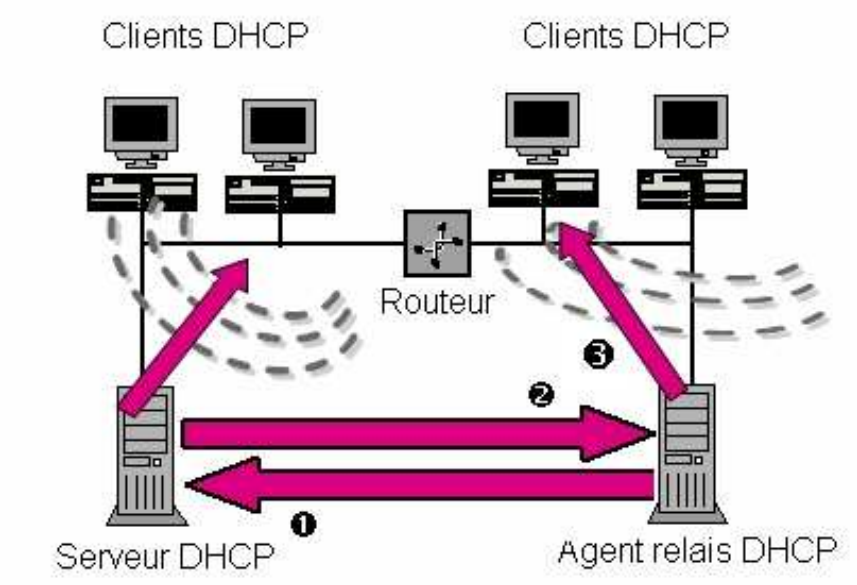


Figure 9.3. Agent de relais DHCP dans un réseau routé

Travaux pratiques : installation d'un serveur DHCP

Table of Contents

Indications pour la réalisation du TP

Installation du serveur

Configuration du serveur

Installation des clients

Procédure de test

Réalisation du TP

Indications pour la réalisation du TP

L'atelier propose

- d'installer un serveur DHCP sous Linux,
- d'installer un client DHCP sous Linux
- d'installer un client DHCP sous Windows
- de réaliser une phase de test avec les commandes winipcfg et ipconfig de Windows

Matériel nécessaire :

Deux machines en dual boot Linux / Windows en réseau.

Les éléments sur l'analyse de trame, notamment les trames bootp, seront retraités lors des TP sur la métrologie.

Installation du serveur

Les paquets sont déjà installés.

Attention : vous pouvez avoir sur votre distribution, plusieurs serveurs DHCP.

dhcpxd est conforme à la RFC 2131. Il fournit un exemple de configuration assez détaillé.

dhcp3, intègre l'inscription auprès d'un DNS Dynamique. C'est ce package que nous allons utiliser dans le TP. Par contre si vous n'avez pas de DNS dynamique sur le réseau, vous devrez mettre en entête du fichier `dhcpd.conf`, la ligne :

```
ddns-update-style none;
```

Configuration du serveur

La configuration consiste à créer 2 fichiers :

- `/etc/dhcp3/dhcpd.conf`, ce fichier sert à la configuration même du serveur (plage d'adresses, paramètres distribués),
- `/var/lib/dhcp3/dhcpd.leases`, ce fichier va servir à l'inscription des clients. Chaque client DHCP, génère l'écriture d'un enregistrement dans ce fichier. Cela permet le suivi, les statistiques de l'activité du serveur.

Le fichier de configuration dhcpd.conf

On n'abordera pas tous les paramètres. Vous trouverez un exemple de fichier commenté qui permet de réaliser cet atelier. Vous pouvez créer ce fichier avec un éditeur.

```
$>more dhcpd.conf

# ici il s'agit du réseau 192.168.0.0
subnet 192.168.0.0 netmask 255.255.255.0 {

#La plage d'adresse disponible pour les clients
range 192.168.0.10 192.168.0.20;

# Les clients auront cette adresse comme passerelle par défaut
option routers    192.168.0.254;

# Ici c'est le serveur de noms, on peut en mettre plusieurs
option domain-name-servers   192.168.0.1;

# Enfin on leur donne le nom du domaine
option domain-name   "freeduc-sup.org";

# Et l'adresse utilisée pour la diffusion
option broadcast-address 192.168.0.255;

# Le bail à une durée de 86400 s par défaut, soit 24 h
# On peut configurer les clients pour qu'ils puissent demander
# une durée de bail spécifique
default-lease-time  86400;

# On le laisse avec un maximum de 7 jours
max-lease-time 604800;

#Ici on désire réserver des adresses à des machines
group {
#use-host-decl-names indique que toutes les machines dans l'instruction «
group »
# auront comme nom, celui déclaré dans l'instruction host.
use-host-decl-names true ;

# ici définir les machines
host m1 {
hardware ethernet 00:80:23:a8:a7:24;
fixed-address 192.168.0.125;
} # End m1

host m2 {
hardware ethernet a0:81:24:a8:e8:3b;
fixed-address 192.168.0.126;
} # End m2
```

```
    } # End Group
}     # End dhcp.conf
```

Création d'un fichier d'inscription

Ce fichier doit parfois être créé, sans quoi le serveur DHCP ne pourra pas démarrer. Il suffit de créer un fichier vide. Pour cela, saisissez la commande **touch */var/lib/dhcp3/dhcpd.leases***. Le fichier est créé. Voici ce qu'il peut contenir après l'inscription du premier client :

```
[root@master /etc]# more  /var/lib/dhcp3/dhcpd.leases

lease 192.168.0.10 {

 starts 1 2002/12/14 18:33:45;

 ends 1 2002/12/14 18:34:22;

 hardware ethernet 00:40:33:2d:b5:dd;

 uid 01:00:40:33:2d:b5:dd;

 client-hostname "CHA100";

}
```

On distingue les informations suivantes : Début du bail, Fin du bail, adresse MAC du client, le nom d'hôte du client. Attention ce nom est différent du nom Netbios utilisé sur les réseaux Microsoft.

Activation du serveur

Le serveur est configuré, il n'y a plus qu'à le mettre en route. Utilisez la commande suivante pour arrêter ou activer le service : **/etc/init.d/dhcpd3 start | stop**.

Le script lance le serveur en mode daemon. Vous pouvez le lancer en avant plan avec la commande **dhcpd3 -d**. Cela permet de voir les messages et déterminer s'il y a des dysfonctionnement éventuels.

```
root@master:/etc/dhcp3# dhcpd3 -d

Internet Software Consortium DHCP Server V3.0.1rc9

Copyright 1995-2001 Internet Software Consortium.

All rights reserved.

For info, please visit http://www.isc.org/products/DHCP

Wrote 1 leases to leases file.

Listening on LPF/eth0/00:d0:59:82:2b:86/192.168.0.0/24
```

```
Sending on   LPF/eth0/00:d0:59:82:2b:86/192.168.0.0/24
Sending on   Socket/fallback/fallback-net
```

CTRL C pour arrêter.

Installation des clients

Le client sous Windows

L'installation est assez simple si vous avez déjà une carte réseau et le protocole TCP/IP installé. Utilisez les commandes suivantes: Panneau de configuration/Réseau/Protocole TCP IP/Propriétés/Onglet "adresse ip"/ Cochez Obtenir automatiquement une adresse IP

La configuration est terminée, vous pouvez relancer la machine. Le client interrogera un serveur DHCP pour qu'il lui délivre un bail (sorte d'autorisation de séjour sur le réseau) contenant au minimum une adresse Ip et le masque correspondant .

Le client sous Linux

Vous allez réaliser une configuration manuelle

Allez dans le répertoire `/etc/network`, ouvrez le fichier **interfaces**. C'est ici qu'est la configuration des cartes installées sur la machine. Remplacez `static` par `dhcp` dans la configuration de l'interface `eth0`. Mettez tous les paramètres de cette interface (address, netmask, network....) en commentaire.

La configuration de la carte est terminée, vous pouvez tester en relançant le service réseau.

Vous pouvez egalement tester dynamiquement en ligne de commande:

```
root@m1:# dhclient eth0
```

Procédure de test

Sur Windows vous allez pouvoir utiliser (selon les versions) les commandes **`IPCONFIG`** et **`Winipcfg`**.

Utilisez **`ipconfig /?`** pour voir comment utiliser la commande

Vous pouvez utiliser l'interface graphique **`winipcfg`** sous Windows 9x uniquement. Allez dans Démarrer puis Exécuter et saisissez **`winipcfg`**. Une fois la fenêtre activée vous pouvez utiliser les fonctions de libération et de renouvellement de bail. Si vous avez plusieurs cartes sur la station, la liste déroulante " Cartes Ethernet Informations " vous permet d'en sélectionner une.

Réalisation du TP

1. Installez un serveur DHCP minimal sous Linux et vérifiez le bon démarrage du service
2. Installez un client DHCP sous Linux, vérifiez le bon démarrage du service réseau et l'inscription dans le fichier dhcp.leases du serveur. Testez le renouvellement du bail. Il suffit de relancer le service réseau.
3. Installez un client DHCP sous Windows, vérifiez le bon démarrage du service réseau et l'inscription dans le fichier `dhcp.leases` du serveur. Testez le renouvellement du bail.
4. Modifiez l'étendue du serveur. Vérifiez le bon fonctionnement de la distribution d'adresses aux clients.
5. Modifiez la configuration du serveur afin qu'il distribue également l'adresse de la passerelle par défaut, l'adresse du serveur de nom. Testez la configuration.
6. Modifiez la configuration du serveur DHCP afin de réserver une adresse au client, vérifiez que le processus a bien fonctionné.

Travaux pratiques : installation d'un agent relais DHCP

Table of Contents

Routeur et agent relais DHCP (RFC 1542)
La maquette
Installation

Routeur et agent relais DHCP (RFC 1542)

Les trames arp, bootp ne traversent pas les routeurs. Sur un réseau segmenté par des routeurs il est donc impossible de servir tous les segments avec le même serveur DHCP. Il faut donc mettre un serveur DHCP sur chaque segment, ou alors utiliser un agent de relais DHCP.

Un agent relais DHCP relaie les messages DHCP échangés entre un client et un serveur DHCP situés sur des sous-réseaux différents.

Il est généralement installé sur un routeur pour pouvoir diriger les messages vers le serveur DHCP, mais ce n'est pas obligatoire. L'agent doit connaître l'adresse du serveur DHCP mais ne peut pas être lui-même client DHCP.

Serveur DHCP et agent de relais ont des adresses ip statiques. Le dialogue traverse le routeur et se fait en unicast.

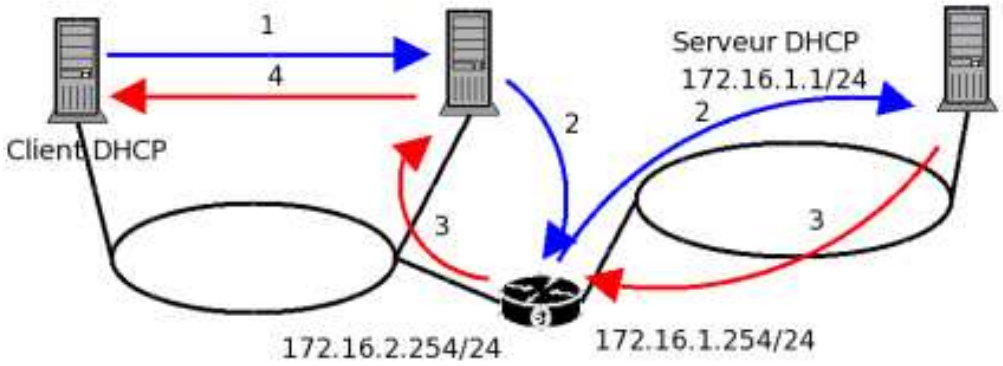


Figure 1. Dialogue client DHCP, agent de relai DHCPet serveur DHCP

Après avoir envoyé une trame de broadcast, le client DHCP dialoque avec l'agent de relai DHCP en unicast (1). L'agent demande une adresse au serveur DHCP dont il connaît l'adresse (2). Le serveur retourne à l'agent une adresse (3) qui est donnée au client DHCP par l'agent (4).

Le principal problème du service DHCP est la mise à jour des serveurs de noms d'hôtes. Bind sous Linux permet la mise à jour dynamique (RFC 2136) grâce à un serveur "updater" qui peut ajouter ou supprimer dynamiquement des enregistrements de ressources. Il faut pour corriger cela installer un serveur DNS dynamique (DDNS) qui accepte les inscriptions des clients DHCP.

La maquette

Construisez une maquette en adaptant le schéma ci-dessous :

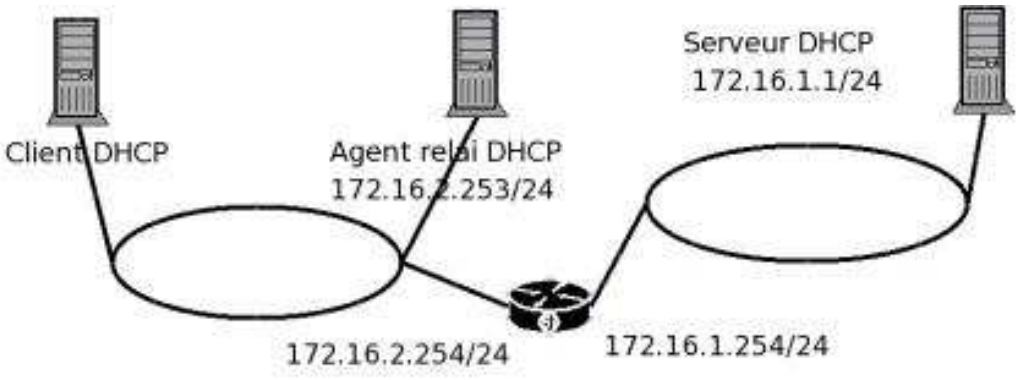


Figure 2. Maquette agent relais DHCP

Configurez les interfaces réseau de chaque machine, vérifiez que le routeur est opérationnel et que les paquets traversent bien.

Installation

Les packages : vous avez normalement les paquets sur la distribution Linux (client, serveur et agent de relais) :

Vous allez devoir installer ces produits pour les configurer et disposer de la documentation de ces produits.

Description :

Sous Linux il existe un agent relais DHCP (dhcprelay). Ce produit de l'ISC (Internet Software Consortium) permet de router des requêtes BOOTP et DHCP provenant de clients d'un réseau sur lequel il n'y a pas de serveur DHCP vers un autre segment sur lequel un serveur pourra répondre.

Mode de fonctionnement :

L'agent relais DHCP écoute les requêtes et les réponses BOOTP et DHCP. Quand une requête arrive, l'agent route la requête vers la liste de serveurs spécifiée sur la ligne de commande. Quand une réponse arrive d'un serveur, l'agent transmet la réponse (broadcast ou unicast cela dépend de la réponse) sur le segment d'où provenait la requête (broadcast) ou directement vers le client (unicast).

Ligne de commande :

```
dhcrelay3 [-p port] [-d] [-q] [-i if0 [... -i ifN ] ]server0 [ ...serverN ]
L'agent
- écoute  sur  toutes  les  interfaces  à  moins que certaines
```

```
  soient spécifiées   avec l'option -i,
- utilise, comme le  protocole  Bootp, le port 67 par défaut
  (voir /etc/services ) modifiable avec l'option -p,
- fonctionne  en avant-plan avec l'option -d (option debug),
  sinon en arrière-plan,
- n'affiche pas les informations de démarrage avec l'option -q,
- utilise  les  serveurs  spécifiés sur la ligne de commande
  server0, ...serveurN.
```

Vous allez installer le service serveur DHCP. Inspirez vous de l'exemple ci-dessous :

```
ddns-update-style none;

authoritative;

log-facility local7;
subnet 172.16.11.0 netmask 255.255.255.0 {
 range 172.16.11.2 172.16.11.253;
 option routers 172.16.11.254;
 #option domain-name-servers 192.168.90.77;
 #option domain-name "pat107.org";
  option broadcast-address 172.16.11.255;
 default-lease-time 1200;
 max-lease-time 2400;

}


subnet 172.16.12.0 netmask 255.255.255.0 {
 range 172.16.12.2 172.16.12.253;
 option routers 172.16.12.254;
 option broadcast-address 172.16.12.255;
 default-lease-time 1200;
 max-lease-time 2400;
}
```

Vous adapterez le fichier de configuration du serveur afin qu'il puisse délivrer des adresses pour les deux étendues d'adresses. Chaque segment représentant une étendue.

Vérifiez que le serveur démarre sans erreurs ni warning avec l'option "-d" (debug). Ne le lancez pas en mode daemon.

```
roo:~# dhcpd3 -d
Internet Systems Consortium DHCP Server V3.0.1rc14
Copyright 2004 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/
Wrote 0 deleted host decls to leases file.
Wrote 0 new dynamic host decls to leases file.
Wrote 1 leases to leases file.
Listening on LPF/eth0/00:08:c7:19:25:75/172.16.11.0/24
Sending on   LPF/eth0/00:08:c7:19:25:75/172.16.11.0/24
Sending on   Socket/fallback/fallback-net
```

Vérifiez égalemement que le service est actif et que le port est bien ouvert avec la commande netstat :

```
Proto Recv-Q Send-Q Adresse locale Adresse distante  Etat  PID/Program name
```

```
udp    64232      0 0.0.0.0:67     0.0.0.0:*                           2093/dhcpd3
```

Installer l'agent relais DHCP et activer le service, toujours en mode debug. La commande "dpkg-reconfigure " peut également vous permettre de configurer l'agent et indiquer à quel serveur l'agent doit passer les requêtes.

Sur la ligne de commande, on indique à quel serveur doit s'adresser l'agent :

```
root@PAT109:~# dhcrelay3 172.16.11.1 -d
Internet Systems Consortium DHCP Relay Agent V3.0.1rc14
Copyright 2004 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/
Listening on LPF/eth0/00:03:0d:08:63:bf
Sending on   LPF/eth0/00:03:0d:08:63:bf
Sending on   Socket/fallback
```

Regardez, avec la commande netstat sur quel port par défaut l'agent attend les requêtes.

Démarrez le poste client et regardez le dialogue sur les consoles du serveur et de l'agent. Le client doit obtenir une adresse ip.

Voici un extrait du dialogue sur l'agent :

```
# Transmission de la requête cliente au serveur
forwarded BOOTREQUEST for 00:0a:e4:4e:64:4e to 172.16.11.1
# Transmission de l'adresse reçu du serveur au client
forwarded BOOTREPLY for 00:0a:e4:4e:64:4e to 172.16.12.10
```

Voici un extrait du dialogue sur le serveur :

```
# Le serveur reçoit une requête de l'agent
DHCPREQUEST for 172.16.12.10 from 00:0a:e4:4e:64:4e via 172.16.12.1
# Il fournit une adresse ip et la valide
DHCPACK on 172.16.12.10 to 00:0a:e4:4e:64:4e via 172.16.12.1
```

Réalisez une capture de trame du dialogue agent/serveur sur le routeur.

Analysez la capture

Entre le relais DHCP et le serveur DHCP a-t-on utilisé des adresses de diffusion MAC ?

Comment le serveur DHCP sait-il dans quelle plage d'adresse (étendue) il doit puiser l'adresse ?

Fixez un bail à 1 mn et étudiez le mécanisme de renouvellement automatique en examinant une capture de 3 minutes.

Modifiez la configuration du serveur afin de faire de la réservation d'adresse pour le client. Vérifiez le fonctionnement.

Une fois que tout fonctionne, activez tous les services en mode daemon et vérifiez le fonctionnement de la maquette.

Section 10 Installation d'un serveur DNS

La résolution de noms - Fiche de cours

Table of Contents

Description et objectifs de la séquence
Qu'est ce que le service de résolution de noms de domaine
Présentation des concepts
- Notion de domaine, de zone et de délégation
- le domaine in-addr.arpa
- Fichiers, structure et contenus
- Principaux types d'enregistrements
- Structure des enregistrements
- La délégation
- Serveur primaire et serveur secondaire
- Le cache

Installation et configuration d'un serveur DNS
- Fichiers déjà installés
- rndc, le fichier de configuration, le fichier de clé
- Procédure de configuration du serveur
- Configurer les fichiers
- Configuration du DNS manuellement
- Le fichier `named.conf`
- Le fichier db.foo.org
- Le fichier db.foo.org.rev

Compléments pratiques
- Démarrer ou arrêter le service
- Finaliser la configuration
- Procédure de configuration des clients
- Avec windows
- Avec GNU/Linux

Procédure de tests
- Vérifier la résolution de noms :

Dépannage et outils
- Les erreurs de chargement de bind
- nslookup, dig
- Le cache du DNS
- Les journaux

Remarques
Annexes
- Annexe 1 - extraits de fichiers de configuration
- Annexe 2 - Serveur primaire et serveur secondaire
- Annexe 3 - Mise en place d'une délégation de zone
- Annexe 3 - Outils de diagnostic et de contrôle

Abstract

Ce document décrit la procédure d'installation et de configuration d'un serveur de noms sous GNU/Linux

10.1 Description et objectifs de la séquence

Avant d'installer un service quel qu'il soit, il faut s'assurer du bon fonctionnement de la résolution de noms sur le réseau. Pour cela vous avez le choix entre l'utilisation des fichiers hosts ou du service DNS. C'est ce dernier qui sera utilisé. Vous devez être familiarisé avec l'installation de GNU/Linux.

10.2 Qu'est ce que le service de résolution de noms de domaine

Le service de résolution de noms d'hôtes DNS (Domain Name Services), permet d'adresser un hôte par un nom, plutôt que de l'adresser par une adresse IP. Quelle est la structure d'un nom d'hôte?

```
              Nom_d_hôte       ou bien          Nom_d_hôte.NomDomaine
Exemple :     ns1              ou bien         ns1.foo.org
```

Le nom de domaine identifie une organisation dans l'internet, comme, par exemple, yahoo.com, wanadoo.fr, eu.org. Dans les exemples, nous utiliserons un domaine que l'on considère fictif : “ foo.org ”. Chaque organisation dispose d'un ou plusieurs réseaux. Ces réseaux sont composés de noeuds, ces noeuds (postes, serveurs, routeurs, imprimantes) pouvant être adressés.

Par exemple, la commande **`ping ns1.foo.org`**, permet d'adresser la machine qui porte le nom d'hôte `ns1`, dans le domaine (organisation) `foo.org`.

Quelle différence entre la résolution de noms d'hôtes avec un serveur DNS et les fichiers `hosts` ?

Avec les fichiers `hosts`, chaque machine dispose de sa propre base de données de noms. Sur des réseaux importants, cette base de données dupliquée n'est pas simple à maintenir.

Avec un service de résolution de noms, la base de données est localisée sur un serveur. Un client qui désire adresser un hôte regarde dans son cache local, s'il en connaît l'adresse. S'il ne la connaît pas il va interroger le serveur de noms.

Tous les grands réseaux sous TCP/IP, et internet fonctionnent (schématiquement) sur ce principe.

Avec un serveur DNS, un administrateur n'a plus qu'une seule base de données à maintenir. Il suffit qu'il indique sur chaque hôte, quelle est l'adresse de ce serveur. Ici il y a 2 cas de figures possibles :

- soit les hôtes (clients) sont des clients DHCP (Dynamic Host Configuration Protocol), cette solution est particulière et n'est pas abordée ici.

- soit les clients disposent d'une adresse IP statique. La configuration des clients est détaillée dans ce document.

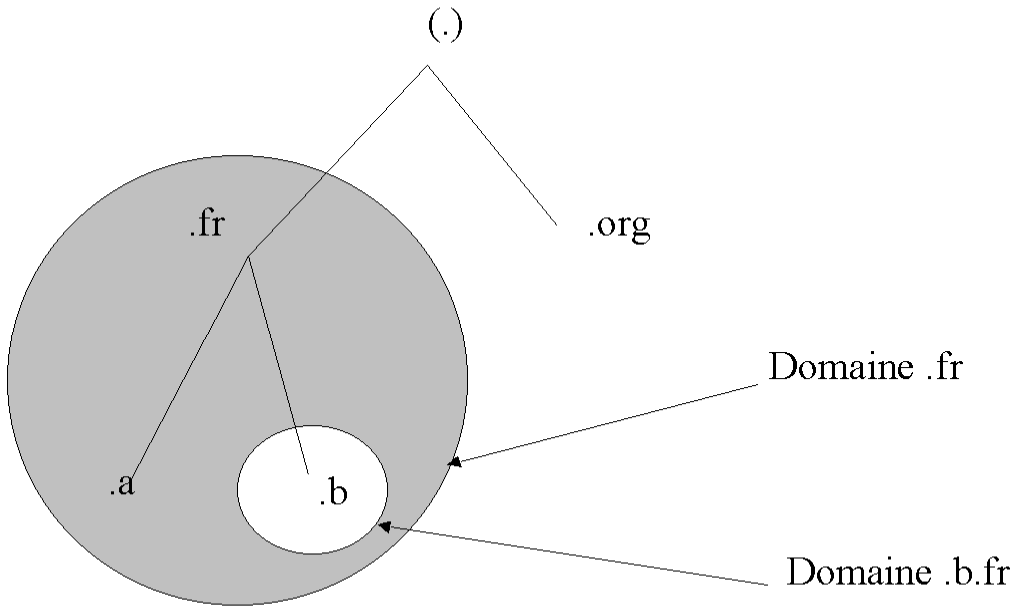
Normalement un service DNS nécessite au minimum deux serveurs afin d'assurer un minimum de redondance. Les bases de données des services sont synchronisées. La configuration d'un serveur de noms secondaire sera expliquée. Nous verrons également en TP le fonctionnement de la réplication des bases de données (bases d'enregistrements de ressources). On peut parler de bases de données réparties et synchronisées.

10.3 Présentation des concepts

10.3.1 Notion de domaine, de zone et de délégation

Un " domaine " est un sous-arbre de l'espace de nommage. Par exemple `.com` est un domaine, il contient toute la partie hiérarchique inférieure de l'arbre sous jacente au noeud `.com`.

Un domaine peut être organisé en sous domaines. `.pirlouit.com` est un sous domaine du domaine `.com`. Un domaine peut être assimilé à une partie ou sous-partie de l'organisation de l'espace de nommage. Voir la diapositive sur les Domaines, zones et délégations.



Un domaine est un sous arbre de l'espace de nommage.

Figure 10.1. Les domaines

Une "zone" est une organisation logique (ou pour être plus précis, une organisation administrative) des domaines. Le rôle d'une zone est principalement de simplifier

l'administration des domaines. Le domaine ".com" peut être découpé en plusieurs zones, z1.com, z2.com...zn.com. L'administration des zones sera déléguée afin de simplifier la gestion globale du domaine. Voir la diapositive sur les zones.

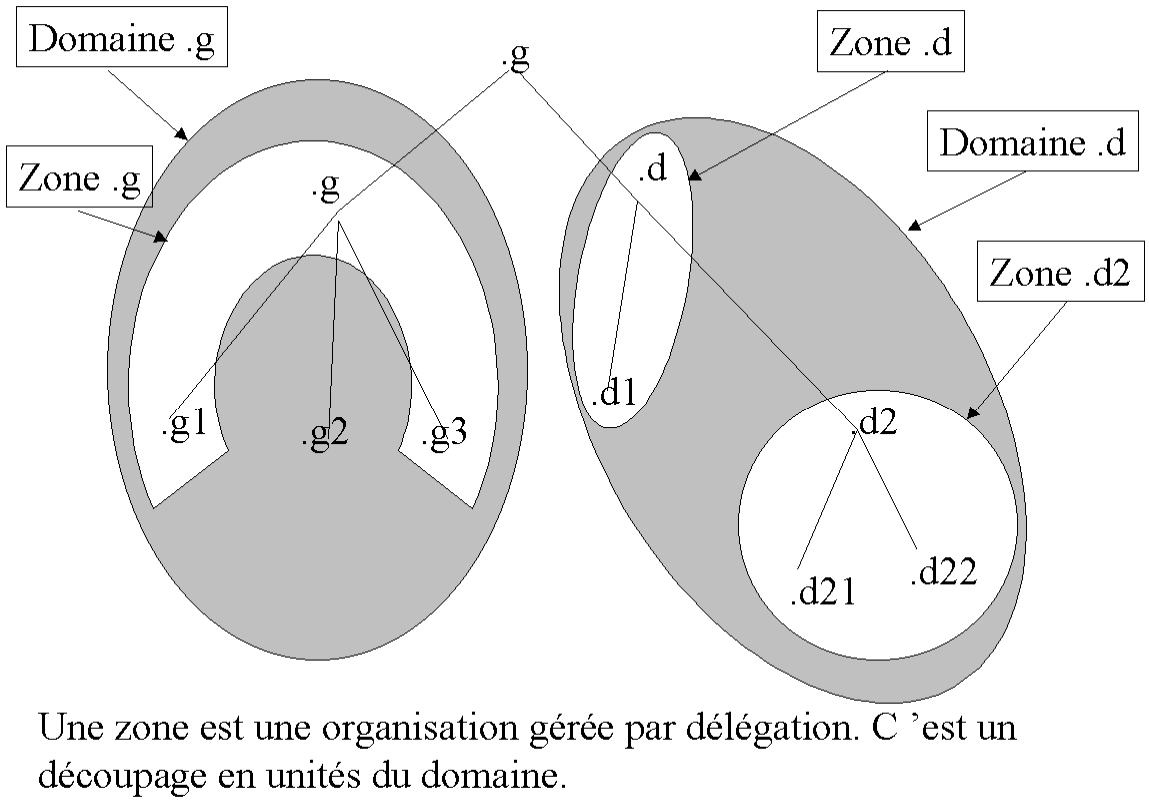


Figure 10.2. Les zones

La délégation consiste à déléguer l'administration d'une zone (ou une sous-zone) aux administrateurs de cette zone. Voir la diapositive sur la délégation.

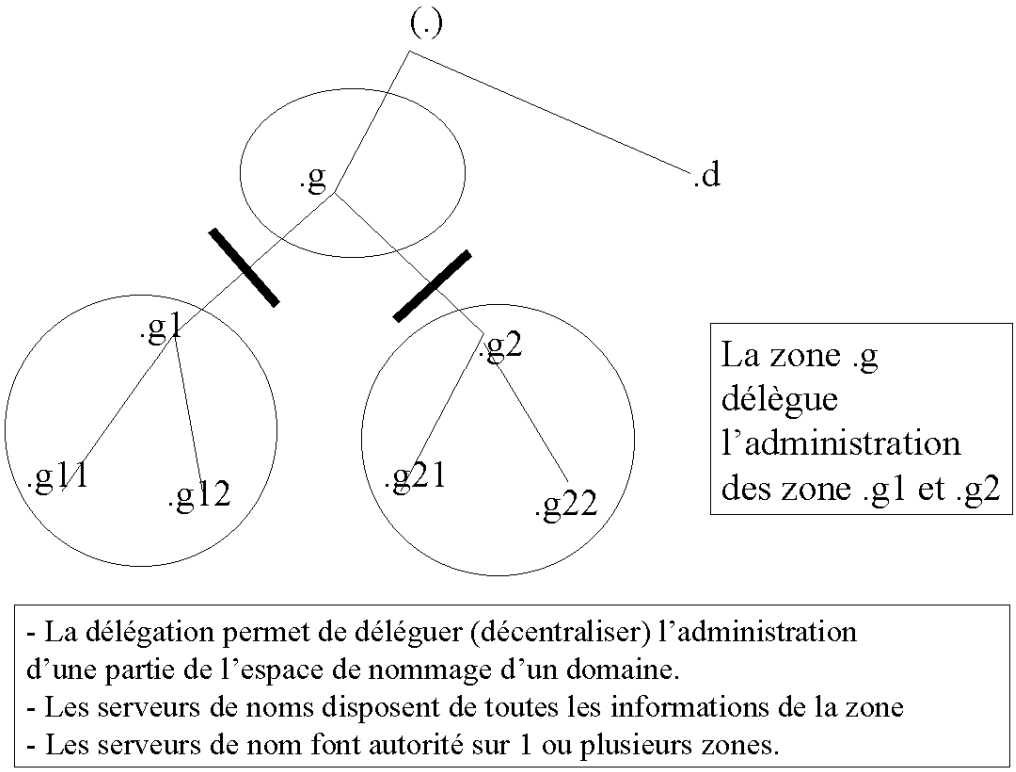


Figure 10.3. La délégation

Attention à ces quelques remarques :

- Un domaine est une organisation de l'espace de nommage. Il peut être attaché à un domaine parent, et/ou peut avoir un ou plusieurs sous-domaines enfants.
- Les zones correspondent à des organisations administratives des domaines. Un domaine peut être administré par plusieurs zones administratives, mais il est possible aussi qu'une zone serve à l'administration de plusieurs domaines. Prenons l'exemple d'un domaine "MonEntreprise.fr", membre de ".fr". Il peut être composé de trois sous-domaines France.MonEntreprise.fr, Italie.MonEntreprise.fr, Espagne.MonEntreprise.fr et de deux zones d'administration. Une en France pour les sous-domaines France.MonEntreprise.fr, Italie.MonEntreprise.fr (il n'y a pas de délégation), et une pour Espagne.MonEntreprise.fr, il y a délégation.
- L'adressage IP correspond à une organisation physique des noeuds sur un réseau IP.
- L'organisation de l'espace de nommage est complètement indépendante de l'implantation géographique d'un réseau ou de son organisation physique. L'organisation physique est gérée par des routes (tables de routage). L'espace de nommage indique pour un nom de domaine N, quelles sont les serveurs de noms qui ont autorité sur cette zone. Elles ne donnent pas la façon d'arriver à ces machines.
- Les seules machines connues au niveau de l'espace de nommage, sont les serveurs de nom "déclarés". Ces informations sont accessibles par des bases de données "whois".
- La cohérence (le service de résolution de noms) entre l'organisation de l'espace de nommage global et les organisations internes des réseaux sur internet est réalisée par les serveurs de noms.

10.3.2 Le domaine in-addr.arpa

Le principe de la résolution de noms, consiste à affecter un nom d'hôte une adresse IP. On parle de résolution de noms directe. Le processus inverse doit pouvoir également être mis en oeuvre. On parle de résolution de noms inverse ou reverse. Le processus doit fournir, pour une adresse IP, le nom correspondant. Pour cela il y a une zone particulière, in-addr.arpa, qui permet la résolution inverse d'adresse IP. Voir la diapositive sur la résolution inverse.

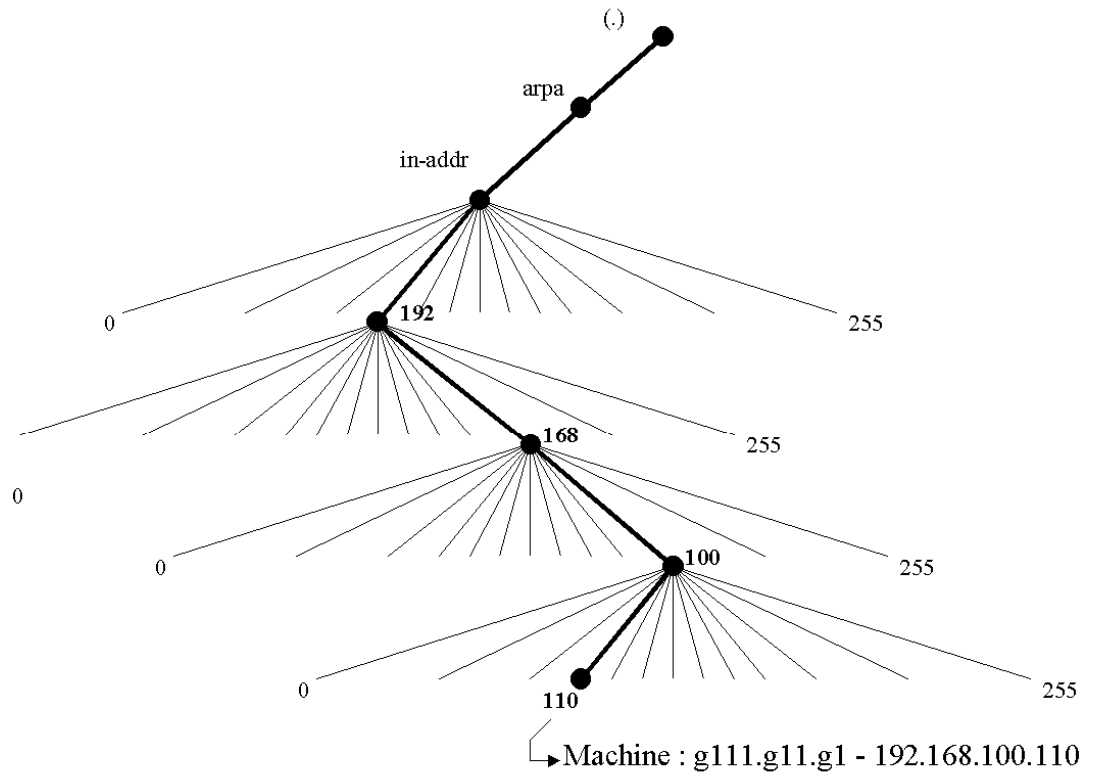


Figure 10.4. La résolution inverse

Par exemple, pour le réseau 192.168.1.0, on créera une zone inverse dans le domaine in-addr.arpa. La zone de recherche inverse dans le domaine deviendra : 1.168.192.in-addr.arpa. Cette zone devra répondre pour toutes les adresses déclarées dans la tranche 192.168.1.0 à 192.168.1.254.

On inscrira dans cette zone tous les noeuds du réseau pour lesquels on désire que la résolution inverse fonctionne. Un serveur de noms peut, pratiquement, fonctionner sans la définition de cette zone tant que le réseau n'est pas relié à l'internet. Si cela était le cas, il faudrait déclarer cette zone, sans quoi, des services comme la messagerie électronique, ne pourrait fonctionner correctement, notamment à causes des règles anti-spam. (Voir www.nic.fr)

10.3.3 Fichiers, structure et contenus

Sur linux nous allons utiliser deux types de fichiers :

- le fichier `/etc/named.conf`, qui décrit la configuration générale du serveur DNS,
- les fichiers qui contiennent les enregistrements de ressources pour la zone dans `/var/named/`. On crée, en général, un fichier pour la résolution directe d'une zone, et un fichier pour la résolution inverse.

Les enregistrements ont une structure et un rôle que nous verrons. Le daemon se nomme named, prononcer “ naime dé ”.

10.3.4 La délégation

La délégation consiste à donner l'administration d'une partie du domaine à une autre organisation. Il y a transfert de responsabilité pour l'administration d'une zone. Les serveurs de la zone auront autorité sur la zone et auront en charge la responsabilité de la résolution de noms sur la zone. Les serveurs ayant autorité sur le domaine auront des pointeurs vers les serveurs de noms ayant autorité sur chaque zone du domaine.

10.3.5 Serveur primaire et serveur secondaire

Le serveur maître (primaire) dispose d'un fichier d'information sur la zone. Le ou les serveurs esclaves (secondaires) obtiennent les informations à partir d'un serveur primaire ou d'un autre serveur esclave. Il y a " transfert de zone". Les serveurs maîtres et esclaves ont autorité sur la zone.

10.3.6 Le cache

L'organisation d'internet est assez hiérarchique. Chaque domaine dispose de ses propres serveurs de noms. Les serveurs peuvent être sur le réseau physique dont ils assurent la résolution de nom ou sur un autre réseau. Chaque zone de niveau supérieur (edu, org, fr...) dispose également de serveurs de nom de niveau supérieur. L'installation du service DNS, installe une liste de serveurs de noms de niveaux supérieurs. Cette liste permet au serveur de résoudre les noms qui sont extérieurs à sa zone. Le serveur enrichit son cache avec tous les noms résolus. Si votre réseau réseau n'est pas relié à internet, vous n'avez pas besoin d'activer cette liste.

Ce fichier est un peu particulier. Il est fourni avec les distributions. Il est utilisé par le serveur de noms à l'initialisation de sa mémoire cache. Si vos serveurs sont raccordés à internet, vous pourrez utiliser une liste officielle des serveurs de la racine (ftp.rs.internic.net).

10.4 Types de serveurs de noms

Il existe quatre types de configuration possibles pour les serveurs de noms primaires :

- *maître* — (master) Stocke les enregistrements de zone originaux faisant autorité pour un espace de nom particulier et répond aux questions d'autres serveurs de noms qui cherchent des réponses quant à cet espace de nom.
- *esclave* — (slave) Répond aux requêtes d'autres serveurs de noms concernant les espaces de nom pour lesquels il est considéré comme faisant autorité. Les serveurs de noms esclaves reçoivent leurs informations d'espace de nom des serveurs de noms maîtres.

- *cache-seulement* — (cacing-only) Offre des services de résolution de nom vers IP mais ne fait pas autorité pour quelque zone que ce soit. Les réponses pour toutes les résolutions sont placées en cache dans une base de données stockée en mémoire pour une période établie qui est spécifiée par l'enregistrement de zone importé.
- *retransmission* — (forwarding) Fait suivre des requêtes de résolution à une liste spécifique de serveurs de noms. Si aucun des serveurs de noms spécifiés ne peut effectuer la résolution, le processus s'arrête et la résolution a échoué.

Un serveur de noms appartenir à un ou plusieurs de ces types. Par exemple, un serveur de noms peut être non seulement maître pour certaines zones, esclave pour d'autres mais il peut également offrir seulement la transmission d'une résolution pour d'autres zones.

10.5 BIND en tant que serveur de noms

Le serveur de noms BIND fournit ses services de résolution de noms à l'aide du démon `/usr/sbin/named`. BIND contient également un utilitaire d'administration appelé `/usr/sbin/rndc`. De plus amples informations sur `rndc` sont dans la section `rndc`

BIND stocke ses fichiers de configuration aux emplacements suivants :

- le fichier `/etc/named.conf` — Le fichier de configuration du démon `named`.
- le répertoire `/var/named/` — Le répertoire de travail de `named` qui stocke les fichiers de zone, de statistiques et les fichiers de cache.

Les sections suivantes examinent les fichiers de configuration de manière plus détaillée.

10.5.1 /etc/named.conf

Le fichier `named.conf` est une suite de déclarations utilisant des options imbriquées qui sont placées entre accolades, `{ }`. Lorsqu'ils modifient le fichier `named.conf`, les administrateurs doivent veillez tout particulièrement à ne pas faire de fautes de syntaxe car des erreurs mineures en apparence empêcheront le démarrage du service `named`.

Avertissement

Ne modifiez pas manuellement le fichier `/etc/named.conf` ou tout autre fichier du répertoire `/var/named/` si vous utilisez l'**Outil de configuration du service de noms de domaines**. Tous les changements apportés manuellement à ces fichiers seront annulés lors d'une l'utilisation ultérieure de l'**Outil de configuration du service de noms de domaines**.

Un fichier `named.conf` typique est organisé de manière semblable à l'extrait ci-dessous :

```
<statement-1> ["<statement-1-name>"] [<statement-1-class>] {
   <option-1>;
   <option-2>;
   <option-N>;
};

<statement-2> ["<statement-2-name>"] [<statement-2-class>] {
   <option-1>;
```

```
   <option-2>;
   <option-N>;
};

<statement-N> ["<statement-N-name>"] [<statement-N-class>] {
   <option-1>;
   <option-2>;
   <option-N>;
};
```

10.5.2 Types courants de déclarations

Les types de déclarations suivants sont couramment utilisés dans `/etc/named.conf` :

10.5.2.1 Déclaration `acl`

La déclaration `acl` (de l'anglais access control list, ou déclaration de liste de contrôle d'accès) définit des groupes d'hôtes qui peuvent ensuite être autorisés ou non à accéder au serveur de noms.

Une déclaration `acl` se présente sous le format suivant :

```
acl <acl-name> {
    <match-element>;
    [<match-element>; ...]
};
```

Dans cette déclaration, remplacez *`<acl-name>`* par le nom de la liste du contrôle d'accès et remplacez *`<match-element>`* par une liste d'adresses IP séparées entre elles par un point virgule. La plupart du temps, une adresse IP individuelle ou la notation réseau de l'IP (telle que `10.0.1.0/24`) est utilisée pour identifier les adresses IP dans la déclaration `acl`.

Les listes de contrôle d'accès suivantes sont déjà définies en tant que mots-clés afin de simplifier la configuration :

- `any` — Correspond à toutes les adresses IP.
- `localhost` — Correspond à toute adresse IP utilisée par le système local.
- `localnets` — Correspond à toute adresse IP sur tout réseau auquel le système local est connecté.
- `none` — Ne correspond à aucune adresse IP.

Lorsqu'elles sont utilisées avec d'autres déclarations (telles que la déclaration `options`), les déclarations `acl` peuvent être très utiles pour éviter la mauvaise utilisation d'un serveur de noms BIND.

L'exemple ci-dessous établit deux listes de contrôle d'accès et utilise une déclaration `options` pour définir la manière dont elles seront traitées par le serveur de noms :

```
 acl black-hats {
    10.0.2.0/24;
    192.168.0.0/24;
 };
```

```
acl red-hats {
    10.0.1.0/24;
 };

options {
    blackhole { black-hats; };
    allow-query { red-hats; };
    allow-recursion { red-hats; };
 }
```

Cet exemple comporte deux listes de contôle d'accès, `black-hats` et `red-hats`. Les hôtes de la liste `black-hats` se voient dénier l'accès au serveur de noms, alors que ceux de la liste `red-hats` se voient eux donner un accès normal.

10.5.2.2 Déclaration `include`

La déclaration `include` permet à des fichiers d'être inclus dans un fichier `named.conf`. Ce faisant, des données de configurations critiques (telles que les clés, `keys`) peuvent être placées dans un fichier séparé doté de permissions restrictives.

Une déclaration `include` se présente sous le format suivant :

```
include  "<file-name>"
```

Dans cette déclaration, `<file-name>` est remplacé par le chemin d'accès absolu vers un fichier.

10.5.2.3 Déclaration `options`

La déclaration `options` définit les options globales de configuration serveur et établit des valeurs par défaut pour les autres déclarations. Cette déclaration peut être utilisée entre autres pour spécifier l'emplacement du répertoire de travail `named` ou pour déterminer les types de requêtes autorisés.

La déclaration `options` se présente sous le format suivant :

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

Dans cette déclaration, les directives `<option>` sont remplacées par une option valide.

Ci-dessous figure une liste des options couramment utilisées :

- `allow-query` — Spécifie les hôtes autorisés à interroger ce serveur de noms. Par défaut, tous les hôtes sont autorisés à interroger le serveur de noms. Il est possible

d'utiliser ici une liste de contrôle d'accès ou un ensemble d'adresses IP ou de réseaux afin de n'autoriser que des hôtes particuliers à interroger le serveur de noms.

- `allow-recursion` — Semblable à `allow-query`, cette option s'applique à des demandes récursives. Par défaut, tous les hôtes sont autorisés à effectuer des demandes récursives sur le serveur de noms.
- `blackhole` — Spécifie les hôtes qui ne sont pas autorisés à interroger le serveur de noms.
- `directory` — Change le répertoire de travail `named` pour une valeur autre que la valeur par défaut, `/var/named/`.
- `forward` — Contrôle le comportement de retransmission d'une directive `forwarders`.

  Les options suivantes sont acceptées :

  - `first` — Établit que les serveurs de noms spécifiés dans la directive `forwarders` soient interrogés avant que `named` ne tente de résoudre le nom lui-même.
  - `only` — Spécifie que `named` ne doit pas tenter d'effectuer lui-même une résolution de nom dans le cas où des demandes vers les serveurs de noms spécifiés dans la directive `forwarders` échouent.
- `forwarders` — Spécifie une liste d'adresses IP valides correspondant aux serveurs de noms vers lesquels les requêtes devraient être envoyées pour la résolution.
- `listen-on` — Spécifie l'interface réseau sur laquelle `named` prend note des requêtes. Par défaut, toutes les interfaces sont utilisées.

  De cette manière, si le serveur DNS sert également de passerelle, BIND peut être configuré de telle sorte qu'il réponde seulement aux requêtes en provenance de l'un des réseaux.

  Une directive `listen-on` peut ressembler à l'extrait ci-dessous :

```
options {
   listen-on { 10.0.1.1; };
};
```

  Dans cet exemple, seules les requêtes qui proviennent de l'interface réseau servant le réseau privé (`10.0.1.1`) sont acceptées.

- `notify` — Établit si `named` notifie les serveurs esclaves lorsqu'une zone est mise à jour. Les options suivantes sont acceptées :
  - `yes` — Notifie les serveurs esclaves.
  - `no` — Ne notifie pas les serveurs esclaves.
  - `explicit` — Notifie seulement les serveurs esclaves spécifiés dans une liste `also-notify` à l'intérieur d'une déclaration de zone.
- `pid-file` — Spécifie l'emplacement du fichier de processus ID créé par `named`.
- `root-delegation-only` — Active l'application des propriétés de délégation dans les TLD (de l'anglais top-level domains ou domaines de premier niveau) et les zones root avec une liste d'exclusion facultative. Le procédé dit de *Délégation* consiste à diviser une zone unique en multiples sous-zones. Afin de créer une zone déléguée, des éléments connus sous le nom *NS records* sont utilisés. Ces informations NameServer

(ou delegation records) précise les serveurs de noms d'autorité pour une zone particulière.

L'exemple suivant de `root-delegation-only` spécifie une liste d'exclusion de TLD desquels des réponses non-déléguées sont attendues en toute confiance :

```
options {
   root-delegation-only exclude { "ad"; "ar"; "biz"; "cr"; "cu";
"de"; "dm"; "id; "lu"; "lv"; "md"; "ms"; "museum"; "name"; "no";
"pa"; "pf"; "se"; "sr"; "to"; "tw"; "us"; "uy"; };
};
```

- `statistics-file` — Spécifie un autre emplacement des fichiers de statistiques. Par défaut, les statistiques `named` sont enregistrées dans le fichier `/var/named/named.stats`.

De nombreuses autres options sont également disponibles, dont beaucoup dépendent les unes des autres pour fonctionner correctement.

10.5.2.4 Déclaration zone

Une déclaration `zone` définit les caractéristiques d'une zone tels que l'emplacement de ses fichiers de configuration et les options spécifiques à la zone. Cette déclaration peut être utilisée pour remplacer les déclarations globales d'`options`.

Une déclaration `zone` se présente sous le format suivant :

```
zone <zone-name> <zone-class> {
     <zone-options>;
     [<zone-options>; ...]
};
```

Dans la déclaration, *<zone-name>* correspond au nom de la zone, *<zone-class>* à la classe optionnelle de la zone et *<zone-options>* représente une liste des options caractérisant la zone.

L'attribut *<zone-name>* de la déclaration de zone est particulièrement important. Il représente la valeur par défaut assignée à la directive `$ORIGIN` utilisée au sein du fichier de zone correspondant qui se trouve dans le répertoire `/var/named/`. Le démon `named` ajoute le nom de la zone à tout nom de domaine qui n'est pas pleinement qualifié, énuméré dans le fichier de zone.

Par exemple, si une déclaration `zone` définit l'espace de nom pour `example.com`, utilisez `example.com` comme *<zone-name>* afin qu'il soit placé à la fin des noms d'hôtes au sein du fichier de zone `example.com`.

Parmi les options les plus courantes de la déclaration de `zone` figurent :

- `allow-query` — Spécifie les clients qui sont autorisés à demander des informations à propos de cette zone. Par défaut toutes les requêtes d'informations sont autorisées.

- `allow-transfer` — Spécifie les serveurs esclaves qui sont autorisés à demander un transfert des informations relatvies à la zone. Par défaut toutes les requêtes de transfert sont autorisées.
- `allow-update` — Spécifie les hôtes qui sont autorisés à mettre à jour dynamiquement les informations dans leur zone. Par défaut aucune requête de mise à jour dynamique n'est autorisée.

  Soyez très prudent lorsque vous autorisez des hôtes à mettre à jour des informations concernant leur zone. N'activez cette option que si l'hôte est sans aucun doute digne de confiance. De manière générale, il est préférable de laisser un administrateur mettre à jour manuellement les enregistrements de la zone et recharger le service `named`.

- `file` — Spécifie le nom du fichier qui figure dans le répertoire de travail `named` et qui contient les données de configuration de la zone.
- `masters` — Spécifie les adresses IP à partir desquelles demander des informations sur la zone faisant autorité. Cette option ne doit être utilisée que si la zone est définie en tant que `type slave`.
- `notify` — Détermine si `named` notifie les serveurs esclaves lorsqu'une zone est mise à jour. Cette directive accepte les options suivantes :
  - `yes` — Notifie les serveurs esclaves.
  - `no` — Ne notifie pas les serveurs esclaves.
  - `explicit` — Notifie seulement les serveurs esclaves spécifiés dans une liste `also-notify` à l'intérieur d'une déclaration de zone.
- `type` — Définit le type de zone. Les types énumérés ci-dessous peuvent être utilisés.

  Ci-après figure une liste des options valides :

  - `delegation-only` — Applique l'état de délégation des zones d'infrastructure comme COM, NET ou ORG. Toute réponse reçue sans délégation explicite ou implicite est traitée en tant que `NXDOMAIN`. Cette option est seulement applicable dans des fichiers de zone TLD ou root en implémentation récursive ou de mise en cache.
  - `forward` — Retransmet toutes les requêtes d'informations concerant cette zone vers d'autres serveurs de noms
  - `hint` — Représente un type spécial de zone utilisé pour diriger des transactions vers les serveurs de noms racines qui résolvent des requêtes lorsqu'une zone n'est pas connue autrement. Aucune configuration autre que la valeur par défaut n'est nécessaire avec une zone `hint`.
  - `master` — Désigne le serveur de noms faisant autorité pour cette zone. Une zone devrait être configurée comme maître (`master`) si les fichiers de configuration de la zone se trouvent sur le système.
  - `slave` — Désigne le serveur de noms comme serveur esclave (slave) pour cette zone. Cette option spécifie également l'adresse IP du serveur de noms maître pour cette zone.
- `zone-statistics` — Configure `named` pour qu'il conserve des statistiques concernant cette zone en les écrivant soit dans l'emplacement par défaut (`/var/named/named.stats`), soit dans le fichier spécifié dans l'option `statistics-file` de la déclaration `server`.

10.5.2.5 Exemples de déclarations zone

La plupart des changements apportés au fichier `/etc/named.conf` d'un serveur de noms maître ou esclave implique l'ajout, la modification ou la suppression de déclarations de `zone`. Alors que ces déclarations de `zone` peuvent contenir de nombreuses options, la plupart des serveurs de noms nécessitent seulement quelques options pour fonctionner de manière efficace. Les déclarations de `zone` figurant ci-dessous représentent des exemples très élémentaires pour illustrer une relation de serveurs de noms maître/esclave.

Ci-dessous se trouve un exemple de déclaration `zone` pour le serveur de noms primaire hébergeant `example.com` (`192.168.0.1`) :

```
zone "example.com" IN {
  type master;
  file "example.com.zone";
  allow-update { none; };
};
```

Dans cette déclaration, la zone est identifiée en tant que `example.com`, le type est défini comme `master` et le service `named` a comme instruction de lire le fichier `/var/named/example.com.zone`. Elle indique à `named` de refuser la mise à jour à tout autre hôte.

La déclaration `zone` d'un serveur esclave pour `example.com` est légèrement différente de l'exemple précédent. Pour un serveur esclave, le type est `slave` et une directive indiquant à `named` l'adresse IP du serveur maître remplace la ligne `allow-update`.

Ci-dessous se trouve un exemple de déclaration `zone` de serveur de noms pour la zone `example.com` :

```
zone "example.com" {
  type slave;
  file "example.com.zone";
  masters { 192.168.0.1; };
};
```

Cette déclaration `zone` configure `named` sur le serveur esclave de manière à ce qu'il cherche le serveur maître à l'adresse IP `192.168.0.1` pour y trouver les informations concernant la zone appelée `example.com`. Les informations que le serveur esclave reçoit du serveur maître sont enregistrées dans le fichier `/var/named/example.com.zone`.

10.5.3 Autres types de déclarations

Ci-dessous se trouve une liste de types de déclarations moins fréquemment utilisés mais nénamoins disponibles au sein de `named.conf` :

- `controls` — Configure diverses contraintes de sécurité nécessaires â l'utilisation de la commande `rndc` pour administrer le service `named`.

- `key "<key-name>"` — Définit une clé spécifique par nom. Les clés servent à valider diverses actions, comme les mises à jour sécurisées ou l'utilisation de la commande `rndc`. Deux options sont utilisées avec `key` :
  - `algorithm <algorithm-name>` — Représente le type d'algorithme utilisé, tel que `dsa` ou `hmac-md5`.
  - `secret "<key-value>"` — Représente la clé cryptée.
- `logging` — Permet d'utiliser de multiples types de journaux (ou logs) appelés canaux (ou *channels*). En utilisant l'option `channel` dans la déclaration `logging`, il est possible de construire un type de journal personnalisé, avec des caractèristique propres à savoir nom de fichier (`file`), sa limite de taille (`size`), version (`version`) et son propre niveau d'importance (`severity`). Une fois qu'un canal personnalisé a été défini, une option `category` est utilisée pour catégoriser le canal et commencer la journalisation quand `named` est redémarrée.

  Par défaut, `named` journalise des messages standards grâce au démon `syslog` qui les place dans `/var/log/messages`. Ce processus de journalisation a lieu car plusieurs canaux standards sont intégrés dans BIND, avec plusieurs niveaux d'importance (severity levels), comme celui qui traite les messages de journalisation informationnels (`default_syslog`) et celui qui traite spécifiquement les messages de débogage (`default_debug`). Une catégorie par défaut, appelée `default`, utilise les canaux compris dans BIND pour accomplir la journalisation normale, sans configuration spéciale.

  La personnalisation de la journalisation peut être un processus très détaillé qui dépasse le cadre du présent chapitre. Pour obtenir davantage d'informations sur la création de journaux personnalisés avec BIND.

- `server` — Définit des options particulières qui affectent la façon selon laquelle `named` doit répondreaux serveurs de noms distants, particulièrement pour ce qui est des notifications et des transferts de zone.

  L'option `transfer-format` détermine si un enregistrement de ressources est envoyé avec chaque message (`one-answer`) ou si des enregistrements de ressources multiples sont envoyés avec chaque message (`many-answers`). Alors que l'option `many-answers` est plus efficace, seuls les serveurs de noms BIND les plus récents peuvent la comprendre.

- `trusted-keys` — Contient des clés publiques variées qui sont utilisées pour des DNS sécurisés (DNSSEC).
- `view "<view-name>"` — Crée des vues spéciales en fonction du réseau sur lequel l'hôte interroge le serveur de noms. Ce type de déclaration permet à certains hôtes de recevoir une réponse quant à une zone particulière alors que d'autres hôtes reçoivent des informations totalement différentes. Certains hôtes dignes de confiance peuvent également se voir accorder l'accès à certaines zones alors que d'autres hôtes qui ne sont pas dignes de confiance doivent limiter leurs requêtes à d'autres zones.

  Il est possible d'obtenir de multiples vues mais leurs leurs noms doivent être uniques. L'option `match-clients` spécifie les adresses IP qui s'appliquent à une vue particulière. Toute déclaration `options` peut aussi être utilisée dans une vue, avec

priorité sur les options globales déjà configurées pour `named`. La plupart des déclarations `view` contiennent de multiples déclarations `zone` qui s'appliquent à la liste `match-clients`. L'ordre dans lequel les déclarations `view` sont énumérées est important, puisque c'est la première déclaration `view` qui correspond à l'adresse IP d'un client, qui est utilisée.

10.5.4 Balises de commentaire

La liste suivante regroupe les balises (ou tags) de commentaire valides qui sont utilisés dans `named.conf` :

- `//` — Lorsque ce symbole est placé en début de ligne, cette dernière n'est pas prise en compte par `named`.
- `#` — Lorsque ce symbole est placé en début de ligne, cette dernière n'est pas prise en compte par `named`.
- `/*` et `*/` — Lorsque du texte est placé entre ces symboles, le bloc de texte en question n'est pas pris en compte par `named`.

10.6 Fichiers de zone

Les *Fichiers de zone* contiennent des informations sur un espace de nom particulier et sont stockés dans le répertoire de travail `named` qui est par défaut `/var/named/`. Chaque fichier de zone est nommé selon les données d'options de `file` dans la déclaration `zone`, et ce, généralement d'une manière qui se réfère au domaine en question et identifie le fichier comme contenant des données de zone, telles que `example.com.zone`.

Chaque fichier de zone peut contenir des *directives* et des *enregistrements de ressources*. Les directives donnent au serveur de noms l'instruction d'effectuer une certaine tâche ou d'appliquer des paramètres spéciaux à la zone. Les enregistrements de ressources définissent les paramètres de la zone, assignant des identités aux hôtes individuels. Les directives sont facultatives, mais les enregistrements de ressources sont requis pour fournir un service de nom à une zone.

Toutes les directives et enregistrements de ressources doivent être spécifiées sur des lignes individuelles.

Des commentaires peuvent être placés dans les fichiers de zone après les caractères points-virgules (`;`).

10.6.1 Directives des fichiers de zone

Les directives sont identifiées par le symbole dollar (`$`) suivit du nom de la directive. Elles apparaissent généralement en haut du fichier de zone.

Les directives les plus couramment utilisées sont les suivantes :

- `$INCLUDE` — Configure `named` de façon à ce qu'il inclue un autre fichier de zone dans ce fichier de zone à l'endroit où la directive apparaît. Ce faisant, il est possible de

stocker des configurations de zone supplémentaires à l'écart du fichier de zone principal.

- `$ORIGIN` — Attache le nom de domaine à des enregistrements non-qualifiés, comme ceux qui spécifient seulement l'hôte et rien de plus.

  Par exemple, un fichier de zone peut contenir la ligne suivante :

  ```
  $ORIGIN example.com.
  ```

  Tous les noms utilisés dans les enregistrement de ressources qui ne se terminent pas par un point (`.`) se verront ajouter `example.com` .

  Remarque

  L'utilisation de la directive `$ORIGIN` n'est pas nécessaire si l'on nomme la zone dans `/etc/named.conf` parce que le nom de la zone est utilisé par défaut, comme la valeur de la directive `$ORIGIN`

- `$TTL` — Règle la valeur par défaut de *Time to Live (TTL)* (ou temps de vie) pour la zone. Cette valeur exprimée en secondes, correspond à la durée pendant laquelle un enregistrement de ressources de zone est valide. Chaque enregistrement de ressources peut contenir sa propre valeur TTL, qui remplace alors cette directive.

  L'aumentation de cette valeur permet aux serveurs de noms distants de mettre en cache ces informations de zone pendant plus longtemps, réduisant ainsi nombre de requêtes effectuées au sujet de cette zone et rallongeant le temps nécessaire pour la prolifération des changements apportés aux enregistrements de ressources.

10.6.2 Enregistrements de ressources des fichiers de zone

Les enregistrements de ressources représentent le premier composant d'un fichier de zone.

Il existe de nombreux types d'enregistrements de ressources des fichiers de zone. Ceux énumérés ci-dessous sont néanmoins les plus fréquemment utilisés :

- `A` — Enregistrement d'adresse qui spécifie une adresse IP à assigner à un nom, comme dans l'exemple ci-dessous :

  ```
  <host>      IN      A      <IP-address>
  ```

- Si la valeur `<host>` est omise, alors un enregistrement `A` renvoie à une adresse IP par défaut pour la partie supérieure de l'espace de nom. Ce système est la cible de toutes les requêtes non-FQDN.
- Examinons les exemples d'enregistrements `A` suivants pour le fichier de zone `example.com` :

  ```
                IN      A       10.0.1.3
  server1       IN      A       10.0.1.5
  ```

- Les requêtes pour `example.com` sont dirigées vers 10.0.1.3, alors que les requêtes pour `server1.example.com` sont orientées vers 10.0.1.5.
- `CNAME` — Enregistrement de nom canonique mappant un nom à un autre. Ce type d'enregistrement est plus connu sous le nom d'enregistrement d'alias.

  L'exemple suivant indique à `named` que toute requête envoyée à *<alias-name>* devrait être dirigée vers l'hôte, *<real-name>*. Les enregistrements `CNAME` sont généralement utilisés pour pointer vers les services qui utilisent un procédé commun de nommage, comme par exemple, `www` pour les serveurs Web.

```
<alias-name>      IN      CNAME        <real-name>
```

  Dans l'exemple suivant, un enregistrement `A` lie un nom d'hôte à une adresse IP alors qu'un enregistrement `CNAME` pointe le nom d'hôte `www` le fréquemment utilisé vers l'adresse.

```
server1       IN      A        10.0.1.5
www           IN      CNAME    server1
```

- `MX` — Enregistrement Mail eXchange, qui indique où devrait aller le courrier envoyé à un nom d'espace particulier contrôlé par cette zone.

```
      IN      MX      <preference-value>  <email-server-name>
```

- Dans cet exemple, *<preference-value>* permet de classer numériquement les serveurs de messagerie pour un espace de nom, en donnant une préférence à certains systèmes de messagerie par rapport à d'autres. L'enregistrement de ressources `MX` doté de la valeur *<preference-value>* la plus basse est préféré aux autres. Toutefois, de multiples serveurs de messagerie peuvent avoir la même valeur pour distribuer uniformément le trafic d'emails entre eux.
- L'option *<email-server-name>* peut être un nom d'hôte ou un FQDN.

```
      IN      MX      10      mail.example.com.
      IN      MX      20      mail2.example.com.
```

- Dans cet exemple, le premier serveur de messagerie `mail.example.com` est préféré au serveur de messagerie `mail2.example.com` lors de la réception des courriers électroniques destinés au domaine `example.com`.
- `NS` — Enregistrement de serveur de noms (NameServer) annonçant les serveurs de noms faisant autorité pour une zone particulière.

  Ci-dessous figure un exemple d'enregistrement `NS` :

```
      IN      NS      <nameserver-name>
```

  L'élément *<nameserver-name>* devrait correspondre à un FQDN.

Ensuite, deux serveurs de noms sont répertoriés comme faisant autorité pour le domaine. Le fait que ces serveurs de noms soient esclaves ou que l'un d'eux soit maître n'a pas d'importance ; ils sont tous les deux considérés comme faisant autorisé.

```
        IN      NS      dns1.example.com.
        IN      NS      dns2.example.com.
```

- `PTR` — Enregistrement PoinTeR, conçu pour pointer vers une autre partie de l'espace de nom.

  Les enregistrements `PTR` servent essentiellement à la résolution inverse des noms, puisqu'ils renvoient les adresses IP vers un nom particulier.

- `SOA` — Enregistrement de ressources Start Of Authority, proclame des informations importantes faisant autorité sur un espace de nom pour le serveur de noms.

  Situé après les directives, un enregistrement de ressources `SOA` est le premier enregistrement de ressources dans un fichier de zone.

  L'exemple qui suit montre la structure de base d'un enregistrement de ressources `SOA` :

```
@       IN      SOA     <primary-name-server>     <hostmaster-email> (
                        <serial-number>
                        <time-to-refresh>
                        <time-to-retry>
                        <time-to-expire>
                        <minimum-TTL> )
```

  Le symbole `@` place la directive `$ORIGIN` (ou le nom de zone, si la directive `$ORIGIN` n'est pas déterminée) en tant qu'espace de nom défini par le présent enregistrement de ressources `SOA`. Le nom d'hôte du serveur de noms primaire faisant autorité pour ce domaine est utilisé pour le *<primary-name-server>* et l'adresse électronique de la personne à contacter à propos de cet espace de nom est remplacée par *<hostmaster-email>*.

  La directive *<serial-number>* est incrémentée lors de chaque modification du fichier de zone afin que `named` sache qu'il doit recharger cette zone. La valeur *<time-to-refresh>* indique à tout serveur esclave combien de temps il doit attendre avant de demander au serveur de noms maître si des changements ont été effectués dans la zone. La valeur *<serial-number>* est utilisée par le serveur esclave pour déterminer s'il est en train d'utiliser des données de zone périmées et doit par conséquent les rafraîchir.

  La valeur *<time-to-retry>* précise au serveur de noms esclave l'intervalle pendant lequel il doit attendre avant d'émettre une autre requête de rafraîchissement, au cas où le serveur de noms maître ne répondrait pas. Si le serveur maître n'a pas répondu à une requête de rafraîchissement avant que la durée indiquée dans *<time-to-expire>* ne se soit écoulée, le serveur esclave cesse de répondre en tant qu'autorité pour les requêtes au sujet de cet espace de nom.

La valeur *<minimum-TTL>* demande que d'autres serveurs de noms mettent en cache les informations pour cette zone pendant au moins cette durée définie.

Lors de la configuration de BIND, toutes les durées sont exprimées en secondes. Toutefois, il est égalemnt possible d'utiliser des abréviations pour des unités de temps autres que des secondes, comme les minutes (`M`), les heures (`H`), les jours (`D`) et les semaines (`W`). Le Tableau 10-1 montre une durée exprimée en secondes et la période équivalente dans un autre format.

Secondes	Autres unités de temps
60	1M
1800	30M
3600	1H
10800	3H
21600	6H
43200	12H
86400	1D
259200	3D
604800	1W
31536000	365D

Tableau 10-1. Les secondes comparées à d'autres unités de temps

L'exemple suivant montre ce à quoi l'enregistrement d'une ressource `SOA` peut ressembler lorsqu'il est configuré avec des valeurs réelles.

```
@     IN     SOA    dns1.example.com.     hostmaster.example.com. (
                    2001062501 ; serial
                    21600      ; refresh after 6 hours
                    3600       ; retry after 1 hour
                    604800     ; expire after 1 week
                    86400 )    ; minimum TTL of 1 day
```

10.6.3 Exemples de fichiers de zone

Si on les observe individuellement, les directives et enregistrements de ressources peuvent être difficiles à comprendre. Cependant, tout devient beaucoup plus simple lorsqu'on peut les observer ensemble dans un seul fichier commun.

L'exemple suivant illustre un fichier de zone très élémentaire.

```
$ORIGIN example.com.
$TTL 86400
@     IN     SOA    dns1.example.com.     hostmaster.example.com. (
                    2001062501 ; serial
                    21600      ; refresh after 6 hours
                    3600       ; retry after 1 hour
                    604800     ; expire after 1 week
                    86400 )    ; minimum TTL of 1 day

      IN     NS     dns1.example.com.
      IN     NS     dns2.example.com.

      IN     MX     10     mail.example.com.
      IN     MX     20     mail2.example.com.

             IN     A       10.0.1.5

server1      IN     A       10.0.1.5
server2      IN     A       10.0.1.7
dns1         IN     A       10.0.1.2
dns2         IN     A       10.0.1.3

ftp          IN     CNAME   server1
mail         IN     CNAME   server1
mail2        IN     CNAME   server2
www          IN     CNAME   server2
```

Dans cet exemple sont utilisées des directives et des valeurs `SOA` standard. Les serveurs de noms faisant autorité seront `dns1.example.com` et `dns2.example.com`, qui ont des enregistrements A les liant respectivement à `10.0.1.2` et à `10.0.1.3`.

Les serveurs de messagerie configurés par les enregistrements `MX` pointent vers les serveurs `server1` et `server2` au moyen des enregistrements `CNAME`. Puisque les noms des serveurs `server1` et `server2` ne finissent pas par un point (`.`), le domaine `$ORIGIN` est attaché, rallongeant le nom en `server1.example.com` et `server2.example.com`. Grâce aux enregistrements de ressources `A` associés, leurs adresses IP peuvent être déterminées.

Les services FTP et Web services, disponibles aux noms `ftp.example.com` et `www.example.com` standard, sont pointés vers les serveurs appropriés en utilisant les enregistrements `CNAME`.

10.6.4 Fichiers de résolution de noms inverse

Un fichier de zone de résolution de nom inverse est utilisé pour traduire une adresse IP dans un espace de nom particulier en un FQDN. Il ressemble beaucoup à un fichier de zone

standard, sauf que les enregistrements de ressources `PTR` servent à lier les adresses IP au nom d'un domaine pleinement qualifié.

Un enregistrement `PTR` ressemble à l'extrait ci-dessous :

```
<last-IP-digit>      IN     PTR    <FQDN-of-system>
```

L'élément *<last-IP-digit>* fait référence au dernier chiffre d'une adresse IP qui pointe vers le FQDN d'un système particulier.

Dans l'exemple suivant, les adresses IP allant de `10.0.1.20` à `10.0.1.25` pointent vers les FQDN correspondants.

```
$ORIGIN 1.0.10.in-addr.arpa.
$TTL 86400
@     IN     SOA    dns1.example.com.     hostmaster.example.com. (
                    2001062501 ; serial
                    21600      ; refresh after 6 hours
                    3600       ; retry after 1 hour
                    604800     ; expire after 1 week
                    86400 )    ; minimum TTL of 1 day

      IN     NS     dns1.example.com.
      IN     NS     dns2.example.com.

20    IN     PTR    alice.example.com.
21    IN     PTR    betty.example.com.
22    IN     PTR    charlie.example.com.
23    IN     PTR    doug.example.com.
24    IN     PTR    ernest.example.com.
25    IN     PTR    fanny.example.com.
```

Ce fichier de zone serait mis en service avec une déclaration `zone` dans le fichier `named.conf` similaire à l'extrait suivant :

```
zone "1.0.10.in-addr.arpa" IN {
  type master;
  file "example.com.rr.zone";
  allow-update { none; };
};
```

Il existe peu de différences entre cet exemple et une déclaration `zone` standard, sauf pour ce qui est de la manière de nommer l'hôte. Notez qu'une zone de résolution de noms inverse nécessite que les trois premiers blocs de l'adresse IP soient inversés, puis suivis de l'entité `.in-addr.arpa`. Ce faisant, il est possible d'associer correctement à cette zone le bloc unique de nombres IP utilisé dans le fichier de zone de résolution de nom inverse.

10.7 Utilisation de rndc

BIND contient un utilitaire appelé `rndc` qui permet d'utiliser des lignes de commande pour administrer le démon `named` à partir de l'hôte local ou d'un hôte distant.

Afin d'empêcher l'accès non-autorisé au démon `named`, BIND utilise une méthode d'authentification à clé secrète partagée pour accorder des privilèges aux hôtes. Ainsi, une clé identique doit être présente aussi bien dans `/etc/named.conf` que dans le fichier de configuration de `rndc`, à savoir `/etc/rndc.conf`

10.7.1 Configuration de `/etc/named.conf`

Pour que `rndc` puisse se connecter à un service `named`, une déclaration `controls` doit être présente dans le fichier `/etc/named.conf` du serveur BIND.

La déclaration `controls`, décrite dans l'exemple qui suit, permet à `rndc` de se connecter à partir d'un hôte local.

```
controls {
  inet 127.0.0.1 allow { localhost; } keys { <key-name>; };
};
```

Cette déclaration indique à `named` qu'il doit écouterle port TCP 953 par défaut de l'adresse inversée et doit 'autoriser les commandes `rndc` provenant de l'hôte local, si la clé adéquate est donnée. Le *<key-name>* fait référence à la déclaration `key`, qui se trouve dans le fichier `/etc/named.conf`. L'exemple suivant illustre une déclaration `key`.

```
key "<key-name>" {
  algorithm hmac-md5;
  secret "<key-value>";
};
```

Dans ce cas, la déclaration *<key-value>* utilise l'algorithme HMAC-MD5. Afin de créer des clés à l'aide de l'algorithme HMAC-MD5, utilisez la commande suivante :

```
dnssec-keygen -a hmac-md5 -b <bit-length> -n HOST <key-file-name>
```

Une clé d'au moins 256 bits de long est un bon choix. La clé qui doit être placée dans la zone *<key-value>* se trouve dans le fichier *<key-file-name>* généré par cette commande.

Avertissement

Étant donné que `/etc/named.conf` est lisible par tout un chacun, il est judicieux de placer la déclaration `key` dans un fichier séparé que seul le super-utilisateur (ou root) peut lire et d'utiliser ensuite une déclaration `include` pour le référencer. Par exemple :

```
include "/etc/rndc.key";
```

10.7.2 Configuration de `/etc/rndc.conf`

La déclaration `key` représente la déclaration la plus importante du fichier `/etc/rndc.conf`.

```
key "<key-name>" {
  algorithm hmac-md5;
  secret "<key-value>";
```

```
};
```

Les éléments <*key-name*> et <*key-value*> doivent être absolument identiques aux paramètres les concernant qui figurent dans `/etc/named.conf`.

Pour établir la correspondance entre les clés spécifiées dans le fichier `/etc/named.conf` du serveur cible, ajoutez les lignes reproduites ci-dessous au fichier `/etc/rndc.conf`.

```
options {
  default-server  localhost;
  default-key     "<key-name>";
};
```

Cette directive détermine une clé globale par défaut. Toutefois, le fichier de configuration `rndc` peut également spécifier différentes clés pour différents serveurs, comme le montre l'exemple suivant :

```
server localhost {
  key  "<key-name>";
};
```

Attention

Assurez-vous que seul le super-utilisateur (ou root) puisse effectuer des opérations de lexture et d'écriture dans le fichier /etc/rndc.conf.

Pour obtenir davantage d'informations sur le fichier `/etc/rndc.conf`, consultez la page de manuel de `rndc.conf`.

10.7.3 Options de ligne de commande

Une commande `rndc` se présente sous le format suivant :

```
rndc <options> <command> <command-options>
```

Lors de l'exécution de `rndc` sur un hôte local configuré de façon appropriée, les commandes suivantes sont disponibles :

- `halt` — Arrête immédiatement le service `named`.
- `querylog` — Journalise toutes les requêtes effectuées auprès de ce serveur de noms.
- `refresh` — Rafraîchit la base de données du serveur de noms.
- `reload` — Recharge les fichiers de zone mais conserve toutes les réponses précédemment mises en cache. Cette commande permet également d'apporter des changements aux fichiers de zone sans perdre toutes les résolutions de nom stockées.

  Si les changements n'affectent qu'une zone particulière, rechargez seulement cette zone en ajoutant le nom de la zone après la commande `reload`.

- `stats` — Vide les statistiques courantes de `named` vers le fichier `/var/named/named.stats`.

- `stop` — Arrête correctement le serveur, en enregistrant préalablement toute mise à jour dynamique et toute donnée *Incremental Zone Transfers* (*IXFR*).

Dans certaines situations, il sera peut-être nécessaire d'annuler les paramètres par défaut contenus dans le fichier `/etc/rndc.conf`. Les options suivantes sont disponibles :

- `-c <configuration-file>` — Spécifie l'autre emplacement d'un fichier de configuration.
- `-p <port-number>` — Spécifie le numéro de port à utiliser pour la connexion de `rndc`, autre que le port par défaut 953.
- `-s <server>` — Spécifie un serveur autre que le `default-server` (serveur pardéfaut) précisé dans `/etc/rndc.conf`.
- `-y <key-name>` — Spécifie une clé autre que l'option `default-key` (clé par défaut) présente dans le fichier `/etc/rndc.conf`.

Des informations supplémentaires sur ces options sont disponibles dans la page de manuel de `rndc`.

10.8 Fonctionnalités avancées de BIND

La plupart des implémentations de BIND utilisent `named` pour fournir un service de résolution de noms ou pour faire autorité pour un domaine ou sous-domaine particuliers. Toutefois, la version 9 de BIND possède aussi un certain nombre de fonctionnalités avancées qui permettent d'offrir un service DNS plus efficace et plus sécurisé.

Attention

Certaines de ces fonctionnalités avancées, comme DNSSEC, TSIG et IXFR, ne doivent être utilisées que dans les environnements réseau ayant des serveurs de noms qui prennent en charge ces fonctionnalités. Si votre environnement réseau inclut des serveurs de noms autres que BIND ou des versions de BIND plus anciennes, vérifiez que chaque fonctionnalité avancée est bien prise en charge avant d'essayer de l'utiliser.

10.8.1 Améliorations du protocole DNS

BIND prend en charge les transferts de zone incrémentaux (ou IXFR de l'anglais Incremental Zone Transfers) dans lesquels le serveur de noms esclave ne télécharge que les portions mises à jour d'une zone modifiée sur un serveur de noms maître. Le processus de transfert standard nécessite que la zone entière soit transférée vers chaque serveur de noms esclave même pour des changements mineurs. Pour des domaines très populaires avec des fichiers de zones très longs et pour de nombreux serveurs de noms esclaves, IXFR rend la notification et les processus de mise à jour bien moins exigeants en ressources.

Notez que IXFR n'est disponible que si vous utilisez une *mise à jour dynamique* pour apporter des modifications aux informations de zone maître. Si vous éditez manuellement des fichiers de zone pour effectuer des changements, le processus de transfert AXFR (Automatic Zone Transfer) est utilisé.

10.8.2 Vues multiples

En fonction de l'utilisation de la déclaration `view` dans `named.conf`, BIND peut fournir différentes informations en fonction du réseau puis lequel la requête provient.

Cette option est utilisée essentiellement pour refuser des entrées DNS confidentielles depuis des clients externes au réseau local, tout en permettant des requêtes provenant de clients internes au réseau local.

La déclaration `view` utilise l'option `match-clients` pour établir la correspondance entre des adresses IP ou des réseaux entiers et pour leur attribuer des options et des données de zones spéciales.

10.8.3 Sécurité

BIND supporte plusieurs méthodes différentes pour protéger la mise à jour et le transfert de zones, aussi bien sur les serveurs de noms maîtres qu'esclaves :

- *DNSSEC* — Abréviation de *DNS SECurity*, cette fonctionnalité permet de signer cryptographiquement des zones avec une *clé de zone*.

  De cette façon, on peut vérifier que les informations relatives à une zone spécifique proviennent d'un serveur de noms qui les a signées avec une clé privée particulière, du moment que le receveur possède la clé publique de ce serveur de noms.

  La version 9 de BIND prend aussi en charge la méthode d'authentification de messages SIG(0) utilisant un système de clé publique/privée.

- *TSIG* — Abréviation de *Transaction SIGnatures* ; cette fonctionnalité permet d'effectuer un transfert de maître à esclave, mais seulement après vérification qu'une clé secrète partagée existe sur le serveur maître et sur le serveur esclave.

  Cette fonctionnalité renforce la méthode d'autorisation de transfert basée sur l'adresse IP standard. Un agresseur devra non seulement accéder à l'adresse IP pour transférer la zone, mais il devra aussi connaître la clé secrète.

  La version 9 de BIND prend aussi en charge *TKEY*, qui est une autre méthode de clé secrète partagée pour autoriser les transferts de zone.

10.8.4 IP version 6

La version 9 de BIND prend en charge un service de noms dans des environnements IP version 6 (IPv6) grâce aux enregistrements de zone `A6`.

Si votre environnement réseau inclut aussi bien des hôtes IPv4 que des hôtes IPv6, utilisez le démon de résolution léger `lwresd` sur tous les clients du réseau. Ce démon est un serveur de noms très efficace, de type caching-only, qui prend en charge les nouveaux enregistrements d'informations `A6` et `DNAME` utilisés sous IPv6. Consultez la page de manuel de `lwresd` pour obtenir davantage d'informations.

10.9 Erreurs courantes à éviter

De manière générale, les débutants font fréquemment des erreurs en éditant des fichiers de configuration BIND. Évitez les problèmes suivants :

- *Assurez-vous de bien incrémenter le numéro de série lors de toute modification d'un fichier de zone.*

  Si le numéro de série n'est pas incrémenté, le serveur de noms maître possède certes les nouvelles informations correctes, mais les serveurs de noms esclaves ne sont jamais notifiés du changement ou ne tentent pas de rafraîchir leurs données concernant cette zone.

- *Assurez-vous de bien utiliser ellipses et points-virgules correctement dans le fichier `/etc/named.conf`.*

  Un point-virgule omis ou une ellipse non-fermée empêcheront `named` de démarrer.

- *Rappelez-vous bien de placer des points (`.`) dans les fichiers de zone après tous les FQDN mais de les omettre pour les noms d'hôtes.*

  Un point à la fin d'un nom de domaine indique un nom de domaine pleinement qualifié (en d'autres termes, complet). Si le point est omis, `named` ajoutera le nom de la zone ou la valeur `$ORIGIN` après le nom pour le compléter.

- *Si votre pare-feu crée des problèmes en bloquant les connexions du programme `named` vers d'autres serveurs de noms, éditez son fichier de configuration.*

  La version 9 de BIND utilise par défaut des ports aléatoires supérieurs à 1024 pour envoyer des requêtes à d'autres serveurs de noms. Toutefois certains pare-feu s'attendent à ce que tous les serveurs de noms communiquent uniquement en utilisant le port 53. Pour forcer `named` à utiliser le port 53, ajoutez la ligne reproduite ci-dessous dans la déclaration `options` de `/etc/named.conf` :

```
query-source address * port 53;
```

10.10 Ressources supplémentaires

Les sources d'information mentionnées ci-dessous fournissent de la documentation supplémentaire sur l'utilisation de BIND.

10.10.1 Documentation installée

- BIND propose une gamme complète de documentation installée couvrant de nombreux sujets, chacun d'eux étant placé dans son propre répertoire thématique :
  - `/usr/share/doc/bind-<version-number>/` — Ce répertoire dresse une liste des fonctionnalités les plus récentes. Remplacez `<version-number>` par la version de `bind` qui est installée sur le système.

- `/usr/share/doc/bind-<version-number>/arm/` — Ce répertoire contient les versions HTML et SGML du document intitulé *BIND 9 Administrator Reference Manual*, qui décrit de manière détaillée les ressources nécessaires pour BIND, la façon de configurer différents types de serveurs de noms, d'effectuer la répartition de charges ainsi que d'autres sujets avancés. Pour la plupart des nouveaux utilisateurs de BIND, ces ressources constituent le meilleur point de départ. Remplacez `<version-number>` par la version de `bind` qui est installée sur le système.
- `/usr/share/doc/bind-<version-number>/draft/` — Ce répertoire contient des documents techniques variés quui abordent les problèmes liés au service DNS et propose certaines solutions pour les résoudre. Remplacez `<version-number>` par la version de `bind` qui est installée sur le système.
- `/usr/share/doc/bind-<version-number>/misc/` — Ce répertoire contient des documents conçus pour traiter de problèmes spécifiques avancés. Les utilisateurs de la version 8 de BIND devraient consulter le document `migration` pour s'informer des modifications importantes à effectuer pour passer à la version 9 de BIND. Le fichier `options` dresse une liste de toutes les options implémentées dans BIND 9 qui sont utilisées dans `/etc/named.conf`. Remplacez `<version-number>` par la version de `bind` qui est installée sur le système.
- `/usr/share/doc/bind-<version-number>/rfc/` — Ce répertoire fournit tout document RFC en relation avec BIND. Remplacez `<version-number>` par la version de `bind` qui est installée sur le système.

- Les pages de manuel de BIND — Il existe un certain nombre de pages de manuel pour les diverses applications et les fichiers de configuration intervenant avec BIND. La liste suivante mentionne certaines des pages de manuel les plus importantes.

  Applications administratives

  - `man rndc` — Explique les différentes options disponibles lors de l'utilisation de la commande `rndc` pour contrôler un serveur de noms BIND.

  Applications serveur

  - `man named` — Examine les arguments assortis qui peuvent être utilisés pour contrôler le démon du serveur de noms BIND.
  - `man lwresd` — Décrit le but et les options disponibles pour le démon de résolution très léger.

  Fichiers de configuration

  - `man named.conf` — Une liste exhaustive des options disponibles au sein du fichier de configuration `named`.
  - `man rndc.conf` — Une liste exhaustive des options disponibles au sein du fichier de configuration `rndc`.

10.10.2 Sites Web utiles

- http://www.isc.org/products/BIND — La page d'accueil du projet BIND où vous pourrez trouver des informations sur les versions actuelles ainsi qu'une version PDF du document intitulé *BIND 9 Administrator Reference Manual*.
- http://www.redhat.com/mirrors/LDP/HOWTO/DNS-HOWTO.html — Un document couvrant l'utilisation de BIND en tant que serveur de noms de résolution mettant en cache et la configuration de divers fichiers de zone nécessaires pour qu'il soit utilisé comme serveur de noms primaire pour un domaine.

10.10.3 Livres sur le sujet

- *Guide d'administration système de Red Hat Enterprise Linux* — Le chapitre *Configuration de BIND* explique comment configurer un serveur DNS à l'aide de l'**Outil de configuration du service de noms de domaines**.
- *DNS and BIND* de Paul Albitz et Cricket Liu ; publié par O'Reilly & Associates — Un livre de référence populaire qui explique les options de configuration de BIND des plus simples aux plus ésotériques et fournit aussi des stratégies pour sécuriser votre serveur DNS.
- *The Concise Guide to DNS and BIND* de Nicolai Langfeldt ; publié par Que — Un livre qui examine la connexion entre les services de réseaux multiples et BIND, en mettant l'accent sur les sujets techniques et orientés vers des applications pratiques.

Section 11 Installation d'un serveur DDNS avec bind et DHCP

Le *DNS* dynamique

Table of Contents

Résumé

Eléments sur le service DDNS

Les aspects sur la sécurité

11.1 Résumé

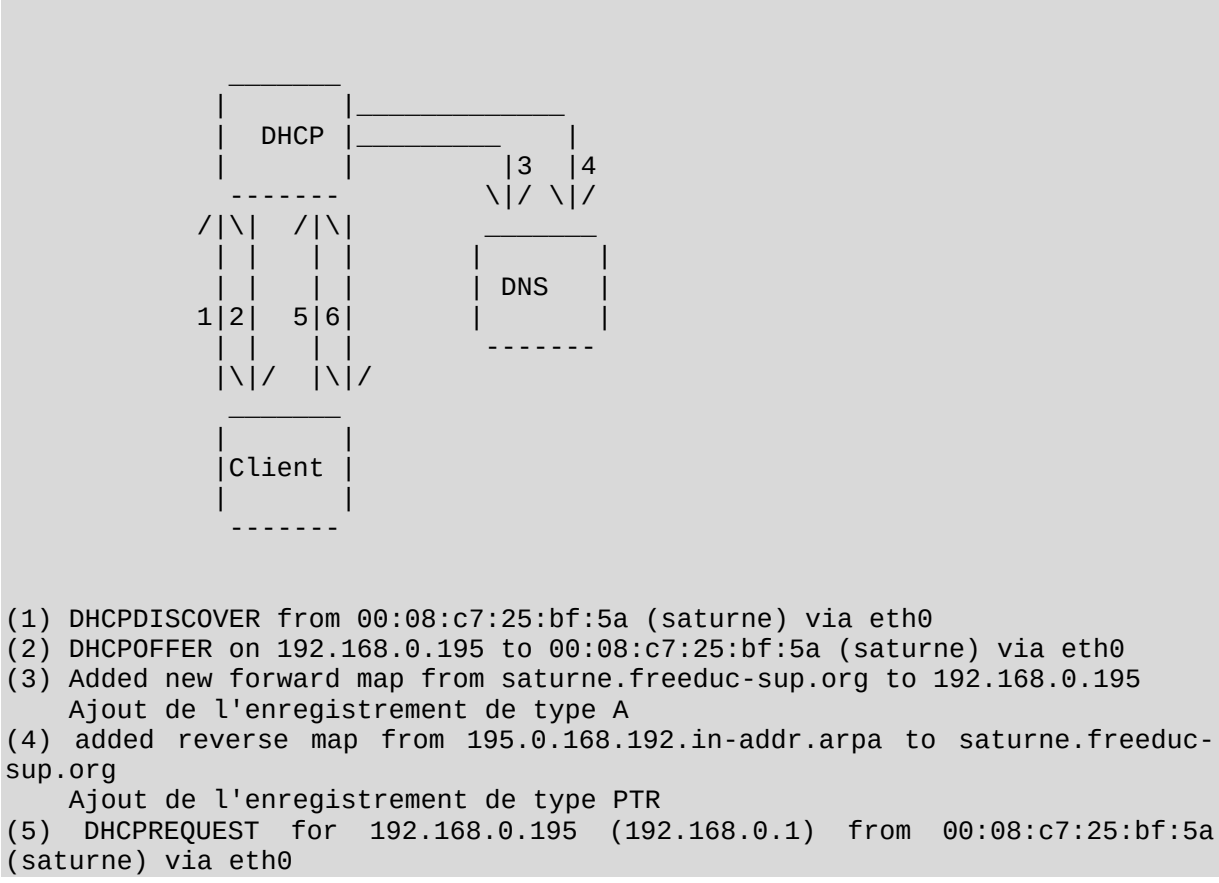
DHCP offre la possibilité de mettre à jour dynamiquement le système de résolution de nom.

Il s'agit, dans cette application, de faire cohabiter et faire fonctionner ensemble le service de résolution de nom bind et le service dhcp.

L'environnement a été testé sur une distribution debian, avec bind9 et dhcp3.

Vous devez savoir configurer un serveur DHCP, un serveur de nom, avoir compris le fonctionnement de rndc et des clés partagées, de dig.

Pour les amateurs d'ASCII-art, voici un schéma qui décrit les processus mis en oeuvre.




```
(1) DHCPDISCOVER from 00:08:c7:25:bf:5a (saturne) via eth0
(2) DHCPOFFER on 192.168.0.195 to 00:08:c7:25:bf:5a (saturne) via eth0
(3) Added new forward map from saturne.freeduc-sup.org to 192.168.0.195
    Ajout de l'enregistrement de type A
(4) added reverse map from 195.0.168.192.in-addr.arpa to saturne.freeduc-
sup.org
    Ajout de l'enregistrement de type PTR
(5) DHCPREQUEST for 192.168.0.195 (192.168.0.1) from 00:08:c7:25:bf:5a
(saturne) via eth0
```

```
(6) DHCPACK on 192.168.0.195 to 00:08:c7:25:bf:5a (saturne) via eth0
```

Les opérations 1, 2, 5, 6 ont déjà été vues lors de l'étude du service DHCP. On voit en étudiant le " log " ci-dessus, que l'inscription dans le DNS d'un client se fait avant l'acceptation du bail et l'inscription finale de ce client (DHCPACK).

Dans cette application, vous installerez successivement le serveur de nom, le serveur dhcp, puis vous ferez les manipulations qui permettent l'intégration.

11.2 Eléments sur le service DDNS

Tout est décrit dans las pages de man de `dhcpd.conf`.

Deux façons de faire sont décrites (ad-hoc et interim) et une troisième est en cours d'élaboration. La méthode ad-hoc n'est semble t-il plus supportée par les paquets, du moins elle ne l'est pas avec le paquet dhcp3 de debian que j'utilise car considérée comme obsolète.

Le processus utilisé est défini par la variable `ddns-updates-style`. Si la mise à jour n'est pas dynamique, la variable prend la valeur `none`, nous, nous utiliserons `interim`.

La méthode " ad-hoc " ne prend pas en charge le protocole `failover` des DHCP. C'est à dire qu'avec cette méthode vous ne pourrez pas avoir 2 serveurs DHCP assurant un système redondant et mettant à jour un même ensemble d'enregistrements DNS.

Le serveur détermine le nom du client en regardant d'abord dans les options de configuration des noms (ddns-hostname). Il est possible de générer dynamiquement un nom pour le client en concaténant des chaînes " dyn+N°+NomDeDomaine ". S'il ne trouve rien, il regarde si le client lui a fait parvenir un nom d'hôte. Si aucun nom n'est obtenu, la mise à jour du DNS n'a pas lieu.

Pour déterminer le nom FQDN, le serveur concatène le nom de domaine au nom d'hôte du client.

Le nom du domaine lui, est défini uniquement sur le serveur DHCP.

Actuellement le processus ne prend pas en charge les clients ayant plusieurs interfaces réseau mais cela est prévu. Le serveur met à jour le DNS avec un enregistrement de type A et un enregistrement de type PTR pour la zone reverse. Nous verrons qu'un enregistrement de type TXT est également généré.

Quand un nouveau bail est alloué, le serveur crée un enregistrement de type TXT qui est une clé MD5 pour le client DHCP (DHCID).

La méthode `interim` est le standard. Le client peut demander au serveur DHCP de mettre à jour le serveur DNS en lui passant ses propres paramètres (nom FQDN). Dans ce cas le serveur est configuré pour honorer ou pas la demande du client. Ceci se fait avec le paramètre *`ignore client-updates`* ou *`allow client-updates`*.

Par exemple, si un client `jschmoe.radish.org` demande à être inscrit dans le domaine `exemple.org` et que le serveur DHCP est configuré pour, le serveur ajoutera un enregistrement PTR pour l'adresse IP mais pas d'enregistrement A. Si l'option *`ignore client-updates`* est configurée, il y aura un enregistrement de type A pour `jschmoe.exemple.org`.

11.3 Les aspects sur la sécurité

Le serveur DNS doit être configuré pour pouvoir être mis à jour par le serveur DHCP. La méthode la plus sûre utilise les signatures TSIG, basées sur une clé partagée comme pour le programme d'administration des serveurs de nom rndc.

Vous devrez en créer une. Pour cela utiliser les éléments fournis dans la partie traitant de bind. Ces aspects y ont déjà été abordés.

Par exemple dans le fichier `named.conf`, le serveur DHCP disposant de la clé *`DHCP_UPDATER`*, pourra mettre à jour la zone directe et la zone reverse pour lesquelles la déclaration *`allow-update`* existe.

Description du fichier `named.conf` :

```
key DHCP_UPDATER {
        algorithm HMAC-MD5.SIG-ALG.REG.INT;
        secret pRP5FapFoJ95JEL06sv4PQ==;
      };

      zone "example.org" {
           type master;
           file "example.org.db";
           allow-update { key DHCP_UPDATER; };
      };

      zone "17.10.10.in-addr.arpa" {
           type master;
           file "10.10.17.db";
           allow-update { key DHCP_UPDATER; };
      };
```

Dans le fichier de configuration du serveur DHCP vous pourrez mettre :

```
key DHCP_UPDATER {
        algorithm HMAC-MD5.SIG-ALG.REG.INT;
        secret pRP5FapFoJ95JEL06sv4PQ==;
      };

      zone EXAMPLE.ORG. {
        primary 127.0.0.1; # Adresse du serveur de noms primaire
        key DHCP_UPDATER;
      }

      zone 17.127.10.in-addr.arpa. {
        primary 127.0.0.1; # Adresse du serveur de noms primaire
        key DHCP_UPDATER;
      }
```

La clé *DHCP_UPDATER* déclarée pour une zone dans le fichier `dhcpd.conf` est utilisée pour modifier la zone si la clé correspond dans le fichier `named.conf`.

Les déclarations de zone doivent correspondre aux enregistrements SOA des fichiers de ressources des zones.

Normalement il n'est pas obligatoire d'indiquer l'adresse du serveur de nom primaire, mais cela peut ralentir le processus d'inscription des enregistrements, voire même ne pas fonctionner, si le serveur de nom n'a pas répondu assez vite.

Travaux pratiques : DDNS

Table of Contents

Réalisation
Les fichiers de configuration
    Le fichier `named.conf`
    Le fichier de zone directe
    Le fichier de zone `in-addr.arpa`
    Le fichier `rndc.conf`
    Le fichier de clé partagée
    Le fichier `dhcpd.conf`
Procédure de tests des services
Intégration des services
Générer un nom dynamiquement pour les clients DHCP

Réalisation

Vous allez réaliser l'opération avec un client windows 2000 serveur et un client Linux. Le serveur Linux sera également serveur de nom.

Vous pouvez utiliser les exemples de fichiers fournis. Vosu aurez bien sûr à les adapter à votre configuration. Voici comment vont se dérouler les étapes :

1. Installation du serveur de nom et test
2. Installation du serveur DHCP et test
3. Intégration des deux services

Nous verrons à la fin comment générer des noms dynamiquement pour les clients.

Les fichiers de configuration

Dans les fichiers il y a des lignes qui sont en commentaires avec ###, elles seront décommentées pour la phase d'intégration des services

Le fichier `named.conf`

```
// Pour journaliser, les fichiers doivent créés
logging {
            channel update_debug {
                 file "/var/log/log-update-debug.log";
                 severity  debug 3;
                 print-category yes;
                 print-severity yes;
                 print-time    yes;
            };
            channel security_info    {
                 file "/var/log/log-named-auth.info";
                 severity  info;
                 print-category yes;
                 print-severity yes;
```

```
                print-time    yes;
            };

            category update { update_debug; };
            category security { security_info; };
        };


// clé partagée entre bind, rndc et dhcp
include "/etc/bind/mykey";

options {
        directory "/var/cache/bind";
        query-source address * port 53;
        auth-nxdomain yes;    # conform to RFC1035
        forwarders { 127.0.0.1; 192.168.0.1;};
};

// Autorisations rndc sur la machine.
controls {
   inet  127.0.0.1   allow {any;} keys {mykey;};
   inet  192.168.0.0 allow {any;} keys {mykey;};
};


zone "." {
        type hint;
        file "/etc/bind/db.root";
};


zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};

zone "freeduc-sup.org" {
        type master;
        file "/etc/bind/freeduc-sup.org.hosts";
// Sert à la mise à jour par DHCP
// Sera décommenté lors de l'intégration des services
###        allow-update { key mykey; };
        };

zone "0.168.192.in-addr.arpa" {
         type master;
```

```
        file "/etc/bind/freeduc-sup.org.hosts.rev";
// Sert à la mise à jour par DHCP
// Sera décommenté lors de l'intégration des services
###        allow-update { key mykey; };
        };
```

Le fichier de zone directe

```
$ORIGIN .
$TTL 86400     ; 1 day
freeduc-sup.org        IN SOA  master.freeduc-sup.org. root.freeduc-sup.org.
(
                               2004050103 ; serial
                               10800      ; refresh (3 hours)
                               3600       ; retry (1 hour)
                               604800     ; expire (1 week)
                               38400      ; minimum (10 hours 40 minutes)
                               )
                       NS      master.freeduc-sup.org.
                       MX      10 master.freeduc-sup.org.
$ORIGIN freeduc-sup.org.
master                 A       192.168.0.1
www                    CNAME   master
```

Le fichier de zone `in-addr.arpa`

```
$ORIGIN .
$TTL 86400     ; 1 day
0.168.192.in-addr.arpa IN SOA  master.freeduc-sup.org. root.freeduc-sup.org.
(
                               2004050103 ; serial
                               10800      ; refresh (3 hours)
                               3600       ; retry (1 hour)
                               604800     ; expire (1 week)
                               38400      ; minimum (10 hours 40 minutes)
                               )
                       NS      master.freeduc-sup.org.
$ORIGIN 0.168.192.in-addr.arpa.
1                      PTR     master.freeduc-sup.org.
```

Le fichier `rndc.conf`

```
include "/etc/bind/mykey";
options {
        default-server  localhost;
        default-key     "mykey";
};

server localhost {
        key     "mykey";
};
```

Le fichier de clé partagée

Ici il est nommé `mykey`.

```
key "mykey" {
        algorithm       hmac-md5;
        secret
"X/ErbPNOiXuC8MIgTX6iRcaq/1OFCEDIlxrmnfPgdqYIOY3U6lsgDMq15jnxXEXmdGvv1g/ayY
tAA73bUQvWBw==";
};
```

Le fichier `dhcpd.conf`

```
ddns-update-style none;
### ddns-update-style interim;
###          deny client-updates;
###          ddns-updates on;
###          ddns-domainname "freeduc-sup.org";
###          ddns-rev-domainname "in-addr.arpa";
authoritative;

subnet 192.168.0.0 netmask 255.255.255.0 {
option broadcast-address  192.168.0.255;
option routers 192.168.0.2;
option domain-name "freeduc-sup.org";
option domain-name-servers 192.168.0.1;
option broadcast-address  192.168.0.255;
option routers 192.168.0.2;
range 192.168.0.100 192.168.0.195;
default-lease-time 600;
max-lease-time 7200;

# Instructions pour la mise à jour des zones
### include "/etc/bind/mykey";

### zone freeduc-sup.org. {
###         primary 192.168.0.1;
###         key mykey;
###         }

### zone 0.168.192.in-addr.arpa. {
###         primary 192.168.0.1;
###         key mykey;
###         }

}
```

Procédure de tests des services

Vous allez pouvoir tester. A partir de maintenant vous devrez consulter les fichiers de logs si vous rencontrez des problèmes de fonctionnement, les tables de processus... bref tout ce qui pourra vous permettre de déterminer la ou les sources possibles des dysfonctionnements si vous en constatez.

Lancez le service bind et tester son fonctionnement avec rndc.

```
root@master:/home/knoppix# rndc status
number of zones: 8
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
server is up and running
root@master:/home/knoppix#
```

Cela permet de vérifier que la clé est bien reconnue.

Vérifiez le fonctionnement du serveur à l'aide de la commande **dig**.

```
root@master:/home/knoppix/tmp# dig @127.0.0.1 freeduc-sup.org axfr
; <<>> DiG 9.2.2 <<>> @127.0.0.1 freeduc-sup.org axfr
;; global options:  printcmd
freeduc-sup.org.          86400    IN       SOA      master.freeduc-sup.org.
root.freeduc-sup.org. 2004050107 10800 3600 604800 38400
freeduc-sup.org.        86400   IN      NS      master.freeduc-sup.org.
freeduc-sup.org.        86400   IN      MX      10 master.freeduc-sup.org.
argo.freeduc-sup.org.   86400   IN      A       192.168.0.253
master.freeduc-sup.org. 86400   IN      A       192.168.0.1
www.freeduc-sup.org.    86400   IN      CNAME   master.freeduc-sup.org.
freeduc-sup.org.          86400    IN       SOA      master.freeduc-sup.org.
root.freeduc-sup.org. 2004050107 10800 3600 604800 38400
;; Query time: 36 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue May  6 19:15:38 2003
;; XFR size: 8 records
```

Vérifier de la même façon le fonctionnement de la zone reverse.

Vérifiez la strucure du fichier `dhcp`.

```
root@master:/home/knoppix# dhcpd3 -t
Internet Software Consortium DHCP Server V3.0.1rc9
Copyright 1995-2001 Internet Software Consortium.
All rights reserved.
For info, please visit http://www.isc.org/products/DHCP
```

Cela permet de vérifier qu'il n'y a pas d'erreur de syntaxe dans le fichier.

Testez le fonctionnement traditionnel de votre serveur DHCP à partir d'un client Linux et windows. Faites des renouvellements de baux.

Intégration des services

Par défaut les clients Linux ne transmettent pas leur nom d'hôte comme c'est le cas pour les clients windows. Modifiez sur le client Linux le fichier `/etc/dhclient.conf` de la façon suivante, nous verrons plus loin comment générer un nom dynamiquement :

```
[root@bestof mlx]# more /etc/dhclient.conf
           send host-name "bestof";
```

Décommentez dans les fichiers `named.conf` et `dhcpd.conf` les lignes commentées par `###`.

Supprimez dans le **dhcpd.conf** la ligne :

```
ddns-update-style none;
```

Relancez le service DNS et testez sont bon fonctionnement

Vérifiez le fichier `dhcpd.conf` avec la commande **`dhcpd3 -t`**.

Lancez dhcp en mode *`foreground`* **`dhcpd3 -d`**, voici ce que vous devriez obtenir :

```
root@master:/etc/dhcp3# dhcpd3 -d
Internet Software Consortium DHCP Server V3.0.1rc9
Copyright 1995-2001 Internet Software Consortium.
All rights reserved.
For info, please visit http://www.isc.org/products/DHCP
Wrote 1 leases to leases file.
Listening on LPF/eth0/00:d0:59:82:2b:86/192.168.0.0/24
Sending on   LPF/eth0/00:d0:59:82:2b:86/192.168.0.0/24
Sending on   Socket/fallback/fallback-net
```

Demandez un bail à partir du client windows (ici windows 2000 Server), voici ce qui devrait se passer :

```
DHCPDISCOVER from 00:08:c7:25:bf:5a (saturne) via eth0
DHCPOFFER on 192.168.0.195 to 00:08:c7:25:bf:5a (saturne) via eth0
Added new forward map from saturne.freeduc-sup.org to 192.168.0.195
added reverse map from 195.0.168.192.in-addr.arpa to saturne.freeduc-
sup.org
DHCPREQUEST for 192.168.0.195 (192.168.0.1) from 00:08:c7:25:bf:5a
(saturne) via eth0
DHCPACK on 192.168.0.195 to 00:08:c7:25:bf:5a (saturne) via eth0
```

Demandez un bail à partir du client Linux, voici ce qui devrait se passer :

```
DHCPDISCOVER from 00:08:c7:25:ca:7c via eth0
DHCPOFFER on 192.168.0.194 to 00:08:c7:25:ca:7c (bestof) via eth0
Added new forward map from bestof.freeduc-sup.org to 192.168.0.194
added reverse map from 194.0.168.192.in-addr.arpa to bestof.freeduc-sup.org
DHCPREQUEST for 192.168.0.194 (192.168.0.1) from 00:08:c7:25:ca:7c (bestof)
via eth0
DHCPACK on 192.168.0.194 to 00:08:c7:25:ca:7c (bestof) via eth0
```

Voici le contenu du fichier de journalisation de bind :

```
Log de Bind log-update-debug.log
root@master:/var/log# more log-update-debug.log
May 06 07:49:50.457 update: info: client 192.168.0.1#32846: updating zone
'freeduc-sup.org/IN': adding an RR
May 06 07:49:50.458 update: info: client 192.168.0.1#32846: updating zone
'freeduc-sup.org/IN': adding an RR
May 06 07:49:50.512 update: info: client 192.168.0.1#32846: updating zone
'0.168.192.in-addr.arpa/IN': deleting an rrse
t
May 06 07:49:50.512 update: info: client 192.168.0.1#32846: updating zone
'0.168.192.in-addr.arpa/IN': adding an RR
May 06 07:50:47.011 update: info: client 192.168.0.1#32846: updating zone
'freeduc-sup.org/IN': adding an RR
May 06 07:50:47.011 update: info: client 192.168.0.1#32846: updating zone
'freeduc-sup.org/IN': adding an RR
May 06 07:50:47.017 update: info: client 192.168.0.1#32846: updating zone
'0.168.192.in-addr.arpa/IN': deleting an rrse
t
May 06 07:50:47.017 update: info: client 192.168.0.1#32846: updating zone
'0.168.192.in-addr.arpa/IN': adding an RR
root@master:/var/log#
```

Voici le contenu du fichier de déclaration de zone avec les nouveaux enregistrements

```
root@master:/var/log# dig @127.0.0.1 freeduc-sup.org axfr

; <<>> DiG 9.2.2 <<>> @127.0.0.1 freeduc-sup.org axfr
;; global options:  printcmd
freeduc-sup.org.          86400    IN       SOA      master.freeduc-sup.org.
root.freeduc-sup.org. 2004050103 10800 3600 604800 38400
freeduc-sup.org.        86400   IN      NS      master.freeduc-sup.org.
freeduc-sup.org.        86400   IN      MX      10 master.freeduc-sup.org.
argo.freeduc-sup.org.   86400   IN      A       192.168.0.253
bestof.freeduc-sup.org.    300                  IN                      TXT
"00e31b2921cd30bfad552ca434b61bda02"
bestof.freeduc-sup.org. 300     IN      A       192.168.0.194
master.freeduc-sup.org. 86400   IN      A       192.168.0.1
saturne.freeduc-sup.org.    300              IN                      TXT
"310e43cfc20efbe1c96798d48672bc76aa"
saturne.freeduc-sup.org. 300    IN      A       192.168.0.195
www.freeduc-sup.org.    86400   IN      CNAME   master.freeduc-sup.org.
freeduc-sup.org.          86400    IN       SOA      master.freeduc-sup.org.
root.freeduc-sup.org. 2004050103 10800 3600 604800 38400
;; Query time: 381 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue May  6 07:56:43 2003
;; XFR size: 12 records
```

Générer un nom dynamiquement pour les clients DHCP

Cela est possible en modifiant le fichier de configuration de DHCP. Vous pourrez retrouver tous les éléments dans la page de manuel.

Par exemple rajoutez dans le fichier la ligne ci-dessous pour adapter le nom à partir de l'adresse MAC du client :

```
#ddns-hostname = binary-to-ascii (16, 8, "-",substring (hardware, 1, 12));
```

Ou celle-ci pour localiser le client :

```
ddns-hostname   =   concat   ("dhcp-a-limoges","-",binary-to-ascii(10,8,"-
",leased-address));
```

Avec cette dernière, voici les enregistrements ajoutés :

```
Added new forward map from dhcp-a-limoges-192-168-0-194.freeduc-sup.org to
192.168.0.194
added reverse map from 194.0.168.192.in-addr.arpa to dhcp-a-limoges-192-
168-0-194.freeduc-sup.org
DHCPREQUEST for 192.168.0.194 from 00:08:c7:25:ca:7c via eth0
DHCPACK on 192.168.0.194 to 00:08:c7:25:ca:7c (bestof) via eth0
```

Le fichier des incriptions :

```
root@master:/home/knoppix# more /var/lib/dhcp3/dhcpd.leases
lease 192.168.0.194 {
  starts 2 2003/05/06 17:38:38;
  ends 2 2003/05/06 17:48:38;
  binding state active;
  next binding state free;
  hardware ethernet 00:08:c7:25:ca:7c;
```

```
  set ddns-rev-name = "194.0.168.192.in-addr.arpa";
  set ddns-txt = "00e31b2921cd30bfad552ca434b61bda02";
  set ddns-fwd-name = "dhcp-192-168-0-194.freeduc-sup.org";
  client-hostname "bestof";
}
```

Les transferts de zones directes et inverses :

```
root@master:/home/knoppix/tmp# dig @127.0.0.1 freeduc-sup.org axfr

; <<>> DiG 9.2.2 <<>> @127.0.0.1 freeduc-sup.org axfr
;; global options:  printcmd
freeduc-sup.org.          86400    IN       SOA      master.freeduc-sup.org.
root.freeduc-sup.org. 2004050116 10800 3600 604800 38400
freeduc-sup.org.        86400   IN      NS      master.freeduc-sup.org.
freeduc-sup.org.        86400   IN      MX      10 master.freeduc-sup.org.
0-8-c7-25-ca-7c.freeduc-sup.org.            300            IN            TXT
"00e31b2921cd30bfad552ca434b61bda02"
0-8-c7-25-ca-7c.freeduc-sup.org. 300 IN A       192.168.0.194
argo.freeduc-sup.org.   86400   IN      A       192.168.0.253
dhcp-192-168-0-194.freeduc-sup.org.           300           IN           TXT
"00e31b2921cd30bfad552ca434b61bda02"
dhcp-192-168-0-194.freeduc-sup.org. 300 IN A    192.168.0.194
dhcp-a-limoges-192-168-0-194.freeduc-sup.org.        300        IN        TXT
"00e31b2921cd30bfad552ca434b61bda02"
dhcp-a-limoges-192-168-0-194.freeduc-sup.org. 300 IN A 192.168.0.194
master.freeduc-sup.org. 86400   IN      A       192.168.0.1
www.freeduc-sup.org.    86400   IN      CNAME   master.freeduc-sup.org.
freeduc-sup.org.          86400    IN       SOA      master.freeduc-sup.org.
root.freeduc-sup.org. 2004050116 10800 3600 604800 38400
;; Query time: 3 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue May  6 19:39:08 2003
;; XFR size: 14 records
```

La zone reverse :

```
root@master:/home/knoppix/tmp# dig @127.0.0.1 0.168.92.in-addr.arpa axfr

; <<>> DiG 9.2.2 <<>> @127.0.0.1 0.168.92.in-addr.arpa axfr
;; global options:  printcmd
; Transfer failed.
root@master:/home/knoppix/tmp# dig @127.0.0.1 0.168.192.in-addr.arpa axfr

; <<>> DiG 9.2.2 <<>> @127.0.0.1 0.168.192.in-addr.arpa axfr
;; global options:  printcmd
0.168.192.in-addr.arpa. 86400    IN       SOA      master.freeduc-sup.org.
root.freeduc-sup.org. 2004050113 10800 3600 604800 38400
0.168.192.in-addr.arpa. 86400   IN      NS      master.freeduc-sup.org.
1.0.168.192.in-addr.arpa. 86400 IN      PTR     master.freeduc-sup.org.
194.0.168.192.in-addr.arpa. 300 IN      PTR     dhcp-192-168-0-194.freeduc-
sup.org.
3.0.168.192.in-addr.arpa. 86400 IN      PTR     argo.freeduc-sup.org.
0.168.192.in-addr.arpa. 86400    IN       SOA      master.freeduc-sup.org.
root.freeduc-sup.org. 2004050113 10800 3600 604800 38400
;; Query time: 3 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue May  6 19:40:08 2003
;; XFR size: 7 records
```

Section 12 Serveur HTTP Apache

Le Serveur HTTP Apache est un serveur Web Open Source robuste de niveau commercial qui a été développé par l'organisation Apache Software Foundation (http://www.apache.org/). Red Hat Enterprise Linux comprend le Serveur HTTP Apache version 2.0 ainsi que de nombreux modules serveur conçus pour améliorer sa fonctionnalité.

Le fichier de configuration par défaut installé avec le Serveur HTTP Apache fonctionne dans la plupart des situations sans devoir être modifié. Ce chapitre décrit brièvement de nombreuses directives présentes dans son fichier de configuration (à savoir `/etc/httpd/conf/httpd.conf`) pour aider les utilisateurs nécessitant une configuration personnalisée ou devant convertir un fichier de configuration dans l'ancien format 1.3 du Serveur HTTP Apache.

Avertissement

Si vous utilisez l'outil graphique Outil de configuration HTTP (system-config-httpd), n'éditez pas manuellement le fichier de configuration du Serveur HTTP Apache car l'Outil de configuration HTTP crée une nouvelle version de ce fichier chaque fois qu'il est utilisé.

12.1 Serveur HTTP Apache 2.0

Il existe des différences importantes entre la version 2.0 et la version 1.3 du Serveur HTTP Apache (la version 1.3 faisait partie de la version 2.1 de Red Hat Enterprise Linux et les versions précédentes). Cette section passe en revue certaines des nouvelles fonctionnalités du Serveur HTTP Apache 2.0 et souligne des changements importants.

12.1.1 Fonctions du Serveur HTTP Apache 2.0

La version 2.0 du Serveur HTTP Apache inclut les fonctionnalités suivantes :

- *API Apache* — Les modules utilisent un nouvel ensemble plus performant d'interfaces de programmation d'applications (ou API de l'anglais Application Programming Interfaces).

  Important

  Les modules élaborés pour le Serveur HTTP Apache 1.3 ne fonctionneront pas s'ils ne sont pas portés vers la nouvelle API. En cas de doute quant au portage d'un module particulier, consultez le développeur *avant* d'effctuer toute mise à niveau.

- *Filtrage* — Les modules peuvent jouer le rôle de filtres de contenu.
- *Prise en charge IPv6* — Le format d'adressage IP nouvelle génération est pris en charge.
- *Directives simplifiées* — Bon nombre de directives complexes ont été supprimées alors que d'autres ont été simplifiées.

- *Réponses multilingues aux erreurs* — Lors de l'utilisation de documents *Server Side Include* (ou *SSI*), des pages de réponse personnalisées en cas d'erreur peuvent être proposées dans plusieurs langues.

Une liste plus complète des changements est disponible en ligne à l'adresse suivante : http://httpd.apache.org/docs-2.0/.

12.1.2 Changements au niveau des paquetages dans le Serveur HTTP Apache 2.0

Depuis la version 3 de Red Hat Enterprise Linux, les paquetages du Serveur HTTP Apache ont été renommés. De plus, certains paquetages connexes ont également été renommés, retirés ou incorporés dans d'autres paquetages.

Ci-dessous figure une liste des changements apportés aux paquetages :

- Les paquetages `apache`, `apache-devel` et `apache-manual` ont été renommés respectivement `httpd`, `httpd-devel` et `httpd-manual`.
- Le paquetage `mod_dav` a été incorporé au paquetage `httpd`.
- Les paquetages `mod_put` et `mod_roaming` ont été supprimés car leur fonctionnalité fait partie d'un sous-ensemble de celle fournie par `mod_dav` (qui est maintenant incorporé dans le paquetage `httpd`).
- Les paquetages `mod_auth_any` et `mod_bandwidth` ont été supprimés.
- Le numéro de version du paquetage `mod_ssl` est désormais synchronisé avec le paquetage `httpd`. Cela signifie que le paquetage `mod_ssl` du Serveur HTTP Apache 2.0 a un numéro de version *inférieur* à celui du paquetage `mod_ssl` pour le Serveur HTTP Apache 1.3.

12.2 Changements apportés au système de fichiers de la version 2.0 du Serveur HTTP Apache

Lors d'une mise à niveau vers la version 2.0 du Serveur HTTP Apache, les changements suivants sont apportés au système de fichiers :

- *Le répertoire de configuration, `/etc/httpd/conf.d/`, a été ajouté* — Ce nouveau répertoire sert à stocker les fichiers de configuration des modules en paquetages individuels, tels que `mod_ssl`, `mod_perl` et `php`. La directive `Include conf.d/*.conf` demande au serveur de charger les fichiers de configuration à partir de cet emplacement au sein du fichier de configuration du Serveur HTTP Apache, `/etc/httpd/conf/httpd.conf`.

  Important

  Lors de la migration d'une configuration existante, il est impératif d'insérer la ligne spécifiant le nouveau répertoire de configuration.

- *Les programmes `ab` et `logresolve` ont été déplacés* — Ces utilitaires sont passés du répertoire `/usr/sbin/` au répertoire `/usr/bin/`. Par conséquent, les scripts disposant de chemins d'accès absolus pour ces binaires ne fonctionneront pas.

- *Remplacement de la commande `dbmmanage`* — La commande `dbmmanage` a été remplacée par `htdbm`.
- *Le fichier de configuration `logrotate` a été renommé* — Le nom du fichier de configuration `logrotate` a été changé de `/etc/logrotate.d/apache` à `/etc/logrotate.d/httpd`.

12.2.1 Configuration de l'environnement global

La section intitulée Environnement global (Global Environment) du fichier de configuration contient des directives qui modifient tout le fonctionnement du Serveur HTTP Apache, comme par exemple, le nombre de requêtes simultanées qu'il peut traiter et les emplacements des divers fichiers. Cette section nécessite un grand nombre de changements et doit être basée sur le fichier de configuration du Serveur HTTP Apache version 2.0 vers lequel tous les anciens paramètres devront être migrés.

12.2.1.1. Liaison des interfaces et des ports

Les directives `BindAddress` et `Port` n'existent plus ; leur fonctionnalité est désormais fournie par une directive plus flexible nommée `Listen`.

Si `Port 80` figurant dans les paramtres du fichier de configuration version 1.3, il est nécessaire de le remplacer par `Listen 80` dans le fichier de configuration version 2.0. Si `Port` avait une valeur *autre que 80*, il est nécessaire d'ajouter le numéro du port au contenu de la directive `ServerName`.

L'extrait ci-dessous représente un exemple de directive du Serveur HTTP Apache version 1.3 :

```
Port 123
ServerName www.example.com
```

Pour transférer ce paramètre vers le Serveur HTTP Apache 2.0, utilisez la structure suivante :

```
Listen 123
ServerName www.example.com:123
```

Pour obtenir davantage d'informations sur le sujet, reportez-vous à la documentation de l'organisation Apache Software Foundation disponible sur le site Web à :

- `http://httpd.apache.org/docs-2.0/mod/mpm_common.html#listen`
- `http://httpd.apache.org/docs-2.0/mod/core.html#servername`

12.2.1.2. Régulation de la taille de server-pool

Lorsque le Serveur HTTP Apache accepte les requêtes, il envoie des processus enfants ou des threads pour les traiter. Ce groupe de processus enfants ou de threads (aussi appelés fils) est appelé un *server-pool* (groupe de serveurs). Avec la version 2.0 du Serveur HTTP Apache, la responsabilité de la création et de la maintenance de ces groupes dépendait d'un groupe de modules appelés *Modules multitache* (ou *MPM*, de l'anglais Multi-Processing Modules).

Contrairement à d'autres modules, seul un module du groupe MPM peut être chargé par le Serveur HTTP Apache. Trois modules MPM sont inclus dans la version 2.0, à savoir, `prefork`, `worker` et `perchild`. Seuls les MPM `prefork` et `worker` sont actuellement disponibles, mais il se peut que le MPM `perchild` soit disponible dans le futur.

Le comportement original du Serveur HTTP Apache 1.3 a été déplacé dans le MPM `prefork`. Ce dernier accepte les mêmes directives que le Serveur HTTP Apache version 1.3, il est donc possible de migrrer les directives suivantes :

- `StartServers`
- `MinSpareServers`
- `MaxSpareServers`
- `MaxClients`
- `MaxRequestsPerChild`

Le MPM `worker` implémente un serveur multitâche, multiprocessus offrant une modulabilité plus importante. Lors de l'utiliseation de ce MPM, les requêtes sont manipulées par des threads (ou fils) économisant donc les ressources système et permettant ainsi à un grand nombres de requêtes d'être servi de façon efficace. Bien que certaines directives acceptées par le MPM `worker` soient les mêmes que celles acceptées par le MPM `prefork`, les valeurs de ces directives ne devraient pas être transférées directement depuis une installation de Serveur HTTP Apache 1.3. Il vaut mieux utiliser les valeurs par défaut comme des recommandations générale et d'expérimenter ensuite afin de déterminer les valeurs qui fonctionnent le mieux dans votre situation particulière.

Important

Pour utiliser le MPM `worker`, créez le fichier `/etc/sysconfig/httpd` et ajoutez-y la directive suivante :

```
HTTPD=/usr/sbin/httpd.worker
```

Pour obtenir davantage d'informations sur le sujet, reportez-vous à la documentation de l'organisation Apache Software Foundation disponible sur le site Web à :

- http://httpd.apache.org/docs-2.0/mpm.html

12.2.1.3. Prise en charge de DSO (Dynamic Shared Object)

Un grand nombre de changements étant nécessaire ici, il est vivement recommandé à toute personne essayant de modifier une configuration du Serveur HTTP Apache version 1.3 pour l'adapter à la version 2.0 (au lieu de migrer les changements vers la configuration version 2.0) de copier cette section du fichier de configuration Serveur HTTP Apache 2.0.

Les utilisateurs ne souhaitant pas copier la section de la configuration du Serveur HTTP Apache version 2.0 devraient prendre note des informations suivantes :

- Les directives `AddModule` et `ClearModuleList` n'existent plus. Elles étaient utilisées pour assurer l'activation des modules dans la bon ordre. L'API du Serveur HTTP

Apache 2.0 API permet aux modules de préciser leur d'activation, éliminant ainsi la raison d'être de ces deux directives.

- L'ordre des lignes de `LoadModule` n'est désormais plus important dans la plupart des cas.
- De nombreux modules ont été ajoutés, supprimés, renommés, divisés ou incorporés les uns aux autres.
- Les lignes `LoadModule` des modules intégrés dans leurs propres RPM (`mod_ssl`, `php`, `mod_perl` et autres) ne sont plus nécessaires puisqu'elles se trouvent dans leur fichier propre inclus dans le répertoire `/etc/httpd/conf.d/`.
- Les diverses définitions `HAVE_XXX` ne sont plus définies.

Important

Lors de la modification du fichier original, notez que le fichier `httpd.conf` doit absolument contenir la directive suivante :

```
Include conf.d/*.conf
```

L'oubli de cette directive entraînerait l'échec de tous les modules contenus dans leurs propres RPM (tels que `mod_perl`, `php` et `mod_ssl`).

12.2.1.4. Autres changements liés à l'environnement global

Les directives suivantes ont été supprimées de la configuration du Serveur HTTP Apache 2.0 :

- *`ServerType`* — Le Serveur HTTP Apache ne peut être exécuté qu'en tant que `ServerType standalone`, rendant ainsi cette directive inutile.
- *`AccessConfig`* et *`ResourceConfig`* — Ces directives ont été supprimées puisqu'elles reflétaient la fonctionnalité de la directive `Include`. Si les directives `AccessConfig` et `ResourceConfig` sont définies, il est nécessaire de les remplacer par des directives `Include`.

  Pour obtenir l'assurance que les fichiers seront lus dans l'ordre désigné par les anciennes directives, il est nécessaire de placer les directives `Include` à la fin du fichier `httpd.conf`, en prenant bien soin de placer celle correspondant à `ResourceConfig` avant celle correspondant à `AccessConfig`. Si les valeurs par défaut sont utilisées, elles doivent être incluses explicitement dans les fichiers `conf/srm.conf` et `conf/access.conf`.

12.2.2 Configuration du serveur principal

La section relative à la configuration du serveur principal du fichier de configuration installe le serveur principal qui répond à toute les requêtes non-traitées par un hôte virtuel définit dans un conteneur `<VirtualHost>`. Des valeurs spécifiées ici offrent aussi des valeurs par défaut pour tous les fichiers conteneurs `<VirtualHost>` définis.

Les directives utilisées dans cette section ont été légèrement modifiées entre la version 1.3 du Serveur HTTP Apache et la version 2.0. Si la configuration du serveur principal a un niveau élevé personnalisation, il sera peut-être plus simple de modifier le fichier de configuration

existant pour l'adapter au Serveur HTTP Apache 2.0. Les utilisateurs ayant une configuration peu personnalisée devraient migrer leurs changements vers la configuration 2.0 par défaut.

12.2.2.1. Mappage de UserDir

La directive `UserDir` est utilisée pour permettre à des URL telles que `http://example.com/~bob/` de se mapper à un sous-répertoire au sein du répertoire personnel de l'utilisateur `bob`, comme par exemple `/home/bob/public_html`. Cette particularité permettant à un éventuel agresseur de déterminer si un nom d'utilisateur donné est présent sur le système, la configuration par défaut pour le Serveur HTTP Apache 2.0 désactive cette directive.

Pour activer le mappage de `UserDir`, changez la directive figurant dans le fichier `httpd.conf` de :

```
UserDir disable
```

à :

```
UserDir public_html
```

Pour obtenir davantage d'informations sur le sujet, reportez-vous à la documentation de l'organisation Apache Software Foundation disponible sur le site Web à :

- http://httpd.apache.org/docs-2.0/mod/mod_userdir.html#userdir

12.2.2.2. Journalisation

Les directives de journalisation suivantes ont été supprimées :

- `AgentLog`
- `RefererLog`
- `RefererIgnore`

Cependant, les journaux Agent et Referrer sont encore disponibles en utilisant les directives `CustomLog` et `LogFormat`.

Pour obtenir davantage d'informations sur le sujet, reportez-vous à la documentation de l'organisation Apache Software Foundation disponible sur le site Web à :

- http://httpd.apache.org/docs-2.0/mod/mod_log_config.html#customlog
- http://httpd.apache.org/docs-2.0/mod/mod_log_config.html#logformat

12.2.2.3. Indexation des répertoires

La directive `FancyIndexing` étant désormais obsolète a été supprimée. Cette même fonctionnalité est toutefois encore disponible par le biais de l'*option* `FancyIndexing` à l'intérieur de la directive `IndexOptions`.

La nouvelle option `VersionSort` appliquée à la directive `IndexOptions` permet de classer dans un ordre plus naturel les fichiers contenant des numéros de version. Par exemple, `httpd-2.0.6.tar` apparaît avant `httpd-2.0.36.tar` dans une page d'index de répertoires.

Les paramètres par défaut pour les directives `ReadmeName` et `HeaderName` ont été transférés des fichiers `README` et `HEADER` vers les fichiers `README.html` et `HEADER.html`.

Pour obtenir davantage d'informations sur le sujet, reportez-vous à la documentation de l'organisation Apache Software Foundation disponible sur le site Web à :

- http://httpd.apache.org/docs-2.0/mod/mod_autoindex.html#indexoptions
- http://httpd.apache.org/docs-2.0/mod/mod_autoindex.html#readmename
- http://httpd.apache.org/docs-2.0/mod/mod_autoindex.html#headername

12.2.2.4. Négociation du contenu

La directive `CacheNegotiatedDocs` retient désormais les critères `on` ou `off`. Les cas existants de `CacheNegotiatedDocs` devront être remplacés par `CacheNegotiatedDocs on`.

Pour obtenir davantage d'informations sur le sujet, reportez-vous à la documentation de l'organisation Apache Software Foundation disponible sur le site Web à :

- http://httpd.apache.org/docs-2.0/mod/mod_negotiation.html#cachenegotiateddocs

12.2.2.5. Documents d'erreur

Afin de pouvoir utiliser un message codé en dur avec la directive `ErrorDocument`, le message doit apparaître entre guillemets (["]), plutôt que d'être seulement précédé par des guillemets, comme c'était la cas avec le Serveur HTTP Apache 1.3.

L'extrait ci-dessous représente un exemple de directive du Serveur HTTP Apache version 1.3 :

```
ErrorDocument 404 "The document was not found
```

Pour transférer un paramètre `ErrorDocument` vers le Serveur HTTP Apache 2.0, utilisez la structure suivante :

```
ErrorDocument 404 "The document was not found"
```

Notez bien la présence des guillemets à la fin de l'exemple de directive `ErrorDocument` reproduit ci-dessus.

Pour obtenir davantage d'informations sur le sujet, reportez-vous à la documentation de l'organisation Apache Software Foundation disponible sur le site Web à :

- http://httpd.apache.org/docs-2.0/mod/core.html#errordocument

12.2.3 Modules et Serveur HTTP Apache 2.0

Dans la version 2.0 du Serveur HTTP Apache, le système de modules a été modifié afin de permettre aux modules d'être désormais liés ensemble ou combinés de plusieurs manières différentes. Les scripts *CGI*(de l'anglais *Common Gateway Interface*) par exemple, sont capables de générer des documents HTML analysés par le serveur qui peuvent ensuite être traités par `mod_include`. Grâce à ce développement, il existe désormais de nombreuses possibilités de combinaison des modules afin d'atteindre un objectif spécifique.

Cette situation est possible car chaque requête est servie par un seul module *handler*, suivi d'aucun ou de plusieurs modules *filter*.

Sous la version 1.3 du Serveur HTTP Apache par exemple, un script Perl serait traité dans son intégralité par le module Perl (`mod_perl`). Sous la version 2.0 du Serveur HTTP Apache, en revanche, la requête est initialement *traitée* par le module principal— qui sert des fichiers statiques — et est ensuite *filtrée* par `mod_perl`.

L'explication exacte de l'utilisation de cette fonction particulière et de toutes les autres nouvelles fonctions du Serveur HTTP Apache 2.0, va bien au-delà de la portée de ce document ; toutefois, la conversion a des ramifications non-négligeables si vous avez utilisé la directive `PATH_INFO` pour un document traité par un module qui est désormais traité comme un filtre étant donné que chaque directive contient des informations de chemin peu importantes après le vrai nom de fichier. Le module mémoire, qui traite initialement la requête, ne comprend pas par défaut `PATH_INFO` et renverra des erreurs de types `404 Not Found` pour les requêtes qui contiennent de telles informations. Vous pouvez également utiliser la directive `AcceptPathInfo` pour obliger le module mémoire à accepter les requêtes contenant `PATH_INFO`.

Ci-dessous figure un exemple de cette directive :

```
AcceptPathInfo on
```

Pour obtenir davantage d'informations sur le sujet, reportez-vous à la documentation de l'organisation Apache Software Foundation disponible sur le site Web à :

- http://httpd.apache.org/docs-2.0/mod/core.html#acceptpathinfo
- http://httpd.apache.org/docs-2.0/handler.html
- http://httpd.apache.org/docs-2.0/filter.html

12.2.3.1. Module suexec

Dans la version du Serveur HTTP Apache 2.0, le module `mod_suexec` utilise la directive `SuexecUserGroup` (plutôt que les directives `User` et `Group`) pour la configuration des hôtes virtuels. Les directives `User` et `Group` peuvent toujours être utilisées d'une manière générale mais ne peuvent plus être utilisées pour la configuration des hôtes virtuels.

L'extrait ci-dessous représente un exemple de directive du Serveur HTTP Apache version 1.3 :

```
<VirtualHost vhost.example.com:80>
    User someone
    Group somegroup
</VirtualHost>
```

Pour transférer ce paramètre vers le Serveur HTTP Apache 2.0, utilisez la structure suivante :

```
<VirtualHost vhost.example.com:80>
    SuexecUserGroup someone somegroup
</VirtualHost>
```

12.2.3.2. Module mod_ssl

La configuration de mod_ssl a été transférée du fichier httpd.conf au fichier /etc/httpd/conf.d/ssl.conf. Pour que ce dernier soit chargé et que mod_ssl puisse fonctionner, la déclaration Include conf.d/*.conf doit figurer dans le fichier httpd.conf.

Les directives ServerName dans les hôtes virtuels avec SSL doivent explicitement spécifier le numéro de port.

L'extrait ci-dessous représente un exemple de directive du Serveur HTTP Apache version 1.3 :

```
<VirtualHost _default_:443>
    # General setup for the virtual host
    ServerName ssl.example.name
    ...
</VirtualHost>
```

Pour transférer ce paramètre vers le Serveur HTTP Apache 2.0, utilisez la structure suivante :

```
<VirtualHost _default_:443>
    # General setup for the virtual host
    ServerName ssl.host.name:443
    ...
</VirtualHost>
```

Il est également important de noter que les deux directives SSLLog et SSLLogLevel ont été supprimées. Le module mod_ssl obéit désormais aux directives ErrorLog et LogLevel.

Pour obtenir davantage d'informations sur le sujet, reportez-vous à la documentation de l'organisation Apache Software Foundation disponible sur le site Web à :

- http://httpd.apache.org/docs-2.0/mod/mod_ssl.html
- http://httpd.apache.org/docs-2.0/vhosts/

12.2.3.3. Module mod_proxy

Les instructions relatives au contrôle de l'accès proxy sont maintenant placées dans un bloc <Proxy> plutôt que dans un répertoire <Directory proxy:>.

La fonctionnalité de stockage temporaire de l'ancien `mod_proxy` a été divisée en trois modules, à savoir :

- `mod_cache`
- `mod_disk_cache`
- `mod_mem_cache`

Ceux-ci utilisent généralement des directives similaires aux versions plus anciennes du module `mod_proxy`. Il est cependant conseillé de vérifier chaque directive avant de migrer tout paramètre de cache.

Pour obtenir davantage d'informations sur le sujet, reportez-vous à la documentation de l'organisation Apache Software Foundation disponible sur le site Web à :

- http://httpd.apache.org/docs-2.0/mod/mod_proxy.html

12.2.3.4. Module `mod_include`

Le module `mod_include` fonctionnant désormais comme un filtre, il est activé d'une façon différente.

L'extrait ci-dessous représente un exemple de directive du Serveur HTTP Apache version 1.3 :

```
AddType text/html .shtml
AddHandler server-parsed .shtml
```

Pour transférer ce paramètre vers le Serveur HTTP Apache 2.0, utilisez la structure suivante :

```
AddType text/html .shtml
AddOutputFilter INCLUDES .shtml
```

Notez que, comme auparavant, la directive `Options +Includes` est toujours nécessaire pour le conteneur `<Directory>` ou dans un fichier `.htaccess`.

Pour obtenir davantage d'informations sur le sujet, reportez-vous à la documentation de l'organisation Apache Software Foundation disponible sur le site Web à :

- http://httpd.apache.org/docs-2.0/mod/mod_include.html

12.2.3.5. Modules `mod_auth_dbm` et `mod_auth_db`

Le Serveur HTTP Apache 1.3 prenait en charge deux modules d'authentification, à savoir, `mod_auth_db` et `mod_auth_dbm`, qui utilisaient respectivement les bases de données Berkeley et DBM. Ces modules ont été rassemblés dans un seul module nommé `mod_auth_dbm` dans la version 2.0 du Serveur HTTP Apache, pouvant accéder à plusieurs formats de base de données différents. Pour effectuer une migration depuis le fichier `mod_auth_db`, il est nécessaire de modifier les fichiers de configuration en remplaçant `AuthDBUserFile` et `AuthDBGroupFile` par les équivalents de `mod_auth_dbm` à savoir `AuthDBMUserFile` et

`AuthDBMGroupFile`. Il est également nécessaire d'ajouter la directive `AuthDBMType DB`pour préciser le type de fichier de base de données utilisé.

Ci-dessous figure un exemple de configuration `mod_auth_db` pour le Serveur HTTP Apache 1.3 :

```
<Location /private/>
  AuthType Basic
  AuthName "My Private Files"
  AuthDBUserFile /var/www/authdb
  require valid-user
</Location>
```

Pour transférer ce paramètre vers la version 2.0 du Serveur HTTP Apache, utilisez la structure suivante :

```
<Location /private/>
  AuthType Basic
  AuthName "My Private Files"
  AuthDBMUserFile /var/www/authdb
  AuthDBMType DB
  require valid-user
</Location>
```

Notez que la directive `AuthDBMUserFile` peut également être utilisée dans des fichiers `.htaccess`.

Dans la version 2.0 du Serveur HTTP Apache, le script Perl `dbmmanage` utilisé pour manipuler les bases de données des noms d'utilisateur et mots de passe a été remplacé par `htdbm`. Le programme `htdbm` offre des fonctionnalités équivalentes et, tout comme le module `mod_auth_dbm`, peut exploiter une grande variété de formats de bases de données ; l'option `-T` peut être utilisée sur la ligne de commande pour spécifier le format à utiliser.

Le Tableau 12-1 montre comment migrer d'un format de base de données DBM vers `htdbm` en utilisant `dbmmanage`.

Action	**Commande dbmmanage (1.3)**	**Équivalent de la commande htdbm (2.0)**
Ajoute l'utilisateur à la base de données (en utilisant le mot de passe donné)	`dbmmanage authdb add username password`	`htdbm -b -TDB authdb username password`
Ajoute l'utilisateur à la base de données (invite à fournir le mot de passe)	`dbmmanage authdb adduser username`	`htdbm -TDB authdb username`
Retire l'utilisateur de la base de	`dbmmanage authdb`	`htdbm -x -TDB authdb`

Action	Commande dbmmanage (1.3)	Équivalent de la commande htdbm (2.0)
données	`delete username`	`username`
Répertorie les utilisateurs dans la base de données	`dbmmanage authdb view`	`htdbm -l -TDB authdb`
Vérifie un mot de passe	`dbmmanage authdb check username`	`htdbm -v -TDB authdb username`

Tableau 12-1. Migration de `dbmmanage` vers `htdbm`

Les options `-m` et `-s` fonctionnant avec `dbmmanage` et `htdbm`, il est possible d'utiliser les algorithmes MD5 ou SHA1 respectivement, pour le hachage des mots de passe.

Lors de la création d'une nouvelle base de données avec `htdbm`, il est nécessaire d'utiliser l'option `-c`.

Pour obtenir davantage d'informations sur le sujet, reportez-vous à la documentation de l'organisation Apache Software Foundation disponible sur le site Web à :

- http://httpd.apache.org/docs-2.0/mod/mod_auth_dbm.html

12.2.3.6. Module `mod_perl`

La configuration du module `mod_perl` a été transférée du fichier `httpd.conf` au fichier `/etc/httpd/conf.d/perl.conf`. Pour que ce fichier soit chargé et permette ainsi à `mod_perl` de fonctionner correctement, la déclaration `Include conf.d/*.conf` doit figurer dans le fichier `httpd.conf`.

Les occurrences de `Apache::` contenues dans votre fichier `httpd.conf` doivent être remplacées par `ModPerl::`. En outre, la façon dont les gestionnaires de signaux (ou handlers) sont enregistrés a été modifiée.

Ci-dessous figure un exemple de la configuration du Serveur HTTP Apache 1.3 pour le module `mod_perl` :

```
<Directory /var/www/perl>
    SetHandler perl-script
    PerlHandler Apache::Registry
    Options +ExecCGI
</Directory>
```

Ci-après se trouve le module `mod_perl` équivalent pour la version 2.0 du Serveur HTTP Apache :

```
<Directory /var/www/perl>
    SetHandler perl-script
    PerlResponseHandler ModPerl::Registry
    Options +ExecCGI
</Directory>
```

La plupart des modules pour `mod_perl` 1.x devraient fonctionner sans modification, avec `mod_perl` 2.x. Les modules XS nécessitent une recompilation et des modifications mineures de Makefile seront peut-être également nécessaires.

12.2.3.7. Module `mod_python`

La configuration du module `mod_python` a été transférée du fichier `httpd.conf` au fichier `/etc/httpd/conf.d/python.conf`. Pour que ce dernier soit chargé et permette ainsi à `mod_python` de fonctionner correctement, la déclaration `Include conf.d/*.conf` doit figurer dans le fichier `httpd.conf`.

12.2.3.8. PHP

La configuration de PHP a été transférée du fichier `httpd.conf` au fichier `/etc/httpd/conf.d/php.conf`. Pour que celui-ci soit chargé, la déclaration `Include conf.d/*.conf` doit figurer dans le fichier `httpd.conf`.

> **Remarque**
>
> Toutes les directives de configuration de PHP utilisées avec le Serveur HTTP Apache 1.3 sont désormais entièrement compatibles, lors de la migration vers le Serveur HTTP Apache 2.0 sur Red Hat Enterprise Linux 4.

Dans PHP 4.2.0 et les versions postérieures, l'ensemble des variables par défaut qui sont prédéfinies et ont généralement une portée globale, a changé. Les variables d'entrée individuelle et les variables de serveur ne sont plus, par défaut, directement placées dans la portée globale. Ce changement risque d'interrompre les scripts. Revenez à l'ancien comportement en réglant `register_globals` sur `On` dans le fichier `/etc/php.ini`.

Pour obtenir davantage d'informations sur le sujet et pour obtenir des renseignements détaillés sur les changements au niveau de la portée globale, reportez-vous à l'URL suivante :

- http://www.php.net/release_4_1_0.php

12.2.3.9. Module `mod_authz_ldap`

Red Hat Enterprise Linux est fournit avec le module `mod_authz_ldap` pour le Serveur HTTP Apache. Ce module utilise le nom raccourci du nom distinct (ou distinguished name) d'un sujet et de l'émetteur du certificat SSL client afin de déterminer le nom distinct de l'utilisateur au sein d'un répertoire LDAP. Il est également à même d'effectuer les tâches suivantes : autoriser un utilisateur en fonction des attributs de l'entrée du répertoire LDAP de cet utilisateur, déterminer l'accès aux ressources en fonction des privilèges octroyés aux utilisateurs et aux groupes quant à ces ressources et peut également refuser l'accès aux

utilisateurs dont le mot de passe a expiré. Le module `mod_ssl` est nécessaire lorsque le module `mod_authz_ldap` est utilisé.

Important

Le module `mod_authz_ldap` n'effectue pas l'authentification d'un utilisateur par rapport à un répertoire LDAP au moyen d'un hachage de mots de passe cryptés. Cette fonctionnalité est fournie par le module expérimental `mod_auth_ldap` qui ne fait pas partie de Red Hat Enterprise Linux. Pour obtenir de plus amples informations sur le statut de ce module, reportez-vous au site Web de l'organisation Apache Software Foundation qui se trouve à l'adresse suivante : http://www.apache.org/.

Le fichier `/etc/httpd/conf.d/authz_ldap.conf` configure le module `mod_authz_ldap`.

Pour obtenir de plus amples informations sur la configuration du module tiers `mod_authz_ldap`, reportez-vous au fichier `/usr/share/doc/mod_authz_ldap-<version>/index.html` (en prenant soin de remplacer `<version>` par le numéro de version du paquetage).

12.3 Après l'installation

Après l'installation du paquetage `httpd`, passez en revue la documentation du Serveur HTTP Apache disponible en ligne à l'adresse suivante : http://httpd.apache.org/docs-2.0/.

La documentation du Serveur HTTP Apache contient une liste exhaustive de toutes les options de configuration accompagnée d'une description complète. Ce chapitre fournit de brèves descriptions des directives de configuration utilisées par le Serveur HTTP Apache 2.0.

La version 2.0 du Serveur HTTP Apache offre la possibilité de définir des serveurs Web sécurisés au moyen du fort cryptage SSL offert par les paquetages `mod_ssl` et `openssl`. Notez que le fichier de configuration contient aussi bien un serveur Web non-sécurisé qu'un serveur Web sécurisé. Le serveur Web sécurisé fonctionne comme un hôte virtuel configuré dans le fichier `/etc/httpd/conf.d/ssl.conf`.

12.4 Démarrage et arrêt de `httpd`

Le RPM `httpd` installe le script `/etc/init.d/httpd` qui est accessible à l'aide de la commande `/sbin/service`.

Pour démarrer le serveur, saisissez la commande suivante en étant connecté en tant que super-utilisateur :

```
/sbin/service httpd start
```

Pour arrêter le serveur, saisissez la commande suivante en étant connecté en tant que super-utilisateur :

```
/sbin/service httpd stop
```

L'option `restart` est une façon rapide d'arrêter et de redémarrer le Serveur HTTP Apache.

Pour redémarrer le serveur, saisissez la commande suivante en étant connecté en tant que super-utilisateur :

```
/sbin/service httpd restart
```

Remarque

Lors de l'exécution du Serveur HTTP Apache en tant que serveur sécurisé, il est nécessaire de saisir le mot de passe du serveur chaque fois que des options `start` ou `restart` sont utilisées.

Après avoir modifié le fichier `httpd.conf`, il n'est toutefois pas nécessaire d'arrêter et de redémarrer le serveur. Utilisez à la place l'option `reload` qui entraînera un rechargement du fichier.

Afin de recharger le fichier de configuration, saisissez la commande suivante en étant connecté en tant que super-utilisateur :

```
/sbin/service httpd reload
```

Remarque

Lors de l'exécution du Serveur HTTP Apache en tant que serveur sécurisé, vous *n'aurez pas* besoin de saisir votre mot de passe lors de l'utilisation de l'option `reload` (recharger).

Par défaut, le service `httpd` ne se lancera *pas* automatiquement au démarrage. Pour configurer le service `httpd` de manière à ce qu'il soit lancé au démarrage, utilisez un utilitaire de script d'initialisation (ou initscript) tel que `/sbin/chkconfig`, **/sbin/ntsysv** ou l'**Outil de configuration des services**.

Remarque

Lors de exécution du Serveur HTTP Apache en tant que serveur sécurisé, le mot de passe de ce dernier doit être saisi après le démarrage de l'ordinateur lorsqu'une clé SSL privée et cryptée est utilisée.

Pour toute information sur l'installation d'un serveur sécurisé HTTP Apache, reportez-vous au chapitre intitulé *Configuration du serveur sécurisé HTTP Apache* du *Guide d'administration système de Red Hat Enterprise Linux.*

12.5 Directives de configuration dans `httpd.conf`

Le fichier de configuration du Serveur HTTP Apache est `/etc/httpd/conf/httpd.conf`. Dans le fichier `httpd.conf` figurent de nombreux commentaires qui rendent sont contenu très explicite. La configuration par défaut fonctionne dans la plupart des situations ; cependant, il est important de bien connaître certaines des options de configuration les plus importantes.

12.5.1 Astuces générales de configuration

Lors de la configuration du Serveur HTTP Apache, modifiez `/etc/httpd/conf/httpd.conf` puis rechargez, redémarrez ou arrêtez le processus `httpd`.

Avant de modifier `httpd.conf`, faites une copie de sauvegarde du fichier original. Ainsi, si vous commettez une erreur lors de la modification du fichier de configuration, vous pourrez toujours utiliser la copie de sauvegarde pour résoudre d'éventuels problèmes.

Si une erreur est commise et que le serveur Web ne fonctionne pas correctement, passez d'abord en revue les passages modifiés du fichier `httpd.conf` afin de corriger toute faute de frappe.

Consultez ensuite le journal d'erreurs du serveur Web, `/var/log/httpd/error_log`. Selon votre expérience, le journal d'erreurs peut paraître quelque peu difficile à interpréter. Ceci étant, les dernières entrées du journal d'erreurs devraient fournir des informations utiles.

Les sections suivantes contiennent de brèves descriptions des directives contenues dans le fichier `httpd.conf`. Ces descriptions ne sont pas exhaustives. Pour obtenir de plus amples informations, reportez-vous à la documentation de l'organisation Apache disponible en ligne à l'adresse suivante : http://httpd.apache.org/docs-2.0/.

Pour obtenir davantage d'informations sur les directives `mod_ssl`, reportez-vous à la documentation disponible en ligne à l'adresse suivante : http://httpd.apache.org/docs-2.0/mod/mod_ssl.html.

12.5.2 `ServerRoot`

Le répertoire `ServerRoot` est le répertoire de niveau supérieur contenant les fichiers du serveur. Par défaut, la directive `ServerRoot` est paramétrée sur `"/etc/httpd"` aussi bien pour le serveur sécurisé que pour le serveur non-sécurisé.

12.5.3 `PidFile`

`PidFile` est le nom du fichier dans lequel le serveur enregistre son identifiant de processus (PID). Le PID par défaut est `/var/run/httpd.pid`.

12.5.4 `Timeout`

`Timeout` définit la durée, exprimée en secondes, pendant laquelle le serveur attend des réceptions et des émissions pendant les communications. La valeur de `Timeout` est paramétrée sur `300` secondes par défaut, ce qui est approprié pour la plupart des situations.

12.5.5 `KeepAlive`

`KeepAlive` définit si votre serveur autorisera plus d'une requête par connexion ; cette directive peut servir à empêcher un client particulier d'utiliser une trop grande quantité des ressources dont le serveur est doté.

Par défaut, la valeur de `Keepalive` est réglée sur `off`. Si la valeur de `Keepalive` est `on` et que le serveur devient très occupé, il peut générer rapidement le nombre maximum de processus enfants. Dans ce cas, le serveur sera considérablement ralenti. Si la directive `Keepalive` est activée, il est recommandé de donner à `KeepAliveTimeout` une valeur basse et de contrôler le fichier journal `/var/log/httpd/error_log` du serveur. Ce journal indique si le serveur est sur le point d'atteindre le maximum de processus enfants.

12.5.6 `MaxKeepAliveRequests`

Cette directive définit le nombre maximum de requêtes autorisées par connexion persistante. L'organisation Apache Project recommande l'utilisation d'un paramétrage élevé, ce qui entraîne une amélioration des performances du serveur. Par défaut, la valeur de `MaxKeepAliveRequests` paramétrée sur `100` est approprié pour la plupart des situations.

12.5.7 `KeepAliveTimeout`

`KeepAliveTimeout` définit la durée exprimée en secondes pendant laquelle le serveur attend après avoir servi une requête, avant d'interrompre la connexion. Une fois que le serveur reçoit une requête, c'est la directive `Timeout` qui s'applique à sa place. Par défaut, la valeur donnée à la directive `KeepAliveTimeout` est de 15 secondes.

12.5.8 `IfModule`

Les balises `<IfModule>` et `</IfModule>` créent un conteneur conditionnel dont les directives ne sont activées que si le module spécifié est chargé. Les directives placées entre les balises `IfModule` sont traitées dans l'un des deux cas suivants. Les directives sont traitées si le module contenu dans la balise de début `<IfModule>` est chargé. En revanche, si un point d'exclamation ([!]) figure devant le nom du module, les directives ne sont traitées que si le module contenu dans la balise `<IfModule>` n'est *pas* chargé.

12.5.9 Directives de server-pool spécifiques aux MPM

Sous le Serveur HTTP Apache 2.0, la responsabilité de gérer les caractéristiques de server-pool est octroyée à un groupe de modules appelé MPM. Les caractéristiques de server-pool sont différentes selon le MPM spécifique utilisé. C'est la raison pour laquelle un conteneur `IfModule` est nécessaire pour définir le server-pool du MPM qui est utilisé.

Par défaut, le Serveur HTTP Apache 2.0 définit le server-pool aussi bien pour le MPM `prefork` que le MPM `worker`.

Ci-dessous figure une liste des directives figurant dans les conteneurs de server-pool spécifiques aux MPM.

12.5.9.1. `StartServers`

La directive `StartServers` définit le nombre de processus serveur créés au démarrage. Étant donné que le serveur Web supprime et crée dynamiquement des processus serveur en fonction de la charge du trafic, il n'est pas nécessaire de modifier ce paramètre. Le serveur Web est

configuré de manière à lancer 8 processus serveur au démarrage pour le MPM prefork et 2 pour le MPM worker.

12.5.9.2. MaxRequestsPerChild

MaxRequestsPerChild définit le nombre total de requêtes que chaque processus serveur enfant sert avant de s'arrêter. L'attribution d'une valeur à MaxRequestsPerChild est importante afin d'éviter des pertes de mémoire induites par des processus longs. La valeur par défaut de MaxRequestsPerChild pour le MPM prefork est 4000 et 0 pour le MPM worker.

12.5.9.3. MaxClients

MaxClients fixe une limite au nombre total de processus serveur ou de clients connectés simultanément qui peuvent s'exécuter en même temps. L'objectif principal de cette directive est d'éviter qu'un Serveur HTTP Apache surchargé n'entraîne le plantage de votre système d'exploitation. Pour des serveurs très solicités, cette valeur devrait être élevée. La valeur par défaut du serveur est 150, indépendamment du MPM utilisé. Toutefois, il n'est pas recommandé d'attribuer à MaxClients une valeur supérieure à 256 lors de l'utilisation du MPM prefork.

12.5.9.4. MinSpareServers and MaxSpareServers

Ces valeurs ne sont pas seulement utilisées avec le MPM prefork. Elles ajustent la façon selon laquelle le Serveur HTTP Apache s'adapte dynamiquement à la charge reçue en maintenant un nombre approprié de processus serveur de secours déterminiés en fonction du nombre de requêtes entrantes. Le serveur vérifie le nombre de serveurs attendant une requête et en supprime certains s'ils sont plus nombreux que MaxSpareServers ou en crée d'autres s'ils sont moins nombreux que MinSpareServers.

La valeur par défaut donnée à MinSpareServers est 5 ; la valeur par défaut attribuée à MaxSpareServers est 20. Ces paramètres par défaut devraient être adaptés à presque toutes les situations. Ne donnez pas à MinSpareServers une valeur très élevée car un tel choix se traduira en une charge de traitement importante sur le serveur, même si le trafic est faible.

12.5.9.5. MinSpareThreads et MaxSpareThreads

Ces valeurs ne sont pas seulement utilisées avec le MPM worker. Elles ajustent la façon selon laquelle le Serveur HTTP Apache s'adapte dynamiquement à la charge reçue en maintenant un nombre approprié de processus serveur de secour déterminés en fonction du nombre de requêtes entrantes. Le serveur vérifie le nombre de threads de serveurs attendant une requête et en supprime certains s'ils sont plus nombreux que MaxSpareThreads ou en crée d'autres s'ils sont moins nombreux que MinSpareThreads.

La valeur par défaut donnée à MinSpareThreads est 25 ; la valeur par défaut attribuée à MaxSpareThreads est 75. Ces paramètres par défaut devraient être appropriés à la plupart des situations. La valeur de MaxSpareThreads doit être supérieure ou égale à la somme de MinSpareThreads et de ThreadsPerChild, dans le cas contraire, le Serveur HTTP Apache la corrigera automatiquement.

12.5.9.6. ThreadsPerChild

Cette valeur est utilisée uniquement avec le MPM `worker`. Il définit le nombre de threads (ou fils) au sein de chaque processus enfant. La valeur par défaut pour cette directive est `25`.

12.5.10 Listen

La commande `Listen` identifie les ports sur lesquels votre serveur Web acceptera les demandes entrantes. Par défaut, le Serveur HTTP Apache est paramétré pour écouter les communications Web non-sécurisées sur le port 80et (dans `/etc/httpd/conf.d/ssl.conf` définissant tout serveur sécurisé) les communications Web sécurisées sur le port 443.

Si le Serveur HTTP Apache est configuré pour écouter l'activité sur un port dont le numéro est inférieur à 1024, seul le super-utilisateur peut le lancer. En revanche, pour les ports dont le numéro est égal ou supérieur à 1024, `httpd` peut être lancée en tant que simple utilisateur.

La directive `Listen` peut également être utilisée pour spécifier des adresses IP particulières sur lesquelles le serveur acceptera des connexions.

12.5.11 Include

`Include` permet d'inclure d'autres fichiers de configuration au moment de l'exécution.

Le chemin d'accès vers ces fichiers de configuration peut être absolu ou relatif par rapport au `ServerRoot`.

> **Important**
>
> Pour que le serveur utilise individuellement des modules paquetés, tels que `mod_ssl`, `mod_perl` et `php`, la directive suivante doit être intégrée dans la `Section 1 : Global Environment` du `httpd.conf` :

```
Include conf.d/*.conf
```

12.5.12 LoadModule

`LoadModule` est utilisée pour charger des modules DSO (de l'anglais Dynamic Shared Object, objet partagé dynamiquement). Notez que l'ordre du chargement des modules *n'est plus important* avec le Serveur HTTP Apache 2.0.

12.5.13 ExtendedStatus

La directive `ExtendedStatus` spécifie si Apache doit produire des informations élémentaires (`off`) ou détaillées (`on`) sur l'état des serveurs, lorsque le gestionnaire de signal `server-status` est appelé. Ce dernier est appelé à l'aide des balises `Location`.

12.5.14 IfDefine

Les balises IfDefine entourent des directives de configuration. Elles s'appliquent si résultat du "test" spécifié dans la balise IfDefine est vrai. Les directives sont ignorées si le résultat du test est faux.

Le test dans les balises IfDefine est un nom de paramètre (comme par exemple, HAVE_PERL). Si le paramètre est défini (c'est-à-dire spécifié comme argument de la commande de démarrage du serveur), le test est vrai. Dans ce cas, lorsque le serveur Web est démarré, le test est vrai et les directives contenues dans les balises IfDefine sont appliquées.

12.5.15 SuexecUserGroup

La directive SuexecUserGroup figurant dans le module mod_suexec, permet de spécifier les privilèges d'exécution des programmes CGI qui s'applique à l'utilisateur et au groupe. Des requêtes non-CGI continuent à être traitées en fonction de l'utilisateur et du groupe spécifié dans les directives User et Group.

Remarque

La directive SuexecUserGroup remplace la configuration du Serveur HTTP Apache 1.3 configuration qui utilisait les directives User (utilisateur) et Group (Groupe) au sein de la configuration des sections VirtualHosts (Hôtes virtuels).

12.5.16 User

La directive User définit le nom d'utilisateur du processus serveur et détermine les fichiers auxquels le serveur peut avoir accès. Tous les fichiers auxquels cet utilisateur n'aura pas accès seront également inaccessibles aux clients se connectant au Serveur HTTP Apache.

La valeur par défaut donnée à User est apache.

Cette directive a été supprimée pour la configuration des hôtes virtuels.

Remarque

Pour des raisons de sécurité, le Serveur HTTP Apache n'est pas exécuté en tant que super-utilisateur.

12.5.17 Group

Spécifie le nom de groupe des processus du Serveur HTTP Apache.

Cette directive a été supprimée pour la configuration des hôtes virtuels.

La valeur par défaut attribuée à Group est apache.

12.5.18 ServerAdmin

Donnez comme valeur à la directive ServerAdmin l'adresse électronique de l'administrateur du serveur Web. Cette adresse électronique apparaîtra dans les messages d'erreur sur les pages Web générées par le serveur afin que les utilisateurs puissent signaler un problème en envoyant un message électronique à l'administrateur du serveur.

La valeur par défaut donnée à ServerAdmin est root@localhost.

Généralement, la valeur donnée à ServerAdmin est Webmaster@example.com. Une fois cette valeur déterminée, créez un alias pour Webmaster établi au nom de la personne responsable du serveur Web dans /etc/aliases et exécutez /usr/bin/newaliases.

12.5.19 ServerName

ServerName permet de définir un nom d'hôte et un numéro de port (en accord avec la directive Listen) pour le serveur. La directive ServerName ne doit pas forcément correspondre au nom d'hôte de l'ordinateur. Par exemple, le serveur Web pourrait être www.example.com bien que le nom d'hôte du serveur soit foo.example.com. La valeur spécifiée dans ServerName doit être un nom de domaine (ou DNS, de l'anglais Domain Name Service) valide qui peut être résolu par le système — ne vous contentez surtout pas d'en inventer un.

Ci-dessous figure un exemple de directive ServerName :

```
ServerName www.example.com:80
```

Lors de la détermination d'un ServerName, assurez-vous que son adresse IP et son nom de serveur figurent bien dans le fichier /etc/hosts.

12.5.20 UseCanonicalName

Lorsque la valeur attribuée à cette directive est on, elle configure le Serveur HTTP Apache de manière à ce qu'il se référence en utilisant les valeurs précisées dans les directives ServerName et Port. En revanche, lorsque la valeur de UseCanonicalName est off, le serveur emploie à la place la valeur utilisée par le client envoyant la requête lorsqu'il fait référence à lui-même.

Par défaut, la valeur attribuée à UseCanonicalName est off.

12.5.21 DocumentRoot

DocumentRoot est le répertoire contenant la plupart des fichiers HTML qui seront servis en réponse aux requêtes. Le défaut de DocumentRoot aussi bien pour le serveur Web sécurisé que pour le serveur Web non-sécurisé est le répertoire /var/www/html. Par exemple, il se peut que le serveur reçoive une demande pour le document suivant :

```
http://example.com/foo.html
```

Le serveur recherche le fichier suivant dans le répertoire par défaut :

```
/var/www/html/foo.html
```

12.5.22 Directory

Les balises `<Directory /path/to/directory>` et `</Directory>` créent un conteneur utilisé pour entourer un groupe de directives de configuration devant uniquement s'appliquer à ce répertoire et à ses sous-répertoires. Toute directive applicable à un répertoire peut être utilisée à l'intérieur de balises `Directory`.

Par défaut, des paramètres très restrictifs sont appliqués au répertoire racine (`/`), à l'aide des directives `Options` et `AllowOverride`. Sous une telle configuration, tout répertoire du système ayant besoin de paramètres plus permissifs doit contenir explicitement ces paramètres.

Dans la configuration par défaut, un autre conteneur `Directory` est également configuré pour `DocumentRoot` ; ce faisant, des paramètres moins rigides sont assignés à l'arborescence de répertoires, de manière à ce que le Serveur HTTP Apache puisse avoir accès à des fichiers placés dans ce dernier.

Le `répertoire` conteneur peut également être utilisé pour configurer des répertoires `cgi-bin` supplémentaires pour des applications côté-serveur en dehors du répertoire spécifié dans la directive `ScriptAlias`.

Pour ce faire, le conteneur `Directory` doit déterminer l'option `ExecCGI` pour ce répertoire.

Par exemple, si les scripts CGI se trouvent dans `/home/my_cgi_directory`, ajoutez le conteneur `Directory` suivant au fichier `httpd.conf` :

```
<Directory /home/my_cgi_directory>
    Options +ExecCGI
</Directory>
```

Ensuite, il est nécessaire d'enlever le symbole de commanetaire présent dans la directive `AddHandler` afin de permettre l'identification des fichiers ayant une extension `.cgi` en tant que scripts CGI.

Pour que cette opération se déroule parfaitement, il est nécessaire de définir les permissions pour les scripts CGI et pour le chemin d'accès complet vers les scripts en tant que 0755.

12.5.23 Options

La directive `Options` contrôle les fonctionnalités spécifiques du serveur qui sont disponibles dans un répertoire particulier. Par exemple, en vertu des paramètres restrictifs spécifiés pour le répertoire root, `Options` est réglée uniquement sur `FollowSymLinks`. Aucune fonctionnalité n'est activée, à l'exception du fait que le serveur est autorisé à suivre les liens symboliques dans le répertoire root.

Par défaut, dans le répertoire `DocumentRoot`, `Options` est paramétrée pour inclure `Indexes` et `FollowSymLinks`. `Indexes` permet au serveur de générer le contenu d'un répertoire si aucun `DirectoryIndex` (par exemple, `index.html`) n'est spécifié. `FollowSymLinks` permet au serveur de suivre des liens symboliques dans ce répertoire.

Remarque

Les déclarations `Options` de la section de configuration du serveur principal doivent être copiées individuellement dans chaque conteneur `VirtualHost`.

12.5.24 `AllowOverride`

La directive `AllowOverride` définit si des `Options` peuvent être annulées par les instructions présentes dans un fichier `.htaccess`. Par défaut, aussi bien le répertoire racine que le répertoire `DocumentRoot` sont paramétrés pour ne permettre aucune annulation via `.htaccess`.

12.5.25 `Order`

La directive `Order` contrôle simplement l'ordre dans lequel les directives `allow` et `deny` sont analysées. Le serveur est configuré pour analyser les directives `Allow` avant d'analyser les directives `Deny` s'appliquant au répertoire `DocumentRoot`.

12.5.26 `Allow`

`Allow` spécifie le client pouvant accéder à un répertoire donné. Le client peut être `all`, un nom de domaine, une adresse IP, une adresse IP partielle, une paire réseau/masque réseau, etc. Le répertoire `DocumentRoot` est configuré afin d'autoriser les requêtes (`Allow`) de quiconque (`all`), de la sorte, tout le monde peut y accéder.

12.5.27 `Deny`

`Deny` fonctionne selon le même principe que `Allow`, sauf que cette fois-ci, l'accès est refusé à un client donné. Le `DocumentRoot` n'est pas configuré par défaut pour refuser (`Deny`) des requêtes provenant d'un client quelconque.

12.5.28 `UserDir`

`UserDir` est le nom du sous-répertoire, au sein du répertoire personnel de chaque utilisateur, où devraient être placés les fichiers HTML personnels devant être servis par le serveur Web. Par défaut, la valeur attribuée à cette directive est `disable` (désactiver).

Dans le fichier de configuration par défaut, le nom du sous-répertoire est `public_html`. Par exemple, il se peut que le serveur reçoive la requête suivante :

```
http://example.com/~username/foo.html
```

Le serveur rechercherait alorsle fichier :

```
/home/username/public_html/foo.html
```

Dans l'exemple ci-dessus, `/home/username/` est le répertoire personnel de l'utilisateur (notez que le chemin d'accès par défaut vers les répertoires personnels des utilisateurs peut être différent).

Assurez-vous que les autorisations relatives aux répertoires personnels des utilisateurs sont correctement définies. Les répertoires personnels des utilisateurs doivent avoir des permissions équivalentes à 0711. Les bits de lecture (r) et d'exécution (x) doivent être définis sur les répertoires `public_html` des utilisateurs (0755 fonctionnera également). Les fichiers qui seront servis dans les répertoires `public_html` des utilisateurs doivent au moins avoir un valeur équivalente à 0644.

12.5.29 `DirectoryIndex`

`DirectoryIndex` est la page servie par défaut lorsqu'un utilisateur demande un index de répertoire en insérant une barre oblique (/) à la fin d'un nom de répertoire.

Lorsqu'un utilisateur demande à accéder à la page http://*exemple*/*ce_répertoire*/, il obtient soit la page `DirectoryIndex` si elle existe, soit une liste de répertoires générée par le serveur. La valeur par défaut de `DirectoryIndex` est le type de topologie `index.html` et `index.html.var`. Le serveur essaie de trouver l'un de ces fichiers et renvoie le premier qu'il trouve. S'il ne trouve aucun des deux fichiers et que `Options Indexes` est paramétrée pour ce répertoire, le serveur génère et renvoie une liste au format HTML, des sous-répertoires et fichiers contenus dans le répertoire (à moins que la fonctionnalité de listage des répertoires ne soit desactivée).

12.5.30 `AccessFileName`

`AccessFileName` nomme le fichier que le serveur doit utiliser pour les informations de contrôle d'accès dans chaque répertoire. La valeur par défaut est `.htaccess`.

Juste après la directive `AccessFileName`, une série de balises `Files` établit un contrôle d'accès sur tout fichier commençant par `.ht`. Ces directives refusent l'accès par le Web à tous les fichiers `.htaccess` (ou d'autres commençant par `.ht`) pour des raisons de sécurité.

12.5.31 `CacheNegotiatedDocs`

Par défaut, votre serveur Web demande aux serveurs proxy de ne pas mettre en cache des documents négociés sur la base du contenu (c'est-à-dire qui peuvent changer avec le temps ou suite à une entrée saisie par le demandeur). Si la valeur de `CacheNegotiatedDocs` est paramétrée sur `on`, cette fonctionnalité est désactivée et les serveurs proxy seront alors autorisés à mettre en cache de tels documents.

12.5.32 `TypesConfig`

`TypesConfig` nomme le fichier qui définit la liste par défaut des correspondances de type MIME (extensions de nom de fichier associées à des types de contenu). Le fichier

TypesConfig par défaut est /etc/mime.types. Au lieu d'éditer /etc/mime.types, il est plutôt recommandé d'ajouter des types MIME à l'aide de la directive AddType.

12.5.33 DefaultType

DefaultType définit un type de contenu par défaut pour le serveur Web devant être utilisé pour des documents dont les types MIME ne peuvent pas être déterminés. La valeur par défaut est text/plain.

12.5.34 HostnameLookups

HostnameLookups peut être paramétrée sur on, off ou double. Si HostnameLookups est paramétrée sur on, le serveur résout automatiquement l'adresse IP pour chaque connexion. La résolution de l'adresse IP suppose que le serveur établisse une ou plusieurs connexions avec un serveur DNS, rallongeant ainsi la durée des opérations de traitement. Si HostnameLookups est paramétrée sur double, le serveur établira une recherche DNS double inversée, rallongeant par là-même encore plus la durée des opérations de traitement.

Afin de conserver des ressources sur le serveur, la valeur par défaut donnée à HostnameLookups est off.

Si des noms d'hôtes sont nécessaires dans les fichiers journaux de serveurs, songez à exécuter l'un des nombreux outils conçus pour analyser les fichiers journaux ; ces derniers effectuent des recherches DNS non seulement de manière plus efficace mais également en masse lors de la rotation des fichiers journaux de serveurs Web.

12.5.35 ErrorLog

ErrorLog spécifie le fichier dans lequel sont journalisées les erreurs concernant les serveurs. La valeur par défaut pour cette directive est /var/log/httpd/error_log.

12.5.36 LogLevel

LogLevel définit le niveau de détail avec lequel les messages d'erreur devraient être enregistrés dans les journaux d'erreurs. Les valeurs possibles de LogLevel sont (du niveau le moins détaillé au niveau le plus détaillé) emerg, alert, crit, error, warn, notice, info ou debug. La valeur par défaut donnée à LogLevel est warn.

12.5.37 LogFormat

La directive LogFormat configure le format des fichiers journaux des différents serveurs Web. Le LogFormat utilisé dépend en fait des paramètres attribués dans la directive CustomLog.

Ci-dessous figurent les options de format s'appliquant si la valeur de la directive CustomLog est combined :

%h (adresse IP de l'hôte distant ou nom d'hôte)

Répertorie l'adresse IP distante du client demandeur. Si la valeur de `HostnameLookups` est `on`, le nom d'hôte du client est enregistré à moins que le DNS ne puisse le fournir.

`%l` (rfc931)

Option non-utilisée. Un tiret ([-]) apparaît à sa place dans le fichier journal.

`%u` (utilisateur authentifié)

Affiche l'identifiant de l'utilisateur enregistré, si l'authentification était nécessaire. Cette option n'étant généralement pas utilisée, dans le fichier journal, un tiret ([-]) figure dans le champ en question.

`%t` (date)

Enregistre la date et l'heure de la requête.

`%r` (chaîne de demandes)

Enregistre la chaîne de demandes telle qu'elle a été envoyée depuis le navigateur ou le client.

`%s` (état)

Enregistre le code d'état HTTP renvoyé à l'hôte client.

`%b` (octets)

Enregistre la taille du document.

`%\"%{Referer}i\"` (referrer)

Enregistre l'URL de la page Web qui a renvoyé l'hôte client au serveur Web.

`%\"%{User-Agent}i\"` (utilisateur-agent)

Enregistre le type de navigateur Web effectuant la requête.

12.5.38 `CustomLog`

`CustomLog` identifie le fichier journal et le format du fichier journal. Par défaut, l'enregistrement se fait dans le fichier `/var/log/httpd/access_log`.

Le format par défaut de `CustomLog` est le format du fichier journal `combined` illustré ci-dessous :

```
remotehost rfc931 user date "request" status bytes referrer user-agent
```

12.5.39 ServerSignature

La directive ServerSignature ajoute une ligne contenant la version du Serveur HTTP Apache et le nom du serveur (ServerName) pour tout document créé par un serveur, comme par exemple, les messages d'erreurs renvoyés aux clients. La valeur par défaut donnée à ServerSignature est on.

La valeur de cette directive peut également être off ou EMail. La valeur EMail ajoute une balise HTML mailto:ServerAdmin à la ligne de signature des réponses produites automatiquement par le système.

12.5.40. Alias

Le paramètre Alias permet d'accéder aux répertoires se trouvant en dehors du répertoire DocumentRoot. Toute URL se terminant par l'alias sera automatiquement convertie en chemin d'accès vers l'alias. Par défaut, un alias pour un répertoire icons est déjà configuré. Un répertoire icons est accessible par le serveur Web, mais le répertoire ne figure pas dans DocumentRoot.

12.5.41. ScriptAlias

La directive ScriptAlias définit l'endroit où se trouvent les scripts CGI. D'une manière générale, il est préférable de ne pas laisser de scripts CGI dans DocumentRoot, où ils peuvent être consultés comme des documents texte. C'est pour cette raison qu'il existe un répertoire spécial en dehors du répertoire DocumentRoot, contenant des exécutables et scripts côté-serveur, qui est désigné par la directive ScriptAlias. Ce répertoire, connu sous le nom cgi-bin, a /var/www/cgi-bin/ comme valeur par défaut.

Il est possible de créer des répertoires pour stocker des exécutables en dehors du répertoire cgi-bin.

12.5.42. Redirect

Lorsqu'une page Web est déplacée, Redirect peut être utilisée pour mapper l'ancienne URL vers une autre URL. Le format est le suivant :

```
Redirect      /<old-path>/<file-name>      http://<current-domain>/<current-path>/<file-name>
```

Dans cet exemple, remplacez d'une part *<old-path>* par les informations de l'ancien-chemin vers *<file-name>* et d'autre part *<current-domain>* et *<current-path>* par les informations relatives au domaine et au chemin actuels pour *<file-name>*.

Dans cet exemple, toute requête pour *<file-name>* à l'ancien emplacement est automatiquement redirigée vers le nouvel emplacement.

Pour obtenir des informations sur les techniques de redirection, utilisez le module mod_rewrite inclus dans le Serveur HTTP Apache. Pour de plus amples informations sur la configuration du module mod_rewrite, reportez-vous à la documentation de l'organisation

Apache Software Foundation disponible en ligne à l'adresse suivante : http://httpd.apache.org/docs-2.0/mod/mod_rewrite.html.

12.5.43. IndexOptions

`IndexOptions` contrôle l'apparence des listes de répertoires générées par le serveur, en ajoutant entre autres, des icônes et des descriptions de fichier. Si `Options Indexes` est définie, le serveur Web génère une liste des répertoires lorsqu'il reçoit une requête HTTP pour un répertoire sans index.

Le serveur Web recherche tout d'abord, dans le répertoire demandé un fichier correspondant aux noms spécifiés dans la directive `DirectoryIndex` (généralement, `index.html`). Si le serveur Web ne trouve aucun fichier `index.html`, le Serveur HTTP Apache génère une liste HTML des répertoires correspondant au répertoire demandé. L'apparence de cette liste de répertoires est contrôlée, en partie, par la directive `IndexOptions`.

La valeur de la configuration par défaut est `FancyIndexing`. Ainsi, un utilisateur peut réorganiser une liste de répertoires en cliquant sur les en-têtes des colonnes. En cliquant deux fois sur la même en-tête, le classement passera d'un ordre ascendant à un ordre descendant. La valeur `FancyIndexing` affiche également différentes icônes selon les types de fichiers, et ce, en fonction de leur extension.

Si l'option `AddDescription` est utilisée avec `FancyIndexing`, une brève description du fichier sera incluse dans les listes de répertoires générées par le serveur.

`IndexOptions` comprend un certain nombre de paramètres supplémentaires pouvant être utilisés pour contrôler l'apparence des répertoires créés par le serveur. Les paramètres `IconHeight` et `IconWidth` nécessitent que le serveur des balises HTML `HEIGHT` et `WIDTH` pour les icônes contenues dans les pages Web générées par le serveur. Le paramètre `IconsAreLinks` associe l'icône graphique à l'ancre du lien HTML, qui contient la cible du lien URL.

12.5.44. AddIconByEncoding

Cette directive nomme des icônes qui s'affichent par fichier avec codage MIME, dans des listes de répertoires générées par le serveur. Par exemple, le serveur Web est paramétré par défaut pour afficher l'icône `compressed.gif` à côté des fichiers codés MIME x-compress et x-gzip dans des listes de répertoires générées par le serveur.

12.5.45. AddIconByType

Cette directive nomme des icônes qui s'affichent à côté des fichiers avec des types MIME dans des listes de répertoires générées par serveur. Par exemple, le serveur est paramétré pour afficher l'icône `text.gif` à côté de fichiers avec un type MIME `text`, dans des listes de répertoires générées par le serveur.

12.5.46. AddIcon

`AddIcon` spécifie l'icône à afficher dans les listes de répertoires générées par le serveur pour des fichiers avec certaines extensions. Par exemple, le serveur Web est paramétré pour afficher l'icône `binary.gif` pour les fichiers portant les extensions `.bin` ou `.exe`.

12.5.47. DefaultIcon

`DefaultIcon` spécifie l'icône à afficher dans les listes de répertoires générées par le serveur pour les fichiers pour lesquels aucune autre icône n'est spécifiée. Le fichier image `unknown.gif` est la valeur par défaut.

12.5.48. AddDescription

Lors de l'utilisation de `FancyIndexing` comme paramètre de `IndexOptions`, la directive `AddDescription` peut être utilisée pour afficher des descriptions spécifiées par l'utilisateur pour certains fichiers ou pour certains types de fichiers dans des listes de répertoires générées par le serveur. La directive `AddDescription` prend en charge les fichiers de listes spécifiques, les expressions à caractères génériques ou les extensions de fichiers.

12.5.49. ReadmeName

`ReadmeName` nomme le fichier qui, s'il existe dans le répertoire, est ajouté à la fin des listes de répertoires générées par serveur. Le serveur Web commence par essayer d'inclure le fichier comme un document HTML, puis essaie de l'inclure comme un simple document texte. Par défaut, `ReadmeName` est paramétré sur `README.html`.

12.5.50. HeaderName

`HeaderName` nomme le fichier qui, s'il existe dans le répertoire, est ajouté au début des listes de répertoires générées par serveur. Comme `ReadmeName`, le serveur essaie, si possible, de l'inclure sous la forme d'un document HTML ou sinon, comme simple texte.

12.5.51. IndexIgnore

`IndexIgnore` affiche une liste d'extensions de fichiers, de noms de fichiers partiels, d'expressions contenant des caractères génériques ou de noms de fichiers complets. Le serveur Web n'inclura dans les listes de répertoires générées par le serveur, aucun fichier correspondant à l'un de ces paramètres.

12.5.52. AddEncoding

`AddEncoding` nomme des extensions de noms de fichiers qui devraient spécifier un type de codage particulier. Il est également possible d'utiliser `AddEncoding` pour donner l'instruction à certains navigateurs de décompresser certains fichiers lors de leur téléchargement.

12.5.53. AddLanguage

`AddLanguage` associe des extensions de noms de fichiers à des langues spécifiques. Cette directive est très utilisée pour le Serveur HTTP Apache (ou plusieurs) qui sert des contenus dans une multitude de langues et ce, en fonction de la préférence linguistique définie sur le navigateur client.

12.5.54. LanguagePriority

`LanguagePriority` permet de déterminer l'ordre de préférence des langues, au cas où aucune préférence linguistique ne serait paramétrée sur le navigateur client.

12.5.55. AddType

Utilisez la directive `AddType` pour définir ou annuler un type de MIME par défaut et des paires d'extensions de fichiers. L'exemple de directive suivant indique à Serveur HTTP Apache de reconnaître l'extension de fichier `.tgz` :

```
AddType application/x-tar .tgz
```

12.5.56. AddHandler

`AddHandler` mappe des extensions de fichiers sur des gestionnaires de signaux spécifiques. Par exemple, le module de commande `cgi-script` peut être utilisé en association avec l'extension `.cgi` pour traiter automatiquement un fichier dont le nom se termine par `.cgi` comme un script CGI. L'exemple suivant est un exemple de directive `AddHandler` pour l'extension `.cgi`.

```
AddHandler cgi-script .cgi
```

Cette directive active les scripts CGI en dehors du répertoire `cgi-bin` afin qu'ils puissent fonctionner dans tout répertoire se trouvant sur le serveur dont l'option `ExecCGI` figure au sein du conteneur de répertoires.

Outre son utilisation avec les scripts CGI, la directive `AddHandler` sert aussi au traitement de fichiers HTML et imagemap analysés par le serveur.

12.5.57. Action

`Action` spécifie l'association d'un type de contenu MIME à un script CGI de sorte que lorsqu' un fichier de ce type de support est demandé, un script CGI particulier est exécuté.

12.5.58. ErrorDocument

La directive `ErrorDocument` associe un code de réponse HTTP à un message ou à une URL qui sera renvoyé au client. Par défaut, le serveur Web renvoie un simple message d'erreur, habituellement obscur, lorsqu'une erreur se produit. La directive `ErrorDocument` force le serveur Web à renvoyer à la place une page ou un message personnalisés.

Important

Pour que le message soit valide, il *doit* se trouver entre guillemets ["].

12.5.59. `BrowserMatch`

La directive `BrowserMatch` permet au serveur de définir des variables d'environnement ou de prendre des mesures appropriées en fonction du champ d'en-tête Utilisateur-Agent HTTP (User-Agent HTTP) — qui identifie le type de navigateur du client. Par défaut, le serveur Web utilise `BrowserMatch` pour refuser des connexions à certains navigateurs présentant des problèmes connus de même que pour désactiver les les activités keepalive et vidages d'en-têtes HTTP pour les navigateurs ayant des problèmes avec ces actions.

12.5.60. `Location`

Les balises `<Location>` et `</Location>` permettent de créer un conteneur dans lequel un contrôle d'accès basé sur l'URL peut être spécifié.

Par exemple, pour permettre aux personnes se connectant depuis le domaine du serveur de consulter des rapports sur l'état du serveur, utilisez les directives suivantes :

```
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from <.example.com>
</Location>
```

Remplacez `<.example.com>` par le nom de domaine de second niveau du serveur Web.

Pour fournir des rapports de configuration des serveurs (y compris des modules installés et des directives de configuration) en réponse à des requêtes en provenance de votre domaine, utilisez les directives suivantes :

```
<Location /server-info>
    SetHandler server-info
    Order deny,allow
    Deny from all
    Allow from <.example.com>
</Location>
```

Ici encore, remplacez `<.example.com>` par le nom de domaine de second niveau du serveur Web.

12.5.61. `ProxyRequests`

Pour configurer le Serveur HTTP Apache de manière à ce qu'il fonctionne comme un serveur Proxy, supprimez le symbole dièse (`#`) placé au début de la ligne `<IfModule mod_proxy.c>`, de la directive ProxyRequests et de chaque ligne figurant dans la section `<Proxy>`. Paramétrez

la directive ProxyRequests sur On et définissez les domaines devant avoir accès au serveur dans la directive Allow from figurant dans la section <Proxy>.

12.5.62. Proxy

Les balises <Proxy *> et </Proxy> permettent de créer un conteneur qui renferme un groupe de directives de configuration devant s'appliquer seulement au serveur proxy. À l'intérieur des balises <Proxy>, il est possible d'utiliser de nombreuses directives s'appliquant à un répertoire.

12.5.63. Directives cache

Un certain nombre de directives cache commentées sont fournies dans le fichier de configuration par défaut du Serveur HTTP Apache. Dans la plupart des situations, il suffit de supprimer le commentaire en retirant le symbole dièse (#) placé au début de la ligne. Ci-après figure une liste de certaines des directives associées au cache ayant une grande importance :

- CacheEnable — Spécifie si le cache est un disque, une mémoire ou un cache de description de fichiers. Par défaut, CacheEnable configure un cache de disque pour les URL au niveau de ou au-dessous de /.
- CacheRoot — Définit le nom du répertoire qui contiendra les fichiers mis en cache. La valeur par défaut donnée à CacheRoot est le répertoire /var/httpd/proxy/.
- CacheSize — Définit la quantité d'espace en kilo-octets (Ko) que le cache peut utiliser. La valeur par défaut pour CacheSize est 5 Ko.

Ci-dessous figure une liste des autres directives courantes associées au cache.

- CacheMaxExpire — Définit la durée pendant laquelle les documents HTML mis en cache seront conservés (sans rechargement à partir du serveur Web duquel ils proviennent). La valeur par défaut est de 24 heures (86400 secondes).
- CacheLastModifiedFactor — Paramètre la création d'une date d'expiration pour un document qui a été reçu depuis le serveur d'origine sans date d'expiration définie. La valeur par défaut pour CacheLastModifiedFactor est réglée sur 0.1, ce qui signifie que la date d'expiration de tout document de ce type est égale à un dixième de la durée écoulée depuis la dernière modification du document.
- CacheDefaultExpire — Détermine la durée exprimée en heures, de l'expiration d'un document qui a été reçu à l'aide d'un protocole ne prenant pas en charge les délais d'expiration. La valeur par défaut est réglée sur 1 heure (3600 secondes).
- NoProxy — Établit une liste de sous-réseaux, d'adresses IP, de domaines ou d'hôtes séparés par des espaces dont le contenu n'est pas mis en cache. Ce paramètre est le plus utile sur les sites Intranet.

12.5.64. NameVirtualHost

La directive NameVirtualHost associe une adresse IP à un numéro de port, si nécessaire, pour tout hôte virtuel portant un nom. La configuration d'hôtes virtuels nommés permet à un Serveur HTTP Apache de servir différents domaines sans devoir pour ce faire utiliser de multiples adresses IP.

Remarque

L'utilisation de tout hôte virtuel nommé fonctionne *seulement* avec des connexions HTTP non-sécurisées. Si vous devez employer des hôtes virtuels avec un serveur sécurisé, utilisez plutôt des hôtes virtuels basés sur l'adresse IP.

Afin d'activer l'hébergement d'hôtes virtuels basés sur le nom, supprimez le symbole de commentaire figurant dans la directive de configuration `NameVirtualHost` et ajoutez la bonne adresse IP. Ajoutez ensuite des conteneurs `VirtualHost` supplémentaires pour chaque hôte virtuel, en fonction des besoins de votre configuration.

12.5.65. `VirtualHost`

Des balises `<VirtualHost>` et `</VirtualHost>` permettent de créer un conteneur soulignant les caractéristiques d'un hôte virtuel. Le conteneur `VirtualHost` accepte la plupart des directives de configuration.

Un conteneur `VirtualHost` commenté est fourni dans `httpd.conf` et illustre le groupe minimum de directives de configuration nécessaires pour chaque hôte virtuel.

Remarque

Le conteneur d'hôtes virtuels SSL par défaut se trouve désormais dans le fichier `/etc/httpd/conf.d/ssl.conf`.

12.5.66. Directives de configuration SSL

Les directives figurant dans le fichier `/etc/httpd/conf.d/ssl.conf` peuvent être configurées pour permettre des communications Web sécurisées à l'aide de SSL et TLS.

12.5.66.1. `SetEnvIf`

La directive `SetEnvIf` permet de régler des variables d'environnement en fonction des en-têtes des connexions entrantes. Il ne s'agit *pas* seulement d'une directive SSL, bien qu'elle soit présente dans le fichier `/etc/httpd/conf.d/ssl.conf` fourni. Dans le présent contexte, elle sert à désactiver la fonction keep-alive HTTP et à autoriser SSL à fermer la connexion sans générer de notification de fermeture de la part du navigateur client. Ce paramètre est nécessaire pour certains navigateurs qui n'interrompent pas la connexion SSL avec une grande fiabilité.

Pour obtenir de plus amples informations sur d'autres directives présentes dans le fichier de configuration SSL, consultez les documents disponibles aux adresses suivantes :

- http://localhost/manual/mod/mod_ssl.html
- http://httpd.apache.org/docs-2.0/mod/mod_ssl.html

Pour vous informer sur l'installation d'un serveur sécurisé HTTP Apache, reportez-vous au chapitre intitulé *Configuration du serveur sécurisé HTTP Apache* du *Guide d'administration système de Red Hat Enterprise Linux*.

Remarque

Dans la plupart des cas, les directives SSL sont configurées de manière appropriée lors de l'installation de Red Hat Enterprise Linux. Faites très attention lors de la modification des directives du serveur sécurisé HTTP Apache car une mauvaise configuration peut être à l'origine de brèches de sécurité, rendant tout système vulnérable.

12.6 Ajout de modules

Le Serveur HTTP Apache prend en charge des objets partagés dynamiquement (ou *DSO* de l'anglais *Dynamically Shared Objects*) ou des modules qui peuvent facilement être chargés selon les besoins.

A l'adresse suivante : http://httpd.apache.org/docs-2.0/dso.html, l'organisation Apache Project fournit une documentation complète en ligne sur les objets partagés dynamiquement (DSO). Sinon, si le paquetage `http-manual` est installé, de la documentation relative aux DSO est disponible en ligne à l'adresse suivante : http://localhost/manual/mod/.

Pour que le Serveur HTTP Apache puisse utiliser un DSO, il doit être spécifié dans une directive `LoadModule` du répertoire `/etc/httpd/conf/httpd.conf` ; si le module est fourni par un paquetage séparé, la ligne doit apparaître au sein du fichier de configuration des modules dans le répertoire `/etc/httpd/conf.d/`.

Lors de l'ajout ou de la suppression de modules du fichier `http.conf`, le Serveur HTTP Apache doit être rechargé ou relancé.

Lors de la création d'un nouveau module, installez tout d'abord le paquetage `httpd-devel` qui contient les fichiers à inclure (include files), les fichiers d'en-têtes ainsi que l'application *Apache eXtenSion* (`/usr/sbin/apxs`), qui utilise les fichiers à inclure et les fichiers d'en-têtes pour compiler les DSO.

Après l'écriture d'un module, utilisez `/usr/sbin/apxs` pour compiler les sources de votre module en dehors de l'arborescence source d'Apache. Pour obtenir de plus amples informations sur l'utilisation de la commande `/usr/sbin/apxs`, reportez-vous à la documentation Apache fournie en ligne à l'adresse suivante : http://httpd.apache.org/docs-2.0/dso.html et à la page de manuel de `apxs`.

Une fois le module compilé, placez-le dans le répertoire `/usr/lib/httpd/`. Ajoutez ensuite une ligne `LoadModule` dans le fichier `httpd.conf` en suivant la structure ci-dessous :

```
LoadModule <module-name> <path/to/module.so>
```

Où *<module-name>* correspond au nom du module et *<path/to/module.so>* au chemin d'accès vers le DSO.

12.7. Hôtes virtuels

L'hébergement intégré des hôtes virtuels du Serveur HTTP Apache permet au serveur de fournir différentes informations en fonction de l'adresse IP, du nom d'hôte ou du port faisant l'objet de la requête. Un guide complet sur l'utilisation des hôtes virtuels est disponible en ligne à l'adresse suivante : http://httpd.apache.org/docs-2.0/vhosts/.

12.7.1 Configuration d'hôtes virtuels

La meilleure façon de créer un hôte virtuel basé sur le nom consiste à utiliser le conteneur d'hôte virtuel fourni à titre d'exemple dans `httpd.conf`.

L'exemple de l'hôte virtuel offert se présente de la manière suivante :

```
#NameVirtualHost *:80
#
#<VirtualHost  *:80>
#    ServerAdmin webmaster@dummy-host.example.com
#    DocumentRoot /www/docs/dummy-host.example.com
#    ServerName dummy-host.example.com
#    ErrorLog logs/dummy-host.example.com-error_log
#    CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>
```

Pour activer la fonction d'hôte virtuel nommé, décommentez la ligne `NameVirtualHost` en retirant le symbole dièse (`#`) et en le remplaçant par le symbole de l'astérisque (`*`) accompagné de l'adresse IP attribuée à l'ordinateur.

Configurez ensuite un hôte virtuel, en décommentant et personnalisant le conteneur `<VirtualHost>`.

Sur la ligne `<VirtualHost>`, remplacez l'astérisque (`*`) par l'adresse IP du serveur. Remplacez aussi `ServerName` par le nom d'un DNS *valide* assigné à l'ordinateur et configurez les autres directives selon les besoins.

Étant donné que le conteneur `<VirtualHost>` accepte presque toutes les directives disponibles dans le cadre de la configuration du serveur principal, sa capacité à être personnalisé est très élevée.

Astuce

Si vous configurez un hôte virtuel pour qu'il écoute un port autre que le défaut, ce port doit être ajouté à la directive `Listen` dans la partie relative aux paramètres globaux du fichier `/etc/httpd/conf/http.conf`.

Afin de pouvoir activer l'hôte virtuel qui vient d'être créé, le Serveur HTTP Apache doit être rechargé ou redémarré.

Des informations complètes sur la création et la configuration d'hôtes virtuels sur la base du nom ou de l'adresse IP sont fournies en ligne à l'adresse suivante : http://httpd.apache.org/docs-2.0/vhosts/.

12.7.2 Hôte virtuel du serveur Web sécurisé

Par défaut, le Serveur HTTP Apache est configuré aussi bien comme un serveur Web non-sécurisé que comme un serveur sécurisé. Aussi bien le serveur non-sécurisé que le serveur sécurisé utilisent la même adresse IP et le même nom d'hôte, mais écoutent des ports différents, à savoir 80 et 443 respectivement. Ce faisant, des communications aussi bien non-sécurisées que sécurisées peuvent être établies simultanément.

Il est important de savoir que les transmissions HTTP améliorées grâce à SSL monopolisent cependant plus de ressources que le protocole HTTP standard et que par conséquent, un serveur sécurisé sert moins de pages par seconde. Dans de telles conditions, il est souvent recommandé de minimiser les informations disponibles à partir du serveur sécurisé, tout particulièrement sur un site Web très solicité.

Important

N'utilisez pas d'hôtes virtuels nommés de concert avec un serveur Web sécurisé car le protocole de transfert SSL intervient avant que la requête HTTP n'identifie l'hôte virtuel nommé approprié. Les hôtes virtuels nommés ne fonctionnent qu'avec un serveur Web non-sécurisé.

Les directives de configuration pour du serveur sécurisé se trouvent entre des balises d'hôte virtuel dans le fichier `/etc/httpd/conf.d/ssl.conf`.

Par défaut, aussi bien le serveur Web, sécurisé que le serveur non-sécurisé partagent le même `DocumentRoot`. Il est cependant recommandé qu'un `DocumentRoot` différent soit disponible pour le serveur Web sécurisé.

Afin que le serveur Web non-sécurisé n'accepte plus de connexions, annulez la ligne `Listen 80` du fichier `httpd.conf` en ajoutant un symbole dièse (`#`) au début de cette ligne. Une fois cette opération terminée, la ligne ressemblera à l'extrait ci-dessous :

```
#Listen 80
```

Pour plus d'informations sur la configuration d'un serveur Web utilisant SSL, reportez-vous au chapitre intitulé *Configuration du serveur HTTP Apache sécurisé* du *Guide d'administration système de Red Hat Enterprise Linux*. Pour obtenir des astuces de configuration avancées, consultez la documentation de l'organisation Apache Software Foundation qui est disponible en ligne aux adresses suivantes :

- http://httpd.apache.org/docs-2.0/ssl/
- http://httpd.apache.org/docs-2.0/vhosts/

12.8 Ressources supplémentaires

Pour en savoir plus sur le Serveur HTTP Apache, consultez les ressources mentionnées ci-dessous.

12.8.1 Sites Web utiles

- http://httpd.apache.org — Le site Web officiel du Serveur HTTP Apache contenant de la documentation non seulement sur toutes les directives mais également sur tous les modules par défaut.
- http://www.modssl.org — Le site Web officiel de `mod_ssl`.
- http://www.apacheweek.com — Une newsletter hebdomadaire complète en ligne concernant Apache.

12.8.2 Livres sur le sujet

- *Apache Desktop Reference* de Ralf S. Engelschall ; Addison Wesley — Ce livre écrit par Ralf Engelschall, un membre de l'organisation 'Apache Software Foundation (ASF) et auteur de `mod_ssl`, *Apache Desktop Reference* constitue un guide de référence concis et exhaustif pour l'utilisation du Serveur HTTP Apache et plus particulièrement pour sa compilation, sa configuration et son exécution. Ce livre est également disponible en ligne à l'adresse suivante : http://www.apacheref.com/.
- *Professional Apache* de Peter Wainwright ; Wrox Press Ltd — Ce manuel, *Professional Apache*, fait partie des nombreux livres de la collection "Programmer to Programmer" de la maison d'édition Wrox Press Ltd, destiné aux administrateurs de serveurs Web aussi bien expérimentés que débutants.
- *Administering Apache* de Mark Allan Arnold ; Osborne Media Group — Ce livre est destiné aux fournisseurs d'accès Internet désireux d'offrir des services plus sécurisés.
- *Apache Server Unleashed* de Richard Bowen, et al ; SAMS BOOKS — Une source encyclopédique pour le Serveur HTTP Apache.
- *Apache Pocket Reference* d'Andrew Ford, Gigi Estabrook ; O'Reilly — La dernière nouveauté de la collection O'Reilly Pocket Reference.
- *Guide d'administration système de Red Hat Enterprise Linux* ; Red Hat, Inc. — Ce manuel contient un chapitre sur la configuration du Serveur HTTP Apache à l'aide de l'**Outil de configuration HTTP** ainsi qu'un chapitre sur la configuration du serveur sécurisé Serveur HTTP Apache.
- *Guide de sécurité de Red Hat Enterprise Linux*; Red Hat, Inc. — Ce manuel contient un chapitre intitulé *Sécurité serveur* qui examine différentes manières de sécuriser le Serveur HTTP Apache et d'autres services.

TP 1 installation d'un serveur HTTP

Table of Contents

Résumé
Installation d'un serveur Web
    Introduction
    Configuration du serveur
    Activation du serveur
    Test de la configuration
    Auto-évaluation sur le premier TP

Résumé

Installation d'un serveur WEB - TP(s)

La séquence est bâtie pour des travaux réalisés avec plusieurs machines. Certaines parties pourront être réalisées sur votre propre machine, celle-ci servant de client et de serveur.

Vous devez avoir un navigateur d'installé, par exemple mozilla, firefox, konqueror, galeon...

La résolution de noms doit fonctionner.

Attention : Les paramètres peuvent différer d'une version à l'autre de Linux ou d'une distribution à l'autre. J'utilise dans ce document des variables, vous devrez y substituer les valeurs réelles de votre environnement.

- $APACHE_HOME, répertoire dans lequel sont stockées les pages du serveur.
- $APACHE_CONF, répertoire dans lequel sont stockés les fichiers de configuration.
- $APACHE_USER, compte utilisateur sous lequel fonctionne Apache.
- $APACHE_GROUP, groupe auquel est rattaché le compte $APACHE_USER.
- D'autres variables peuvent être définies au besoin dans l'ensemble des TP.

Installation d'un serveur Web

Introduction

Vous allez réaliser les opérations suivantes:

- configurer le serveur HTTP pour qu'il soit activé en mode standalone
- activer le serveur HTTP,
- tester le fonctionnement du serveur

A la fin vous devriez pouvoir accéder sur toutes les machines (serveurs HTTP) du réseau à partir du navigateur client.

Attention Vous utiliserez les éléments donnés dans la fiche de cours.

Configuration du serveur

Vous allez réaliser une configuration de base du serveur. Vous allez donc modifier les fichiers de configuration qui vous ont été présentés dans la fiche de cours. Avant toute modification, faites une copie de sauvegarde des fichiers.

Ouvrez le fichier à l'aide d'un éditeur, relevez et vérifiez les paramètres suivants. Pour chacun de ces paramètres vous noterez leurs rôles à partir de la fiche de cours et des commentaires (pensez à enregistrer vos modifications) :

- ServerName, nom pleinement qualifié de votre serveur (FQDN)
- Listen 80
- User $APACHE_USER
- Group $APACHE_GROUP
- ServerAdmin webmaster@localhost
- ServerRoot /etc/apache2
- DocumentRoot $APACHE_HOME
- UserDir public_html
- DirectoryIndex index.html index.shtml index.cgi
- AccessFileName .htaccess
- Alias /icons/ $APACHE_ICONS/icons/
- ScriptAlias /cgi-bin/ $APACHE_CGI/cgi-bin/

Activation du serveur

Regardez dans la fiche de cours les commandes de lancement du service serveur et la procédure de test. Regardez dans les fichiers de log et dans la table de processus si le service est bien démarré. Vérifier avec la commande netstat que le port 80 est bien ouvert.

Notez toutes les commandes que vous utilisez.

Test de la configuration

A ce stade le serveur est configuré et fonctionne. Il ne reste plus qu'à réaliser les tests. Vous devez pour cela activer le navigateur.

Faites les tests à partir de la machine locale et d'une machine distante. Utilisez les adresses localhost, 127.0.0.1, les adresses IP et les noms d'hôtes.

Si tout fonctionne, vous êtes en mesure de déployer votre site. Il suffira pour cela de l'installer dans l'arborescence $APACHE_HOME.

Dépannage: si cela ne fonctionne pas, procédez par élimination.

- 1 - Essayez avec les adresses IP des machines. Si ça fonctionne c'est que la résolution de noms n'est pas en place.
- 2 - Vérifiez que votre serveur est bien actif.
- 3 - Vérifiez que la configuration du serveur est correcte. Si vous apportez des modifications vous devez réinitialiser le serveur HTTP.

Auto-évaluation sur le premier TP

- Quels sont les fichiers de base pour la configuration du serveur apache2 et dans quels répertoires sont-ils situés ?
- Comment active-t-on un site ?
- Comment se nomme le compte d'utilisateur qui utilise le serveur http ?
- Quels sont les permissions d'accès par défaut sur le site principal du serveur ?
- Dans quel répertoire sont installés par défaut les pages HTML du site ?
- Dans quel répertoire par défaut sont stockés les scripts CGI et quel en est l'alias ?
- Quel est le principal rôle des alias ?
- Quelle(s) procédure(s) peut-on utiliser pour déterminer l'état du serveur et son bon fonctionnement ?
- Vous installez un serveur Apache sur une machine d'adresse 192.168.90.1 et de nom foo.foo.org. Lors des tests sur la machine locale, les commandes http://localhost, http://127.0.0.1, http://192.168.90.1 fonctionnent et http://foo.foo.org ne fonctionne pas. Lors des tests à partir d'une machine distante les commandes http://192.168.90.1 et http://foo.foo.org fonctionnent.

  Que peut-on en déduire et comment résoudre le problème?

TP 2 : Création de pages Web

Table of Contents

Résumé
Vérification de la configuration
Installation d'un site Web
Développement d'un site
Test de vos pages
Utilisation des alias
Auto évaluation sur le deuxième TP

Résumé

Vous allez réaliser les opérations suivantes

- Vérifier que la configuration de votre machine est correcte,
- Développer quelques pages HTML puis les déployer,
- Tester les nouvelles pages à partir d'un client Linux et Windows.

Vous utiliserez les archives qui vous sont fournies. Ces archives sont composées des quelques pages html qui vous serviront de site web et de scripts cgi.

Vérification de la configuration

Installez le service serveur et vérifiez qu'il est bien configuré et actif.

Pour tester la configuration de votre serveur, vous pouvez également utiliser la procédure suivante à partir de l'hôte local ou d'un hôte distant.

Lancez la commande suivante "$ telnet @IP du PC 80" (exemple : telnet 192.168.1.1 80 si cette adresse est celle de votre machine)

Cette commande crée une connexion au serveur httpd (port 80). Ce dernier invoque un agent.

Identifiez la connexion réseau dans une autre fenêtre xterm et avec la commande :

```
netstat -atup | grep ESTABLISHED
root@mr:/home# netstat -atup | grep ESTABLISHED
#Vous devriez obtenir quelque chose comme :
tcp        0      0 mr:33073   mr:www           ESTABLISHED 1513/telnet
```

Ensuite, transmettez à l'agent la ligne (commande) suivante : "GET /index.html"

Vérifiez que l'agent transmet de manière transparente le document HTML, et qu'il coupe automatiquement la connexion.

Installation d'un site Web

Vous allez utiliser les documents HTML fournis en annexe. Vous allez procéder de la façon suivante:

- Créez un répertoire $APACHE_HOME/journal pour y mettre toutes les pages html
- Copiez les images dans $APACHE_ICONS/icons (normalement /usr/share/apache2/icons)
- Copiez le script cgi compilé "prog" dans $APACHE_CGI/cgi-bin (normalement /usr/lib/cgi-bin)

Testez le site à partir d'un navigateur avec la commande http://@URLDuServeur/journal/

Le site est maintenant déployé, testez l'enchaînement des pages, l'affichage des images.

Le formulaire ne fonctionne pas encore. Vous avez copié le script compilé "prog" dans "/usr/lib/cgi-bin". Vérifiez quel est le nom du script que le formulaire "form.html" essaie de lancer sur le serveur.

Modifiez le nom du script ou le formulaire en conséquence.

Testez le fonctionnement du formulaire dans un navigateur.

Si vous rencontrez des difficultés sur l'exécution du script, vérifiez dans le fichier adéquat de configuration d'apache que vous avez bien :

```
    ScriptAlias /cgi-bin/ "/usr/lib/cgi-bin/"
et
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options None
        Order allow,deny
        Allow from all
    </Directory>
```

Développement d'un site

Réalisez sous LINUX votre curriculum vitae en langage HTML. Celui-ci devra être composé de plusieurs documents reliés par des liens (ancres). Il sera installé dans $APACHE_HOME/cv et les images dans le répertoire référencé par l'alias /icons/.

- - 1 page d'accueil de présentation avec les liens sur les autres pages,
- - 1 page pour la formation initiale,
- - 1 page pour les expériences professionnelles,
- - 1 page pour les loisirs, passions...

Chaque page doit vous permettre de revenir à la page d'accueil.

Mettez les pages dans le répertoire qui était prévu $APACHE_HOME/cv

Test de vos pages

Vous allez vous connecter à votre site à partir d'un client distant. Utilisez de préférence les adresses URL. Corrigez les erreurs si l'accès n'est pas réalisé.

Utilisation des alias

Afin de comprendre le fonctionnement des alias vous allez maintenant réaliser quelques manipulations. Vous allez déplacer le répertoire qui contient les images de "$APACHE_HOME/icons" vers "/tmp/httpd/icons.

- Réalisez l'opération de déplacement du répertoire vers /tmp/httpd/icons,
- Apportez les modifications nécessaires aux fichiers de configuration d'Apache
- Vérifiez le résultat.

Vous constatez ainsi qu'il est possible de déplacer un répertoire sur un serveur, sans qu'il soit nécessaire pour autant de modifier toutes les pages utilisant ce répertoire.

Auto évaluation sur le deuxième TP

- Quel est le nom de la page par défaut qui est ouverte par le navigateur dans un répertoire du serveur HTTP.
- Quel intérêt procure l'utilisation des alias ?
- Dans quels répertoires sont, par défaut, installés les pages HTML, scripts CGI, images et comment se nomment les alias ?
- On crée un répertoire $APACHE_HOME/html/journal pour y stocker des pages HTML. Il n'est pas possible d'y accéder alors que pour les autres sites tout fonctionne. Voici le message renvoyé par le navigateur. Aucune mesure de sécurité n'a été mise en oeuvre.

```
Forbidden
you don't have permission to access / on this server
```

Quelle est la cause du problème et comment y remédier ?

TP 3 : Configuration des répertoires personnels

Table of Contents

Configurer le compte personnel
Développer un site personnel
Tester l'accès au site personnel
Auto-évaluation sur le troisième TP

Vous allez mettre en place un accès pour les utilisateurs du système. Ceux-ci auront la possibilité de mettre leurs pages personnelles dans leur répertoire privé.

Vous allez réaliser les opérations suivantes:

- Configurer le compte personnel,
- Développer un site personnel,
- Tester l'accès au site personnel.

Relevez dans le fichier de configuration d'Apache2 adéquat le nom du répertoire dans lequel doivent être stockées les pages personnelles.

Configurer le compte personnel

- Créez un compte d'utilisateur. Je vais utiliser, pour la description des opérations le compte "mlx",
- Allez dans le répertoire personnel /home/mlx,
- Créez le répertoire du site Web personnel,
- Dans ce répertoire vous allez créer un répertoire pour les pages, un pour les images, un pour les scripts CGI avec la commande **mkdir**.

Développer un site personnel

Vous allez utiliser les pages HTML fournies en annexes. Utilisez les documents du TP précédent si vous en avez besoin.

Installez les fichiers fournis en annexe dans les répertoires adéquats.

Modifiez les pages HTML à l'aide d'un éditeur pour qu'elles utilisent les images de votre répertoire personnel "~/public_html/images" et le script CGI de votre répertoire personnel "~/public-html/cgi-bin" et pas ceux qui sont dans "$APACHE_HOME/cgi-bin".

Pour autoriser l'utilisation de scripts CGI dans un répertoire, vous devez le déclarer. Voici trois exemples :

```
<Directory /home/*/public_html/cgi-bin>
    Options ExecCGI
    SetHandler cgi-script
</Directory>
<Directory /home/*/public_html>
                Options +ExecCGI
```

```
</Directory>
<Directory /var/www/journal>
                Options +ExecCGI
</Directory>
```

Recherchez également la ligne suivante dans le fichier `apache2.conf` :

```
AddHandler cgi-script .cgi .sh .pl
```

Décommentez là ou ajoutez la si elle n'y est pas.

Copiez le script dans le répertoire que vous réservez pour les scripts. Vérifiez si c'est un programme "C" compilé qu'il porte bien l'extension ".cgi", au besoin renommez-le.

Tester l'accès au site personnel

Vous pouvez maintenant tester votre site personnel. A l'aide d'un navigateur utilisez l'URL "http//localhost/~mlx", (remarquez l'utilisation du "~" pour définir le répertoire personnel.)

Corrigez toutes les erreurs que vous pouvez rencontrer (problèmes d'alias, page principale, page de liens, problèmes de scripts, permissions d'accès au répertoire...)

Faites le test avec les sites personnels situés sur les autres machines.

Auto-évaluation sur le troisième TP

- Quel avantage présente l'utilisation des répertoires personnels pour le développement de sites Web ?
- Vous installez votre site personnel et vos pages. Vous tentez de réaliser un test or vous n'arrivez pas à accéder à vos pages. Quels peuvent être les problèmes et comment y remédier ?
- Vous rencontrez un problème de configuration. Vous apportez les corrections dans les fichiers de configuration, or la modification n'est toujours pas prise en compte sur le client. Que se passe-t-il et comment corriger le problème ?
- Comment avez vous fait pour que les scripts personnels soient chargés et exécutés de /home/mlx/public_html/cgi-bin
- Pour l'utilisateur mlx, sur la machine saturne et le domaine toutbet.edu, donnez:
    1. l'adresse URL de son site personnel,
    2. l'emplacement physique de son répertoire personnel sur la machine,
    3. le nom (et chemin complet) du fichier qui est activé quand on accède à son site.

TP 4 : Mise en place d'un accès sécurisé

Table of Contents

Déployer un site d'accès en ligne
Sécuriser l'accès à ce site par un mot de passe
Tester la configuration.
Les fichiers .htaccess
Auto-évaluation sur le quatrième TP

Vous allez réaliser les opérations suivantes:

1. Déployer un site d'accès en ligne
2. Sécuriser l'accès à ce site par un mot de passe
3. Tester la configuration.

Déployer un site d'accès en ligne

Vous allez utiliser les pages fournies en annexe.

Créez un répertoire sur votre machine. "mkdir $APACHE_HOME/protege"

Copiez les pages dans ce répertoire.

Sécuriser l'accès à ce site par un mot de passe

Dans un premier temps vous allez interdire l'accès à tout le monde. Pour cela vous allez modifier le fichier de configuration d'Apache2 adéquat et y mettre les lignes suivantes :

```
<Directory $APACHE_HOME/protege>
order deny,allow
deny from all
</Directory>
```

Arrêtez puis relancez le serveur et faites un test à partir d'un navigateur. Notez le message qui apparaît. Plus personne n'a accès au site.

Pour mettre un accès sécurisé par mot de passe il manque 2 éléments:

- Modifiez la configuration d'accès au répertoire,
- Créez le mot de passe crypté.

Modifiez la configuration d'accès au répertoire

```
<Directory $APACHE_HOME/protege>
AuthName Protected
AuthType basic
AuthUserFile $APACHE_CONF/users # fichier de mots de passe
<Limit GET POST>
require valid-user   # ici on demande une authentification
```

```
</Limit>
</Directory>
```

Créez le mot de passe crypté.

Le mot de passe est un fichier texte à deux champs séparés par un "deux points" (:). Le premier champ contient le compte d'utilisateur, le deuxième contient le mot de passe crypté.

Pour créer ce fichier, les comptes et les mots de passe, utilisez la commande "htpasswd".

Tester la configuration.

Ouvrez une session à l'aide d'un navigateur et ouvrez l'URL "http://localhost/protege"

Une fenêtre d'authentification doit s'ouvrir, entrez le nom d'utilisateur et le mot de passe.

Réalisez les opérations à partir de machines distantes.

Dépannage:

- Vérifiez le nom du répertoire que vous avez créé et la déclaration dans le fichierde configuration,
- Vérifiez le nom et la structure du fichier dans lequel vous avez mis les mots de passe.
- Si vous faites plusieurs tests, quittez puis relancez le navigateur après chaque session ouverte ou refusée,
- Vérifiez que le répertoire soit bien en mode 755 (chmod)
- Si cela ne fonctionne toujours pas reprenez le processus au début:

  - affectez toutes les permissions à tout le monde,

  - supprimez toutes les permissions à tout le monde,

  - affectez les restrictions.

- Utilisez un compte sans mot de passe. Le fichier $APACHE_CONF/users va contenir uniquement la chaîne: mlx, mais il n'y a pas le mot de passe.

  Si cela fonctionne alors le problème vient du cryptage du mot de passe.

Une fois que vous avez été authentifié, quittez et relancez le navigateur si vous voulez refaire le test d'accès à la ressource protégée.

Les fichiers .htaccess

En vous basant sur ce que vous venez de faire, sécurisez l'accès à un autre répertoire en utilisant un fichier .htaccess.

Auto-évaluation sur le quatrième TP

- Dans quel cas un accès sécurisé peut-il être utilisé ?
- Vous affectez un mot de passe à un utilisateur distant. Vous faites un test sur votre machine tout semble fonctionner. Il vous appelle pour vous dire que ses requêtes sur le site protégé sont toujours rejetées. Que se passe-t-il ?
- Si on utilise les possibilités du système, quelle autre solution peut-on utiliser pour interdire l'accès à un répertoire.

TP 5 : Serveurs webs virtuels et redirection

Table of Contents

Avant de commencer sur les serveurs web virtuels
Serveur web virtuel basé sur les adresses ip
Serveur Web virtuel basé sur le nom
Application sur la redirection
Annexe pour le "web-hosting"

Il est préférable d'avoir réalisé les TP sur les serveurs de noms avant de réaliser celui-ci.

Pour réaliser ce TP, vous devrez avoir un serveur de noms qui fonctionne.

La mise en place de serveurs webs virtuels, permet de faire cohabiter plusieurs serveurs sur un même hôte. Nous verrons qu'il y a plusieurs techniques pour faire cela.

On ajoutera, dans un premier temps le paramétrage des sites virtuels dans le fichier de configuration spécifique situé dans /etc/apache2/sites-available/default mais il est tout à fait possible de créer dans ce même répertoire un fichier pour chaque serveur virtuel.

- Les serveurs virtuels basés sur le nom, dans ce cas, vous devrez désigner pour une adresse IP sur la machine (et si possible le port), quel est le nom utilisé (directive ServerName), et quelle est la racine du site (directive DocumentRoot).

  Le nom de serveur (ServerName) et la racine web (DocumentRoot) doivent exactement correspondre respectivement au nom sous lequel le serveur virtuel sera nommé dans les URL clientes et au chemin du répertoire d'accueil des documents du site.

```
# Ici toutes les adresses ip sont utilisées, on peut mettre *

    NameVirtualHost *

# Toutes les requêtes sur http://www.domain.tld/
# pointeront sur /www/domain
    <VirtualHost *>
    ServerName www.domain.tld
    DocumentRoot /www/domain
    </VirtualHost>

# Toutes les requêtes sur http://www.otherdomain.tld/
# pointeront sur /www/otherdomain
    <VirtualHost *>
    ServerName www.otherdomain.tld
    DocumentRoot /www/otherdomain
    </VirtualHost>

# Ici on utilise une adresse en particulier
# Toutes les requêtes sur http://www.domain.tld/domain
# pointeront sur /web/domain

    NameVirtualHost 111.22.33.44
```

```
    <VirtualHost 111.22.33.44>
    ServerName www.domain.tld
    ServerPath /domain
    DocumentRoot /web/domain
    </VirtualHost>
```

- Dès lors que l'on a des serveurs virtuels qui "tournent" sur des ports différents, il est obligatoire de préciser pour chaque serveur virtuel son port :

```
NameVirtualHost * devient  NameVirtualHost *:80
<VirtualHost *> devient <VirtualHost *:80>
```

Nous le verrons avec le TP sur SSL.

- Les serveurs virtuels basés sur des adresses IP. Dans ce cas, chaque serveur aura sa propre adresse IP. Vous devrez également avoir un serveur de noms sur lequel toutes les zones et serveurs sont déclarés, car c'est ce dernier qui assurera la correspondance "adresse ip <-> nom du serveur" :

```
# Chaque serveur peut avoir son propre administrateur, ses
# propres fichiers de logs.
# Tous fonctionnent avec la même instance d'Apache.
# Il est possible de lancer plusieurs instances d'apache
# mais sur des ports différents

    <VirtualHost www.smallco.com>
    ServerAdmin webmaster@mail.smallco.com
    DocumentRoot /groups/smallco/www
    ServerName www.smallco.com
    ErrorLog /groups/smallco/logs/error_log
    TransferLog /groups/smallco/logs/access_log
    </VirtualHost>

    <VirtualHost www.baygroup.org>
    ServerAdmin webmaster@mail.baygroup.org
    DocumentRoot /groups/baygroup/www
    ServerName www.baygroup.org
    ErrorLog /groups/baygroup/logs/error_log
    TransferLog /groups/baygroup/logs/access_log
    </VirtualHost>
```

La redirection est un service un peu particulier, qui diffère de celui des services web virtuels. Il permet de rediriger une partie du site web à un autre endroit du disque physique. La encore on utilisera une des fonctions d'Apache qui permet de définir des alias.

Prenons par exemple un site installé physiquement sur "/var/www" et accessible par l'url "http://FQDN". L'url "http://FQDN/unREP" devrait correspondre normalement à l'emplacement physique "/var/www/unREP". La redirection va permettre de rediriger physiquement vers un autre répertoire.

Cette technique est largement utilisée par des applications ou pour la création d'espaces documentaires.

L'URL "http://FQDN/compta", pourra pointer physiquement dans "/usr/lib/compta" au lieu de "/var/www/compta".

Dans cet atelier, vous allez réaliser les opérations suivantes:

1. Installer un service de serveurs web virtuels par adresse et par nom,
2. Mettre en place un service de redirection par alias.

Avant de commencer sur les serveurs web virtuels

Configurez votre serveur de noms si ce n'est pas fait. Vous pouvez vous aider de l'annexe sur le "web-hosting" ci-dessous.

Pour créer des alias dynamiquement à votre interface réseau, utilisez la commande suivante:

```
ifconfig eth0:0 192.168.0.1
ifconfig eth0:1 192.168.1.1
ifconfig
eth0      Lien encap:Ethernet  HWaddr 00:D0:59:82:2B:86
          inet adr:192.168.90.2  Bcast:192.168.90.255  Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:95041 errors:0 dropped:0 overruns:0 frame:0
          TX packets:106227 errors:0 dropped:0 overruns:26 carrier:0
          collisions:0 lg file transmission:100
          RX bytes:57490902 (54.8 MiB)  TX bytes:11496187 (10.9 MiB)
          Interruption:11 Adresse de base:0x3000

eth0:0    Lien encap:Ethernet  HWaddr 00:D0:59:82:2B:86
          inet adr:192.168.0.1  Bcast:192.168.0.255  Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interruption:11 Adresse de base:0x3000

eth0:1    Lien encap:Ethernet  HWaddr 00:D0:59:82:2B:86
          inet adr:192.168.1.1  Bcast:192.168.1.255  Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interruption:11 Adresse de base:0x3000

lo        Lien encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:4564 (4.4 KiB)  TX bytes:4564 (4.4 KiB)
```

Créez votre serveur de noms pour deux domaines par exemple (domain1.org et domain2.org), adaptez la configuration du fichier "/etc/resolv.conf", vérifiez le bon fonctionnement de la résolution de noms sur les CNAME (www.domain1.org et www.domain2.org).

Serveur web virtuel basé sur les adresses ip

D'abord les déclarations dans le fichier de configuration :

```
# Web hosting ip based
```

```
NameVirtualHost 192.168.0.1
<VirtualHost 192.168.0.1>
DocumentRoot /home/web/domain1
ServerName www.domain1.org
</VirtualHost>

NameVirtualHost 192.168.1.1
<VirtualHost 192.168.1.1>
DocumentRoot /home/web/domain2
ServerName www.domain2.org
</VirtualHost>
```

Préparation des sites webs (sans trop se casser la tête) :

```
# mkdir -p /home/web/domain1 /home/web/domain2
# echo "<H1>Salut domain1 </H1>" > /home/web/domain1/index.html
# echo "<H1>Salut domain2 </H1>" > /home/web/domain2/index.html
```

On relance Apache :

```
/etc/init.d/apache2 restart
```

C'est terminé, les requêtes sur "http://ww.domain1.org", "http://www.domain2.org", doivent fonctionner et vous renvoyer la bonne page.

Serveur Web virtuel basé sur le nom

On va utiliser l'adresse ip de ns1.domain1.org pour les url w3.domain1.org et wiki.domain1.org. w3 et wiki sont deux zones de déploiement de site web différentes, sur un même serveur HTTP.

Modification du fichier de configuration :

```
# Web hosting name based (basé sur le nom)
#Site secondaire de domain1
 <VirtualHost 192.168.0.1>
ServerName w3.domain1.org
DocumentRoot /home/web/w3
</VirtualHost>

 <VirtualHost 192.168.0.1>
ServerName wiki.domain1.org
DocumentRoot /home/web/wiki
</VirtualHost>
```

Relisez la configuration d'Apache2.

Création des sites web :

```
# mkdir -p /home/web/wiki /home/web/w3
# echo "<H1>Salut w3 </H1>" > /home/web/w3/index.html
# echo "<H1>Salut wiki </H1>" > /home/web/wiki/index.html
```

Mettez les droits nécessaires.

C'est terminé, les requêtes sur :

```
http://w3.domain1.org
http://wiki.domain1.org
```

doivent retourner les bonnes pages.

Application sur la redirection

Nous allons créer un espace documentaire pour le domain "domain1.org". Il sera accessible par l'url "http://www.domain1.org/mesdocs", mais il sera localisé dans "/etc/domain1doc".

Créez un répertoire "domain1doc" dans /etc pour ce nouveau projet et mettez-y le fichier "apache.conf" ci-dessous:

```
# Création du répertoire
mkdir /etc/domain1doc

# Création du fichier /etc/domain1doc/apache.conf
Alias /mesdocs /etc/domain1doc/
<Directory /etc/domain1doc>
Options FollowSymLinks
AllowOverride None
order allow,deny
allow from all
</Directory>
```

Faites une inclusion de ce fichier dans le fichier de configuration d'Apache et relancez le service Apache.

C'est terminé, toutes les requêtes sur "http://www.domain1.org/mesdocs", ouvriront les pages qui sont dans "/etc/domain1doc".

Annexe pour le "web-hosting"

```
=== A rajouter dans /etc/named.conf

zone "domain1.org" {
        type master;
        file "/var/named/domain1.org.hosts";
        };

zone "domain2.org" {
        type master;
        file "/var/named/domain2.org.hosts";
        };



=== fichiers de zone
root@freeduc-sup:/var/named# more domain1.org.hosts
$ttl 38400
@                                       IN          SOA         domain1.org.
hostmaster.domain1.org. (
                        2004052604
                        10800
                        3600
                        604800
                        38400 )
@                         IN      NS      ns1.domain1.org.
ns1                       IN      A       192.168.0.1
www                       IN CNAME ns1
wiki                      IN CNAME ns1
w3                        IN CNAME ns1

root@freeduc-sup:/var/named# more domain2.org.hosts
$ttl 38400
@                                       IN          SOA         domain2.org.
hostmaster.domain2.org. (
                        2004052604
                        10800
                        3600
                        604800
                        38400 )
@                         IN      NS      ns1.domain2.org.
ns1                       IN      A       192.168.1.1
www                       IN CNAME ns1
```

Chapitre 13. Courrier électronique

L'apparition du courrier électronique (ou *email*) remonte au début des années 1960. La boîte aux lettres se présentait sous la forme d'un fichier dans le répertoire personnel d'un utilisateur que seul ce dernier pouvait lire. Les applications de messagerie primitives ajoutaient des nouveaux messages de texte au bas du fichier et l'utilisateur devait parcourir tout le fichier qui ne cessait de grandir, afin de retrouver tout message spécifique. Ce système ne pouvait envoyer de messages qu'aux utilisateurs d'un même système.

Le premier transfert réseau d'un courrier électronique a eu lieu en 1971 lorsqu'un ingénieur informatique nommé Ray Tomlinson a envoyé un message test entre deux ordinateurs via ARPANET — le précurseur de l'Internet. De là, la popularité de la communication par email s'est rapidement développée et en moins de deux ans, elle représentait 75 pour cent du trafic d'ARPANET.

Au fil du temps, les systèmes de messagerie électronique basés sur des protocoles réseau standardisés ont évolué de telle manière qu'ils font désormais partie des services les plus couramment utilisés sur l'Internet. Red Hat Enterprise Linux offre de nombreuses applications avancées permettant de servir et accéder aux emails.

Ce chapitre examine d'une part les protocoles de courrier électronique utilisés à l'heure actuelle et d'autre part, certains des programmes de messagerie électronique conçus pour envoyer et recevoir des emails.

13.1. Protocoles de courrier électronique

De nos jours, le courrier électronique est délivré à l'aide d'une architecture client/serveur. Un message électronique est créé au moyen d'un programme client de messagerie électronique. Ce programme envoie ensuite le message à un serveur. Ce dernier transmet à son tour le message au serveur de messagerie du destinataire où il est transmis au client de messagerie du destinataire final.

Afin de rendre ce processus possible, une vaste gamme de protocoles réseau standard permet à différents ordinateurs exécutant souvent différents systèmes d'exploitation et utilisant des programmes de messagerie électroniques différents, d'envoyer et de recevoir des emails.

Les protocoles suivants qui sont abordés dans ce chapitre sont ceux le plus fréquemment utilisés pour le transfert de courrier électronique entre systèmes.

13.1.1. Protocoles de transfert de courrier électronique

La livraison de courrier d'une application cliente au serveur et d'un serveur d'origine à un serveur de destination est traitée par le protocole nommé *Simple Mail Transfer Protocol* (ou *SMTP*).

13.1.1.1. SMTP

L'objectif primaire de SMTP consiste à transférer le courrier électronique entre les serveurs de messagerie. Toutefois, il a également une importance critique pour les clients de messagerie.

Afin d'envoyer un email, le client envoie le message électronique à un serveur de messagerie sortant, qui à son tour contacte le serveur de messagerie de destination pour la livraison du message. Dans de telles circonstances, il est nécessaire de spécifier un serveur SMTP lors de la configuraton d'un client de messagerie.

Sous Red Hat Enterprise Linux, un utilisateur peut configurer un serveur SMTP sur l'ordinateur local afin qu'il traite la livraison du courrier. Toutefois, il est également possible de configurer des serveurs SMTP distants pour le courrier sortant.

Il est important de noter ici que le protocole SMTP n'a pas besoin d'authentification pour fonctionner. Ainsi, quiconque utilisant l'Internet peut envoyer des emails à toute autre personne ou même à de grands groupes de personnes. C'est cette caractéristique de SMTP qui permet l'envoi de pourriel (aussi appelé junk email) ou de *spam*. Les serveurs SMTP modernes essaient néanmoins de minimiser ce comportement en n'autorisant que les hôtes connus à accéder au serveur SMTP. Les serveurs n'imposant pas ce genre de restriction sont appelés serveurs *open relay*.

Par défaut, Sendmail (`/usr/sbin/sendmail`) est le programme SMTP par défaut sous Red Hat Enterprise Linux. Néanmoins, une application serveur de messagerie plus simple appelée Postfix (`/usr/sbin/postfix`) est également disponible.

13.1.2. Protocoles d'accès au courrier

Pour récupérer le courrier électronique stocké sur les serveurs de messagerie, les applications client de messagerie utilisent deux protocoles primaires : *Post Office Protocol* (ou *POP*) et *Internet Message Access Protocol* (ou *IMAP*).

Contrairement à SMTP, ces deux protocoles exigent des clients qui se connectent de s'authentifier au moyen d'un nom d'utilisateur (aussi appelé identifiant) et d'un mot de passe. Par défaut, les mots de passe pour les deux protocoles sont transmis à travers le réseau de manière non-cryptée.

13.1.2.1. POP

Sous Red Hat Enterprise Linux, le serveur POP par défaut est `/usr/sbin/ipop3d` qui est inclus dans le paquetage `imap`. Lors de l'utilisation d'un serveur POP, les messages électroniques sont téléchargés par des applications client de messagerie. Par défaut, la plupart des clients de messagerie POP sont configurés automatiquement pour supprimer les messages sur le serveur une fois le transfert effectué ; toutefois, cette configuration peut souvent être modifiée.

Le protocole POP est compatible à 100 % avec des normes de messagerie Internet importantes, telles que *Multipurpose Internet Mail Extensions* (ou *MIME*), qui permet l'envoi de pièces jointes.

Le protocole POP est le plus approprié pour les utilisateurs disposant d'un système sur lequel ils peuvent lire leur courrier électronique. Il fonctionne également bien pour des utilisateurs n'ayant pas de connexion continue à l'Internet ou à un réseau sur lequel le serveur de messagerie se trouve. Malheureusement, pour les utilisateurs ayant des connexions réseau lentes, POP requiert que les programmes client, après authentification, téléchargent la totalité

du contenu de chaque message. Cette opération peut être longue si certains messages contiennent des pièces jointes.

La version la plus courante du protocole POP standard est POP3.

Il existe néanmoins de nombreuses variantes moins utilisées du protocole POP :

- *APOP* — POP3 avec authentification MDS. Un hachage codé du mot de passe de l'utilisateur est envoyé du client de messagerie au serveur plutôt qu'une version de ce dernier sous forme non-cryptée.
- *KPOP* — POP3 avec authentification Kerberos.
- *RPOP* — POP3 avec authentification RPOP. Cette variante utilise un identificateur (ID) publié pour chaque utilisateur, semblable à un mot de passe, pour authentifier les requêtes POP. Cependant, étant donné que cet ID n'est pas crypté, RPOP n'est pas plus sécurisé que le POP standard.

Pour une sécurité accrue, il est possible d'utiliser le cryptage *Secure Socket Layer* (*SSL*) pour l'authentification des clients et pour les sessions de transfert de données. Cette fonctionnalité peut être activée en utilisant le service `ipop3s` ou le programme `/usr/sbin/stunnel`.

13.1.2.2. IMAP

Sous Red Hat Enterprise Linux, `/usr/sbin/imapd` est le serveur IMAP par défaut, fourni par le paquetage `imap`. Lors de l'utilisation d'un serveur de messagerie IMAP, le courrier électronique est conservé sur le serveur où les utilisateurs peuvent lire et supprimer les emails. IMAP permet également aux applications client de créer, renommer ou supprimer des répertoires de messagerie sur le serveur afin d'organiser ou de stocker le courrier électronique.

Le protocole IMAP est utile tout particulièrement pour les utilisateurs accédant à leur courrier électronique au moyen d'ordinateurs multiples. Ce protocole est également pratique pour les utilisateurs se connectant au serveur de messagerie par le biais d'une connexion lente, car seule l'information d'en-tête du message est téléchargée jusqu'à ce qu'il soit ouvert, économisant ainsi de la largeur de bande. En outre, l'utilisateur peut également supprimer des messages sans devoir les lire ou les télécharger.

Par commodité, les applications IMAP client peuvent mettre en cache localement des copies des messages afin que l'utilisateur puisse naviguer parmi des messages déjà lus même lorsqu'il n'est pas directement connecté au serveur IMAP.

IMAP, tout comme POP, est compatible à 100 % avec des normes de messagerie Internet importantes, telles que MIME (Multipurpose Internet Mail Extensions) pour permettre l'envoi de pièces jointes.

Pour une sécurité accrue, il est possible d'utiliser le cryptage *SSL* pour l'authentification des clients et pour les sessions de transfert de données. Cette fonctionnalité peut être activée en utilisant le service `imaps` ou le programme `/usr/sbin/stunnel`

D'autres clients et serveurs IMAP libres et commerciaux sont disponibles ; un certain nombre d'entre eux poussent encore plus les possibilités du protocole IMAP et fournissent des

fonctionnalités supplémentaires. Une liste compréhensive de ces derniers est disponible en ligne à l'adresse suivante : http://www.imap.org/products/longlist.htm.

13.2. Classifications des programmes de messagerie électronique

D'une manière générale, toutes les applications de messagerie électronique font partie d'au moins un des trois types d'applications. Chaque type joue un rôle bien précis dans le processus de déplacement et de gestion des messages électroniques. Bien que la plupart des utilisateurs ne connaissent que le programme de courrier électronique qu'ils utilisent pour recevoir et envoyer des messages, chacun de ces trois types d'applications est important pour assurer que les messages arrivent à la bonne destination.

13.2.1. Agent de transfert de courrier (ATC)

L'*Agent de Transfert de Courrier* (*ATC*, ou MTA de l'anglais Mail Transfer Agent) sert à transférer des messages électroniques entre des hôtes utilisant SMTP. Un message peut requérir l'utilisation de plusieurs ATC lors de sa progression vers sa destination finale.

Alors que la livraison de messages entre ordinateurs puisse apparaître comme étant une opération assez simple et directe, l'ensemble du processus permettant de décider si un ATC (aussi appelé MTA selon l'acronyme anglais) donné peut ou devrait accepter la livraison d'un message, est en fait assez complexe. De plus, en raison des problèmes créés par les spams, l'utilisation d'un ATC spécifique est généralement limitée par la configuration même de l'ATC ou par celle de l'accès au réseau sur lequel il se trouve.

De nombreux programmes clients de messagerie peuvent également être utilisés comme des ATC pour envoyer des messages électroniques. Toutefois, il ne faut pas confondre cette opération avec le rôle primaire d'un ATC. La seule raison pour laquelle les programmes clients de messagerie peuvent envoyer des messages comme le fait un ATC réside dans le fait que l'hôte exécutant l'application ne dispose pas de son propre ATC. Cette situation s'applique tout particulièrement aux programmes clients de messagerie faisant partie de systèmes d'exploitation qui ne sont pas basés sur Unix. Cependant, ces programmes clients de messagerie n'envoient que des messages de sortie à un ATC qu'ils sont autorisés à utiliser et n'acheminent pas directement le message au serveur de messagerie du destinataire souhaité.

Étant donné que Red Hat Enterprise Linux installe deux ATC, à savoir Sendmail et Postfix, les programmes clients de messagerie ne sont généralement pas sollicités pour agir en tant qu'ATC. Red Hat Enterprise Linux inclut également Fetchmail, un ATC doté d'un objectif bien spécifique.

13.2.2. Agent de distribution du courrier (ADC)

Un *Agent de Distribution de Courrier* (*ADC* ou MDA de l'anglais Mail Delivery Agent) est utilisé par l'ATC pour distribuer le courrier arrivant dans la boîte aux lettres de l'utilisateur approprié. Dans de nombreuses situations, l'ADC est en fait un *Agent de Distribution Local* (*ADL*ou LDA de l'anglais Local Delivery Agent), comme `mail` ou Procmail.

En fait, tout programme traitant un message à des fins de distribution jusqu'au point où il peut être lu par une application client de messagerie peut être considéré comme un ADC. Telle est la raison pour laquelle certains ATC (comme Sendmail et Postfix) peuvent aussi jouer le rôle d'un ADC lorsqu'ils ajoutent de nouveaux messages électroniques au fichier spoule (aussi écrit spool) de courrier électronique d'un utilisateur local. En général, les ADC n'acheminent pas de messages entre les deux systèmes et ne fournissent pas d'interface utilisateur ; les ADC distribuent et classent les messages sur un ordinateur local pour qu'une application client de messagerie puissent y accéder.

13.2.3. Agent de gestion de courrier (AGC)

Un *Agent de Gestion de Courrier* (*AGC*, ou MUA de l'anglais Mail User Agent) est en fait une application client de messagerie. Un AGC est un programme qui, au minimum, permet à un utilisateur de lire et écrire des messages électroniques. De nombreux AGC peuvent récupérer des messages au moyen de protocoles POP ou IMAP, établissant des boîtes aux lettres pour stocker les messages et envoyant des messages de sortie à un ATC.

Les AGC (aussi appelés MUA selon l'acronyme anglais) peuvent être graphiques, comme **Mozilla Mail**, ou peuvent avoir une interface très simple à base de texte comme `mutt`.

13.3. Agent de transfert de courrier (ATC)

Red Hat Enterprise Linux comprend deux agents ATC principaux (ou MTA selon l'acronyme anglais), à savoir Sendmail et Postfix. Sendmail est configuré comme l'agent de transfert par défaut, bien que Postfix puisse facilement devenir l'ATC par défaut.

> **Astuce**
>
> Pour obtenir des informations sur la manière de changer l'ATC par défaut en remplaçant Sendmail par Postfix, reportez-vous au chapitre intitulé *Configuration de l'Agent de transfert de courrier (ATC)* du *Guide d'administration système de Red Hat Enterprise Linux*.

13.3.1. Sendmail

La tâche principale de Sendmail est de déplacer de façon sécurisée des messages électroniques entre des hôtes, utilisant généralement le protocole SMTP. Toutefois, Sendmail étant hautement configurable, il est possible de contrôler presque tous les aspects du traitement des messages, y compris le protocole à utiliser. De nombreux administrateurs système choisissent d'utiliser Sendmail comme ATC en raison de sa puissance et de sa modulabilité.

13.3.1.1. Objectif et limites

Il est important de bien comprendre ce qu'est Sendmail et ce qu'il peut faire, de même que ce qu'il n'est pas. À l'heure où des applications monolithiques jouent des rôles multiples, on pourrait penser que Sendmail est la seule application nécessaire pour exécuter un serveur de messagerie au sein d'une organisation. Techniquement parlant, c'est la cas puisque Sendmail peut non seulement spouler du courrier sur les répertoires de chaque utilisateur maisl il peut

également livrer des messages sortantspour les utilisateurs. Cependant, la plupart des utilisateurs demandent bien plus que le simple acheminement du courrier. Ils veulent en général interagir avec le courrier électronique à l'aide d'un AGC qui utilise POP ou IMAP pour télécharger leurs messages sur leur ordinateur local. Ou il se peut qu'ils préférent avoir une interface Web pour avoir accès à leur boîte aux lettres. Ces autres applications peuvent fonctionner de concert avec Sendmail, mais elles existent en réalité pour des raisons différentes et peuvent fonctionner indépendamment les unes des autres.

L'explication de tout ce que Sendmail devrait et pourrait faire en fonction de sa configuration va bien au-delà de la portée de cette section. Étant donné la gamme d'options différentes et le nombre de réglages possibles, des volumes entiers ont été écrits pour expliquer toutes les possibilités de Sendmail et les façons de résoudre d'éventuels problèmes.

Cette section passe en revue les fichiers installés par défaut avec Sendmail et examine certaines modifications de configuration élémentaires, y compris comment éviter de recevoir du pourriel (ou spam) et comment augmenter les capacités de Sendmail avec le protocole *Lightweight Directory Access Protocol (LDAP).*

13.3.1.2. Installation de Sendmail par défaut

Le fichier exécutable de Sendmail est `/usr/sbin/sendmail`.

Le fichier de configuration de Sendmail qui est long et détaillé se nomme `/etc/mail/sendmail.cf`. Évitez d'éditer le fichier `sendmail.cf` directement. Pour apporter des modifications à la configuration, éditez plutôt le fichier `/etc/mail/sendmail.mc`, sauvegardez le fichier original `/etc/mail/sendmail.cf` et utilisez ensuite le macroprocesseur `m4` qui est inclus pour créer un nouveau fichier `/etc/mail/sendmail.cf`.

Divers fichiers de configuration Sendmail sont installés dans `/etc/mail/`, notamment :

- `access` — Spécifie les systèmes qui peuvent utiliser Sendmail pour le courrier électronique sortant.
- `domaintable` — Spécifie le mappage de noms de domaine.
- `local-host-names` — Spécifie les alias de l'hôte.
- `mailertable` — Spécifie des instructions qui annulent le routage de domaines particuliers.
- `virtusertable` — Spécifie une forme de dénomination par alias spécifique au domaine, ce qui permet à des domaines virtuels multiples d'être hébergés sur un seul ordinateur.

Plusieurs fichiers de configuration placés dans `/etc/mail/`, tels que `access`, `domaintable`, `mailertable` et `virtusertable`, doivent en fait stocker leurs informations dans des fichiers de base de données avant que Sendmail ne puisse appliquer les modifications apportées à la configuration. Pour inclure les changements apportés à ces fichiers de configuration dans leurs fichiers de base de données, exécutez la commande suivante :

```
makemap hash /etc/mail/<name> < /etc/mail/<name>
```

où *<name>* doit être remplacé par le nom du fichier de configuration à convertir.

Par exemple, pour que tous les messages électroniques destinés au domaine `example.com` soient envoyés à <bob@other-example.com>, ajoutez la ligne reproduite ci-dessous au fichier `virtusertable` :

```
@example.com      bob@other-example.com
```

Pour finaliser cette modification, le fichier `virtusertable.db` doit être mis à jour à l'aide de la commande suivante, en étant connecté en tant que super-utilisateur :

```
makemap hash /etc/mail/virtusertable < /etc/mail/virtusertable
```

Ce faisant, un nouveau fichier `virtusertable.db` est créé, contenant la nouvelle configuration.

13.3.1.3. Modifications courantes de la configuration de Sendmail

Lors de la modification du fichier de configuration Sendmail, il est recommandé de générer un tout nouveau fichier `/etc/mail/sendmail.cf` plutôt que de modifier un fichier existant.

Avertissement

Avant de modifier le fichier `sendmail.cf`, il est toujours conseillé d'effectuer une copie de sauvegarde de la version courante du fichier.

Pour ajouter la fonctionnalité désirée à Sendmail, éditez le fichier `/etc/mail/sendmail.mc` en étant connecté en tant que super-utilisateur. Une fois cette opération terminée, utilisez le macroprocesseur `m4` pour générer un nouveau fichier `sendmail.cf` en exécutant la commande suivante :

```
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

Par défaut, le macroprocesseur `m4` est installé avec Sendmail mais fait partie du paquetage `m4`.

Après avoir créé un nouveau fichier `/etc/mail/sendmail.cf`, redémarrez Sendmail afin que les modifications soient appliquées. Pour ce faire, la meilleure façon de procéder consiste à taper la commande suivante :

```
/sbin/service sendmail restart
```

Important

Le fichier `sendmail.cf` par défaut n'autorise pas Sendmail à accepter des connexions réseau venant de tout hôte autre que l'ordinateur local. Afin de configurer Sendmail en tant que serveur pour d'autres clients, éditez `/etc/mail/sendmail.mc` et changez l'adresse spécifiée dans l'option `Addr=` de la directive `DAEMON_OPTIONS` de `127.0.0.1` à l'adresse IP d'un périphérique réseau actif ou supprimez tout simplement les commentaires de la directive `DAEMON_OPTIONS` en ajoutant `dnl` au début de la ligne. Une fois ces modifications apportées, régénérez le fichier `/etc/mail/sendmail.cf` en

exécutant la commande suivante :

```
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

La configuration par défaut incluse dans Red Hat Enterprise Linux pour être utilisée pour la plupart des sites exclusivement SMTP. Toutefois, elle ne fonctionne pas avec des sites utilisant UUCP (de l'anglais UNIX to UNIX Copy). Si l'utilitaire de connexion UUCP UUCP est utilisé pour les transferts, le fichier `/etc/mail/sendmail.mc` doit être reconfiguré et un nouveau fichier `/etc/mail/sendmail.cf` doit être généré.

Consultez le fichier `/usr/share/sendmail-cf/README` avant de modifier tout fichier contenu dans les répertoires présents sous le répertoire `/usr/share/sendmail-cf`, car ils peuvent affecter la configuration future de fichiers `/etc/mail/sendmail.cf`.

13.3.1.4. Masquage

L'une des configurations courantes de Sendmail consiste à avoir un seul ordinateur qui agit comme passerelle de messagerie pour tous les ordinateurs sur le réseau. Par exemple, une société pourrait souhaiter qu'un ordinateur appelé `mail.bigcorp.com` gère tout son courrier électronique et attribue à tous les messages sortants la même adresse de retour.

Dans ce cas de figure, le serveur Sendmail est obligé de masquer le nom des ordinateurs du réseau de la société de façon à ce que leur adresse de retour soit `user@example.com` plutôt que `user@host.example.com`.

Pour ce faire, ajoutez les lignes suivantes à `/etc/mail/sendmail.mc` :

```
FEATURE(always_add_domain)dnl
FEATURE(`masquerade_entire_domain')
FEATURE(`masquerade_envelope')
FEATURE(`allmasquerade')
MASQUERADE_AS(`bigcorp.com.')
MASQUERADE_DOMAIN(`bigcorp.com.')
MASQUERADE_AS(bigcorp.com)
```

Une fois qu'un nouveau fichier `sendmail.cf` aura été généré à l'aide de `m4`, cette configuration donnera l'impression que tous les messages envoyés à partir du réseau ont été été envoyés depuis `bigcorp.com`.

13.3.1.5. Blocage de pourriel

Les pourriels (aussi appelés spams) peuvent être définis comme étant des messages électroniques inutiles et indésirables reçus par un utilisateur qui n'a jamais demandé ce genre de communication. Il s'agit d'un abus très perturbateur, coûteux et répandu des normes de communication Internet.

Sendmail rend relativement aisé le blocage des nouvelles techniques utilisées pour envoyer des pourriels. Il bloque même par défaut, un grand nombre des méthodes d'envoi de spams les plus courantes.

Par exemple, le réacheminement de messages SMTP, également appelé retransmission (ou relaying), a été désactivé par défaut depuis la version 8.9 de Sendmail. Avant que ce changement n'ait lieu, Sendmail dirigeait l'hôte de messagerie (`x.edu`) de façon à ce qu'il accepte des messages d'un individu (`y.com`) et les envoie à un autre individu (`z.net`). Désormais, Sendmail doit être configuré de façon à autoriser un domaine à retransmettre du courrier par le biais du serveur. Pour configurer les domaines de retransmission, éditez le fichier `/etc/mail/relay-domains` et relancez Sendmail.

Ceci étant, les utilisateurs sont très souvent bombardés de pourriel provenant d'autres serveurs via l'Internet. Dans ce cas, les fonctions de contrôle d'accès de Sendmail, disponibles par l'entremise du fichier `/etc/mail/access` peuvent servir à empêcher les connexions en provenance d'hôtes indésirables. L'exemple suivant illustre comment utiliser ce fichier pour non seulement bloquer mais également autoriser l'accès au serveur Sendmail :

```
badspammer.com        ERROR:550 "Go away and do not spam us anymore"
tux.badspammer.com    OK
10.0                  RELAY
```

Cet exemple stipule que tout message électronique envoyé par `badspammer.com` doit être bloqué à l'aide d'un code d'erreur 550 conforme à la norme RFC-821 et qu'un message doit être renvoyé à l'expéditeur de pourriel. Le courrier envoyé par le sous-domaine `tux.badspammer.com` est en revanche accepté. La dernière ligne montre que tout message envoyé depuis le réseau 10.0.*.* peut être retransmis au moyen du serveur de messagerie.

Étant donné que `/etc/mail/access.db` est une base de données, utilisez `makemap` pour activer toute modification. Pour ce faire, tapez la commande suivante en étant connecté en tant que super-utilisateur :

```
makemap hash /etc/mail/access < /etc/mail/access
```

Cet exemple n'illustre qu'une toute petite partie du potentiel de Sendmail en termes d'autorisation ou d'interdiction d'accès. Reportez-vous au document `/usr/share/sendmail-cf/README` pour obtenir de plus amples renseignements et d'autres exemples sur le sujet.

Étant donné que Sendmail fait appel à l'ADC Procmail pour la livraison de courrier, il est également possible d'utiliser un programme de filtrage de pourriel comme SpamAssassin, pour identifier et classer ce type de courrier à la place de l'utilisateur.

13.3.1.6. Utilisation de Sendmail avec LDAP

L'utilisation de *Lightweight Directory Access Protocol (LDAP)* est une façon très rapide et puissante de trouver des informations spécifiques sur un utilisateur particulier appartenant à un grand groupe. Par exemple, un serveur LDAP peut servir à chercher une adresse électronique spécifique dans un répertoire d'entreprises courant à partir du nom de famille de l'utilisateur. Au niveau de ce genre d'implémentation, LDAP est très différent de Sendmail ; en effet, LDAP stocke les informations hiérarchiques des utilisateurs alors que Sendmail ne s'occupe que de recevoir le résultat de la recherche LDAP par le biais de messages électroniques pré-adressés.

Toutefois, Sendmail prend en charge une intégration beaucoup plus grande avec LDAP, là où il utilise LDAP pour remplacer des fichiers maintenus séparément, tels que `aliases` et `virtusertables`, sur divers serveurs de messagerie qui fonctionnent ensemble pour prendre en charge une organisation de taille moyenne ou supérieure. En bref, LDAP extraie le niveau de routage du courrier depuis Sendmail et ses fichiers de configuration séparés pour en faire un cluster LDAP puissant qui peut être influencé par de nombreuses applications différentes.

La version actuelle de Sendmail inclut la prise en charge pour LDAP. Pour étendre les possibilités de votre serveur Sendmail à l'aide de LDAP, prenez d'abord un serveur LDAP, tel que **OpenLDAP**, opérationnel et correctement configuré. Ensuite, modifiez votre fichier `/etc/mail/sendmail.mc` pour y inclure les éléments suivants :

```
LDAPROUTE_DOMAIN('yourdomain.com')dnl
FEATURE('ldap_routing')dnl
```

Remarque

Ces instructions ne s'appliquent qu'à une configuration très élémentaire de Sendmail avec LDAP. La configuration peut être très différente de celle-ci selon votre implémentation de LDAP, en particulier lors de la configuration de plusieurs ordinateurs Sendmail destinés à utiliser un serveur LDAP commun.

Consultez `/usr/share/doc/sendmail/README.cf` pour obtenir aussi bien des instructions détaillées sur la configuration de routage LDAP que des exemples.

Ensuite, recréez le fichier `/etc/mail/sendmail.cf` en exécutant `m4` et en redémarrant Sendmail.

13.3.2. Postfix

Mis au point à l'origine chez IBM par Wietse Venema, un programmeur et expert en sécurité, Postfix est un ATC compatible avec Sendmail qui est conçu pour être sécurisé, rapide et facile à configurer.

Afin d'accroître la sécurité, Postfix utilise une conception modulaire dans laquelle de petits processus dotés de privilèges limités sont lancés par un démon *maître* (ou master). Les petits processus dotés de privilèges limités effectuent des tâches très spécifiques en relation avec les différentes étapes de livraison du courrier et sont exécutés dans un environnement dit chrooté afin de restreindre l'impact des attaques.

Pour configurer Postfix de sorte qu'il accepte des connexions réseau venant d'hôtes autres que l'ordinateur local, il suffit d'apporter quelques modifications mineures dans son fichier de configuration. Pour ceux ayant des besoins plus complexes, Postfix fournit un certain nombre d'options de configuration ainsi que des possibilités d'ajout de tiers, qui font de ce dernier un ATC non seulement riche en fonctionnalités mais également très versatile.

Les fichiers de configuration de Postfix are lisibles par tout un chacun et prennent en charge plus de 250 directives. Contrairement à Sendmail, aucun macro-traitement n'est nécessaire pour que les modifications prennent effet et la majorité des options les plus courremment utilisées sont décrites dans le fichier contenant de nombreux commentaires.

Important

Avant d'utiliser Postfix, il est nécessaire de changer l'ATC par défaut de Sendmail à Postfix. Reportez-vous au chapitre intitulé *Configuration de l'agent de transfert de courrier (ATC)* du *Guide d'administration système de Red Hat Enterprise Linux* pour obtenir de plus amples informations.

13.3.2.1. Installation de Postfix par défaut

L'exécutable de Postfix est `/usr/sbin/postfix`. Ce démon lance tous les processus connexes nécessaires pour traiter la livraison de courrier.

Postfix stocke ses fichiers de configuration dans le répertoire `/etc/postfix/`. Ci-après figure une liste des fichiers les plus courremment utilisés :

- `access` — Utilisé pour le contrôle d'accès, ce fichier spécifie les hôtes qui sont autorisés à se connecter à Postfix.
- `aliases` — Fournit une liste configurable requise par le protocole de messagerie.
- `main.cf` — Représente le fichier de configuration global de Postfix. La majorité des options de configuration sont spécifiées dans ce fichier.
- `master.cf` — Spécifie la manière selon laquelle Postfix interagit avec différents processus pour effectuer la livraison de courrier.
- `>transport` — Mappe les adresses de courrier électronique pour qu'elles prennent le relais des hôtes.

Important

Le fichier `/etc/postfix/main.cf` par défaut ne permet pas à Postfix d'accepter des connexions réseau venant d'un hôte autre que l'ordinateur local.

Lors de la modification de certaines options dans le répertoire `/etc/postfix/`, il sera peut-être nécessaire de redémarrer le service `postfix` afin que les modifications apportées prennent effet. Pour ce faire, la meilleure façon consiste à taper la commande suivante :

```
/sbin/service postfix restart
```

13.3.2.2. Configuration élémentaire de Postfix

Par défaut, Postfix n'accepte pas de connexions réseau venant d'un hôte autre que l'ordinateur local. Efeectuez les étapes suivantes en étant connecté en tant que super-utilisateur afin d'activer la livraison de courrier pour d'autres hôtes du réseau :

- Éditez le fichier `/etc/postfix/main.cf` à l'aide d'un éditeur de texte tel que `vi`.
- Décommentez la ligne `mydomain` en supprimant la symbole dièse (#) et remplacez *`domain.tld`* par le nom de domaine que le serveur de messagerie sert, tel que `example.com`.
- Décommentez la ligne `myorigin = $mydomain`.
- Décommentez la ligne `myhostname` et remplacez *`host.domain.tld`* par le nom d'hôte de l'ordinateur.

- Décommentez la ligne `mydestination = $myhostname, localhost.$mydomain`.
- Décommentez la ligne `mynetworks` et remplacez *`168.100.189.0/28`* par un paramètre de réseau valide pour les hôtes pouvant se connecter au serveur.
- Décommentez la ligne `inet_interfaces = all`.
- Redémarrez le service `postfix`.

Une fois ces étapes effectuées, l'hôte est en mesure d'accepter la livraison d'emails venant de l'extérieur

Postfix dispose d'une grande gamme d'options de configuration. Une des meilleures façons d'apprendre comment configurer Postfix consiste à lire les commentaires dans `/etc/postfix/main.cf`. Des ressources supplémentaires incluant des informations sur LDAP et l'intégration de SpamAssassin sont disponibles en ligne à l'adresse suivante : http://www.postfix.org/.

13.3.3. Fetchmail

Fetchmail est un ATC qui récupère du courrier électronique depuis des serveurs distants et le transfère à l'ATC local. De nombreux utilisateurs apprécient la possibilité de pouvoir séparer le processus de téléchargement de leurs messages stockés sur un serveur distant, du processus de lecture et d'organisation de leur courrier dans un AGC. Conçu tout spécialement pour les utilisateurs à accès par ligne commutée, Fetchmail se connecte et télécharge rapidement tous les messages électroniques dans le fichier spoule de messagerie à l'aide de nombreux protocoles différents parmi lesquels figurent POP3 et IMAP. Il permet même de réacheminer vos messages vers un serveur SMTP, si nécessaire.

Fetchmail est configuré pour chaque utilisateur grâce à un fichier `.fetchmailrc` du répertoire personnel de l'utilisateur.

Sur la base des préférences spécifiées dans le fichier `.fetchmailrc`, Fetchmail recherche les messages électroniques sur un serveur distant et les télécharge. Il les achemine ensuite sur le port 25 de l'ordinateur local, au moyen de l'ATC local, pour les placer dans le fichier spoule de l'utilisateur approprié. Si Procmail est disponible, il peut être utilisé pour filtrer les messages et les placer dans une boîte à lettres de sorte qu'ils puissent être lus avec par un AGC.

13.3.3.1. Options de configuration de Fetchmail

Bien qu'il soit possible de passer toutes les options nécessaires pour vérifier le courrier sur un serveur distant depuis la ligne de commande lors de l'exécution de Fetchmail, il est beaucoup plus simple d'utiliser un fichier `.fetchmailrc`. Insérez toutes les options de configuration souhaitées dans le fichier `.fetchmailrc` et ces options seront alors utilisées lors de chaque exécution de la commande `fetchmail`. Il est possible d'annuler toute option lors du lancement de Fetchmail en spécifiant l'option en question sur la ligne de commande.

Le fichier `.fetchmailrc` d'un utilisateur contient trois types d'options de configuration :

- *options globales* — Ce type d'options donne à Fetchmail des instructions qui contrôlent le fonctionnement du programme ou fournit des réglages pour toute connexion de vérification du courrier.
- *options serveur* — Ce type d'options spécifie les informations nécessaires concernant le serveur sondé, telles que le nom d'hôte ainsi que les préférences relatives aux serveurs de messagerie spécifiques, comme le port à vérifier ou le nombre de secondes devant s'écouler avant l'interruption de la connexion. Ces options affectent tout utilisateur employant ce serveur.
- *options utilisateur* — Ce type d'options contient des informations, telles que le nom d'utilisateur et le mot de passe, nécessaires pour l'authentification et la vérification du courrier à l'aide d'un serveur de messagerie donné.

Les options globales apparaissent en haut du fichier de configuration `.fetchmailrc`, suivies d'une ou plusieurs options serveur, précisant chacune un serveur de messagerie différent sur lequel Fetchmail devrait vérifier le courrier. Les options utilisateur suivent les options serveur pour chaque compte utilisateur devant être vérifié sur ce serveur de messagerie. Tout comme pour les options serveur, il est possible de spécifier non seulement de multiples options utilisateur à employer avec un serveur donné mais il est également possible de vérifier plusieurs comptes de messagerie sur un même serveur.

Les options serveur à utiliser sont appelées dans le fichier `.fetchmailrc` grâce à l'emploi d'un verbe d'option spécial, `poll` ou `skip`, qui précède toute information concernant le serveur. L'action `poll` indique à Fetchmail d'utiliser cette option serveur lorsqu'il est exécuté ; il vérifie en fait le courrier à l'aide des différentes options utilisateur. Toute option serveur placée après une action `skip` contourne les opérations de vérification, à moins que le nom d'hôte de ce serveur ne soit spécifié lorsque Fetchmail est invoqué. L'option `skip` est utile lors du test de configurations dans `.fetchmailrc` car elle vérifie les serveurs avec cette option uniquement lorsqu'ils sont invoqués spécifiquement et n'affecte pas les configurations actuelles.

Un exemple de fichier `.fetchmailrc` ressemble à l'extrait suivant :

```
set postmaster "user1"
set bouncemail

poll pop.domain.com proto pop3
     user 'user1' there with password 'secret' is user1 here

poll mail.domain2.com
     user 'user5' there with password 'secret2' is user1 here
     user 'user7' there with password 'secret3' is user1 here
```

Dans cet exemple, les options globales sont configurées de façon à ce que l'utilisateur reçoive le courrier seulement en dernier ressort (option `postmaster`) et que toutes les erreurs soient envoyées au maître de poste (ou postmaster) plutôt qu'à l'expéditeur (option `bouncemail`). L'action `set` indique à Fetchmail que cette ligne contient une option globale. Ensuite, deux serveurs de messagerie sont spécifiés ; le premier est configuré pour vérifier POP3 et le second pour essayer divers protocoles afin d'en trouver un qui fonctionne. Deux utilisateurs sont vérifiés dans le cas de la seconde option de serveur, mais tout message électronique trouvé pour l'un ou l'autre des utilisateurs est envoyé dans le fichier spoule de messagerie de l'utilisateur No. 1 (ou `user1`). Ainsi, il est possible de vérifier de multiples boîtes aux lettres

sur plusieurs serveurs, tout en utilisant une une seule corbeille d'arrivée pour l'AGC. Toute information spécifique à chacun des utilisateurs commence par l'action `user`.

Remarque

Les utilisateurs ne doivent pas placer leur mot de passe dans le fichier `.fetchmailrc`. Si la section `with password '<password>'` est omise, Fetchmail demandera la saisie d'un mot de passe lors de son lancement.

Fetchmail offre de nombreuses options aussi bien globales que locales ou serveur. Beaucoup d'entre elles sont rarement utilisées ou ne s'appliquent qu'à des situations très particulières. La page de manuel de `fetchmail` explique chacune de ces options de façon détaillée, mais les options les plus courantes sont énumérées ci-dessous.

13.3.3.2. Options globales

Chaque option globale devrait être placée sur une seule ligne et précédée de l'action `set`.

- `daemon <seconds>` — Spécifie le mode démon où Fetchmail demeure en tâche de fond. Remplacez `<seconds>`par la durée en secondes pendant laquelle Fetchmail doit attendre avant de sonder le serveur.
- `postmaster` — Spécifie un utilisateur local auquel envoyer le courrier en cas de problèmes de distribution.
- `syslog` — Spécifie le fichier journal pour l'enregistrement des messages d'erreurs et d'état. La valeur `/var/log/maillog` est retenue par défaut.

13.3.3.3. Options serveur

Les options serveur doivent figurer sur leur propre ligne dans `.fetchmailrc`, après une action `poll` ou `skip`.

- `auth <auth-type>` — Remplacez `<auth-type>` par le type d'authentification à utiliser. Par défaut, l'authentification `password` est utilisée, mais certains protocoles prennent en charge d'autres types d'authentification, notamment `kerberos_v5`, `kerberos_v4` et `ssh`. Si le type d'authentification `any` est retenu, Fetchmail essaiera d'abord des méthodes qui ne nécessitent aucun mot de passe, puis des méthodes qui masquent le mot de passe et, en dernier ressort, il essaiera d'envoyer le mot de passe en texte en clair pour effectuer l'authentification auprès du serveur.
- `interval <number>` — Indique à Fetchmail de sonder seulement le serveur spécifié après avoir vérifié le courrier sur tous les serveurs configurés un certain nombre de fois (où `<number>` représente ce nombre de fois). Cette option est généralement utilisée pour les serveurs de messagerie sur lesquels un utilisateur ne reçoit que rarement des messages.
- `port <port-number>` — Remplacez `<port-number>` par le numéro du port. Cette valeur annule le numéro du port par défaut pour un protocole spécifié.
- `proto <protocol>` — Remplacez `<protocol>` par le protocole, tel que `pop3` ou `imap`, devant être utilisé pour vérifier le courrier sur le serveur.
- `timeout <seconds>` — Remplacez `<seconds>` par la durée d'inactivité du serveur (exprimée en secondes) après laquelle Fetchmail abandonne une tentative de

connexion. Si cette valeur n'est pas configurée, le système retient une valeur par défaut de `300` secondes.

13.3.3.4. Options utilisateur

Les options utilisateur peuvent être placées sur leurs propres lignes sous une option serveur ou alors sur la même ligne que l'option de serveur. Dans les deux cas, les options définies doivent suivre l'option `user` (définie ci-dessous).

- `fetchall` — Donne l'ordre à Fetchmail de télécharger tous les messages de la file d'attente, y compris les messages qui ont déjà été visualisés. Par défaut, Fetchmail ne récupère que les nouveaux messages.
- `fetchlimit` *<number>* — Remplacez *<number>* par le nombre de messages à extraire avant de s'arrêter.
- `flush` — Donne l'instruction à Fetchmail de supprimer tous les messages de la file d'attente qui ont été lus précédemment avant d'extraire les nouveaux messages.
- `limit` *<max-number-bytes>* — Remplacez *<max-number-bytes>* par la taille maximales en octets autorisée pour des messages lors de leur extraction. Cette option est pratique lors de connexions réseau lentes, particulièrement lorsqu'un gros message prend trop de temps à être télécharger.
- `password '`*<password>*`'` — Remplacez *<password>* par le mot de passe de l'utilisateur.
- `preconnect "`*<command>*`"` — Remplacez *<command>* par une commande à exécuter avant de récupérer les messages pour cet utilisateur.
- `postconnect "`*<command>*`"` — Remplacez *<command>* par une commande à exécuter après avoir récupéré les messages pour cet utilisateur.
- `ssl` — Active le cryptage SSL.
- `user "`*<username>*`"` — Remplacez *<username>* par le nom d'utilisateur employé par Fetchmail pour récupérer les messages électroniques. *Cette option doit être placée avant toute autre option utilisateur.*

13.3.3.5. Options de commande pour Fetchmail

La plupart des options utilisées en ligne de commande lors de l'exécution de la commande `fetchmail`, répliquent les options de configuration de `.fetchmailrc`. Ainsi, Fetchmail peut être utilisé avec ou sans fichier de configuration. Ces options ne sont pas utilisées en ligne de commande par la plupart des utilisateurs car il est plus simple de les laisser dans le fichier `.fetchmailrc`.

Toutefois, il se peut que dans certaines situations la commande `fetchmail` doive être exécutée avec d'autres options dans un but bien précis. Il est possible d'exécuter des options de commande afin d'annuler temporairement un paramètre de `.fetchmailrc` qui crée une erreur étant donné que toute option spécifiée à la ligne de commande annule les options contenues dans le fichiers de configuration.

13.3.3.6. Options d'information ou de débogage

Certaines options utilisées après la commande `fetchmail` permettent d'obtenir des informations importantes.

- `--configdump` — Affiche toutes les options possibles sur la base des informations fournies pas `.fetchmailrc` et les valeurs par défaut de Fetchmail. Lors de l'utilisation de cette option, aucun message électronique n'est téléchargé pour quelque utulisateur que ce soit.
- `-s` — Exécute Fetchmail en mode silencieux, empêchant tout message, autre que des messages d'erreurs, d'apparaître après la commande `fetchmail`.
- `-v` — Exécute Fetchmail en mode prolixe, affichant toute communication entre Fetchmail et les serveurs de messagerie distants.
- `-V` — Affiche des informations détaillées sur la version utilisée, dresse la liste de ses options globales et fournit les paramètres à employer pour chaque utilisateur, y compris le protocole de messagerie et la méthode d'authentification. Lors de l'utilisation de cette option, aucun courrier électronique n'est récupéré pour quelque utilisateur que ce soit.

13.3.3.7. Options spéciales

Ces options peuvent parfois être pratiques pour annuler les valeurs par défaut qui se trouvent souvent dans le fichier `.fetchmailrc`.

- `-a`— Indique à Fetchmail de télécharger tous les messages depuis le serveur de messagerie distant, qu'ils soient nouveaux ou qu'ils aient déjà été consultés. Par défaut, Fetchmail ne télécharge que les nouveaux messages.
- `-k` — Fetchmail laisse les messages sur le serveur de messagerie distant après les avoir téléchargés. Cette option annule le comportement par défaut consistant à supprimer les messages après les avoir téléchargés.
- `-l <max-number-bytes>` — Fetchmail ne télécharge pas les messages dont la taille est supérieure à la taille spécifiée et les laisse sur le serveur de messagerie distant.
- `--quit` — Quitte le processus du démon Fetchmail.

D'autres commandes et options `.fetchmailrc` sont offertes dans la page de manuel de `fetchmail`.

13.4. Agent de distribution de courrier (ADC)

Red Hat Enterprise Linux inclut deux ADC principaux, à savoir Procmail et `mail`. Ces deux applications sont considérées comme des agents de distribution locaux (ou ADL) et elles acheminent toutes les deux le courrier électronique du fichier spoule d'un ADC vers la boîte aux lettres de l'utilisateur. Toutefois, Procmail fournit un système de filtrage robuste.

Cette section examine seulement Procmail de façon détaillée. Pour toute information sur la commande `mail`, consultez la page de manuel qui lui est dédié.

Procmail distribue et filtre le courrier électronique dès qu'il est placé dans le fichier spoule de messagerie de l'hôte local. Il est puissant, peu exigeant en matière de ressources système et d'une utilisation courante. Procmail peut jouer un rôle critique dans la distribution du courrier qui sera lu par les applications client de messagerie.

Il existe différentes façons d'invoquer Procmail. Dès qu'un ATC dépose un message dans le fichier spoule de messagerie, Procmail est lancé. Ce dernier filtre et classe alors le message de

manière à ce que l'AGC puisse le trouver puis quitte le processus. L'AGC peut également être configuré de sorte qu'il exécute Procmail chaque fois qu'un message est reçu afin que le courrier soit acheminé vers les boîtes aux lettres appropriées. Par défaut, la présence d'un fichier `/etc/procmailrc` ou d'un fichier `.procmailrc` (aussi appelé un fichier *rc*) dans le répertoire personnel d'un utilisateur invoquera Procmail dès qu'un ATC reçoit un nouveau message.

Toute action effectuée par Procmail sur un message électronique dépend de la capacité du message à satisfaire un ensemble de conditions ou *recettes* (aussi appelées recipes selon le terme anglais) particulières contenues dans le fichier `rc`. Si un message satisfait une recette, il peut alors être placé dans un fichier donné, être supprimé ou être traité d'une autre façon.

Lors du démarrage de Procmail, ce dernier lit les messages électroniques et sépare les informations relatives au corps du message de celles concernant l'en-tête. Ensuite, Procmail cherche les fichiers `/etc/procmailrc` et `rc` dans le répertoire `/etc/procmailrcs` pour trouver les variables d'environnement et recettes de Procmail s'appliquant par défaut à l'ensemble du système. Procmail cherche ensuite un fichier `.procmailrc` dans le répertoire personnel de l'utilisateur. De nombreux utilisateurs créent des fichiers `rc` supplémentaires pour Procmail, qui sont référencés dans leur fichier `.procmailrc` présent dans leur répertoire personnel.

Par défaut, aucun fichier `rc` s'appliquant à l'ensemble du système n'existe dans le répertoire `/etc` et aucun fichier `.procmailrc` n'existe dans le répertoire personnel de quelque utilisateur que ce soit. Par conséquent, pour utiliser Procmail, chaque utilisateur doit créer un fichier `.procmailrc` contenant des variables d'environnement et des règles spécifiques.

13.4.1. Configuration de Procmail

Les fichiers de configuration de Procmail contiennent des variables d'environnement importantes. Ces dernières précisent à Procmail les messages spécifiques devant être triés et le sort des messages qui ne satisfont aucune recette.

Ces variables d'environnement qui se trouvent généralement au début du fichier `.procmailrc` ont le format suivant :

```
<env-variable>="<value>"
```

Dans cet exemple, `<env-variable>` correspond au nom de la variable alors que l'élément `<value>` définit la variable elle-même.

La plupart des utilisateurs de Procmail De nombreuses n'utilisent pas de nombreuses variables d'environnement mais les plus importantes sont déjà définies par une valeur par défaut. La plupart du temps, les variables suivantes sont utilisées :

- `DEFAULT` — Définit la boîte aux lettres où seront placés les messages qui ne satisfont aucune recette.

  La valeur par défaut de `DEFAULT` est la même que `$ORGMAIL`.

- `INCLUDERC` — Spécifie des fichiers `rc` supplémentaires qui contiennent d'autres recettes que les messages doivent satisfaire. Ceci permet de diviser les listes de recettes Procmail en fichiers individuels qui jouent différents rôles, tels que le blocage de spams et la gestion de listes d'adresses électroniques, qui peuvent ensuite être activés ou désactivés à l'aide de caractères de commentaire dans le fichier `.procmailrc` de l'utilisateur.

  Par exemple, des lignes présentes dans un fichier `.procmailrc` de l'utilisateur peuvent ressembler à l'extrait suivant :

```
MAILDIR=$HOME/Msgs
INCLUDERC=$MAILDIR/lists.rc
INCLUDERC=$MAILDIR/spam.rc
```

  Si l'utilisateur souhaite désactiver le filtrage par Procmail de ses listes d'adresses, mais désire garder le contrôle du pourriel, il n'a qu'à commenter la première ligne `INCLUDERC` avec le symbole dièse (`#`).

- `LOCKSLEEP` — Définit la durée, en secondes, s'écoulant entre les tentatives de Procmail d'utiliser un fichier de verrouillage spécifique. La valeur par défaut est de huit secondes.
- `LOCKTIMEOUT` — Définit la durée, en secondes, qui doit s'écouler après la dernière modification d'un fichier de verrouillage avant que Procmail ne considère le fichier de verrouillage comme étant vieux et pouvant par conséquent être supprimé. La valeur par défaut est de 1024 secondes.
- `LOGFILE` — Représente le fichier dans lequel toutes les informations de Procmail ou tous les messages d'erreurs sont enregistrés.
- `MAILDIR` — Définit le répertoire de travail courant pour Procmail. Si cete variable est déterminée, tous les autres chemins d'accès vers Procmail sont relatifs à ce répertoire.
- `ORGMAIL` — Spécifie la boîte aux lettres originale ou un autre endroit où placer les messages s'ils ne peuvent être placés à l'emplacement par défaut ou à celui exigé par les recettes.

  Par défaut, une valeur de `/var/spool/mail/$LOGNAME` est utilisée.

- `SUSPEND` — Définit la durée, en secondes, pendant laquelle Procmail s'arrête si une ressource nécessaire, telle que l'espace swap, n'est pas disponible.
- `SWITCHRC` — Permet à un utilisateur de spécifier un fichier externe contenant des recettes Procmail supplémentaires, plus ou moins comme le fait l'option `INCLUDERC`, sauf que la vérification des recettes est arrêtée sur le fichier de configuration traitant et que seules les recettes du fichier spécifié avec `SWITCHRC` sont utilisées.
- `VERBOSE` — Fait en sorte que Procmail journalise davantage d'informations. Cette option est pratique pour le débogage.

D'autres variables d'environnement importantes sont obtenues depuis le shell, comme `LOGNAME`, qui est le nom de connexion ; `HOME`, qui est l'emplacement du répertoire personnel ; et `SHELL`, qui est le shell par défaut.

Consultez la page de manuel de `procmailrc` pour obtenir des explications exhaustives sur les variables d'environnement ainsi que sur leurs valeurs par défaut.

13.4.2. Recettes Procmail

Les nouveaux utilisateurs trouvent généralement que les recettes constituent l'élément le plus difficile de l'apprentissage de l'utilisation de Procmail. Ce sentiment est compréhensible jusqu'à un certain point, étant donné que les recettes effectuent la comparaison avec les messages à l'aide d'*expressions régulières*, qui est un format spécifique utilisé pour spécifier des qualifications de concordance de chaînes. Ceci étant, les expressions régulières ne sont pas très difficiles à créer et sont encore moins difficiles à comprendre en les lisant. De plus, la cohérence avec laquelle les recettes Procmail sont écrites, sans tenir compte des expressions régulières, permet d'acquérir de bonnes connaissances facilement, simplement en examinant les exemples.

Les recettes Procmail se présentent sous la forme suivante :

```
:0<flags>: <lockfile-name>

* <special-condition-character> <condition-1>
* <special-condition-character> <condition-2>
* <special-condition-character> <condition-N>

<special-action-character><action-to-perform>
```

Les deux premiers caractères d'une recette Procmail sont le symbole des deux-points et un zéro. Divers indicateurs (ou flags) peuvent être placés après le zéro pour contrôler la manière selon laquelle Procmail traite la recette. Le symbole des deux-points placé après la section `<flags>` spécifie qu'un fichier de verrouillage (lockfile) sera créé pour ce message. Si un fichier de verrouillage est créé, le nom peut être spécifié dans l'espace `<lockfile-name>`.

Une recette peut contenir plusieurs conditions servant à vérifier la concordance d'un message. S'il aucune condition n'est spécifiée, tous les messages auront une concordance positive avec la recette. Les expressions régulières sont placées dans certaines conditions de façon à faciliter la concordance avec les messages. Si des conditions multiples sont utilisées, elles doivent toutes obtenir la concordance pour que l'action soit exécutée. Les conditions sont vérifiées sur la base des indicateurs spécifiés dans la première ligne de la recette. Des caractères spéciaux facultatifs placés après le caractère `*` permettent de contrôler ultérieurement la condition.

L'option `<action-to-perform>` spécifie l'action exécutée lorsque le message correspond à l'une des conditions. Il ne peut y avoir qu'une action par recette. Dans de nombreux cas, le nom d'une boîte aux lettres est utilisé ici pour envoyer dans ce fichier les messages satisfaisant les conditions, permettant ainsi de trier le courrier. Des caractères d'action spéciaux peuvent également être utilisés avant que l'action ne soit spécifiée.

13.4.2.1. Recettes de distribution et de non-distribution

L'action utilisée si la recette correspond à un message donné détermine si cette dernière est considérée comme étant une recette de *distribution* ou de *non-distribution*. Une recette de distribution contient une action qui écrit le message dans un fichier, envoie le message à un autre programme ou réachemine le message vers une autre adresse électronique. Une recette de non-distribution couvre toutes les autres actions, telles que l'utilisation d'un *bloc d'imbrication* (également appelé nesting block). Un bloc d'imbrication est un ensemble

d'actions contenues entre deux accolades, `{ }`, qui sont exécutées sur des messages satisfaisant les conditions de la recette. Les blocs d'imbrication peuvent être emboîtés les uns dans les autres, offrant ainsi plus de contrôle pour l'identification et l'exécution d'actions sur des messages.

Lorsque des messages satisfont une recette de distribution, Procmail effectue l'action spécifiée et arrête de comparer le message à toute autre recette. Les messages qui satisfont les recettes de non-distribution continuent eux à être comparés aux autres recettes.

13.4.2.2. Indicateurs

Les indicateurs (ou flags) sont très importants pour déterminer la façon dont les conditions d'une recette sont comparées à un message. Les indicateurs suivants sont couramment utilisés :

- `A` — Spécifie que cette recette n'est utilisée que si la recette précédente sans indicateur `A` ou `a` a également obtenu la concordance avec ce message.
- `a` — Spécifie que cette recette n'est utilisée que si la recette précédente sans indicateur `A` ou `a` a également obtenu la concordance avec ce message *et* a été exécutée avec succès.
- `B` — Analyse le corps du message et recherche des conditions de concordance.
- `b` — Utilise le corps du message dans toute action découlant de cet indicateur, comme l'écriture du message dans un fichier ou son réacheminement. Il s'agit du comportement par défaut.
- `c` — Génère une copie conforme du message électronique. Cette option peut être utile avec les recettes de distribution, étant donné que l'action requise peut être exécutée sur le message et que la copie du message peut continuer à être traitée dans les fichiers `rc`.
- `D` — Rend la comparaison `egrep` sensible à la casse. Par défaut, le processus de comparaison n'est pas sensible à la casse.
- `E` — Semblable à l'indicateur `A` sauf que les conditions dans cette recette sont comparées à un message seulement si la recette précédant immédiatement la recentte sans indicateur `E` n'a pas obtenu la concordance. Cette action ressemble à une action *else*.
- `e` — Établit la comparaison de la recette au message seulement si l'action spécifiée dans la recette présente juste avant échoue.
- `f` — Utilise le tube (aussi appelé pipe) comme filtre.
- `H` — Analyse l'en-tête du message et recherche des conditions de concordance. Cette situation se produit par défaut.
- `h` — Utilise l'en-tête dans une action découlant de cet indicateur. Ce dernier représente le comportement par défaut.
- `w` — Indique à Procmail d'attendre que le filtre ou le programme spécifiés aient terminé leurs opérations et rapporte si l'opération précédente a réussi ou échoué, avant de considérer le message comme étant filtré.
- `W` — Identique à `w` sauf que les messages de type "Échec du programme" (ou Program failure) sont supprimés.

Pour obtenir une liste détaillée d'indicateurs supplémentaires, reportez-vous à la page de manuel de `procmailrc`.

13.4.2.3. Spécification d'un fichier de verrouillage local

Les fichiers de verrouillage sont très utiles avec Procmail pour garantir que seul un processus essaie de modifier un certain message à un moment donné. Il est possible de spécifier un fichier de verrouillage local en plaçant le symbole des deux points (`:`) après tout indicateur dans la première ligne d'une recette. Ce faisant, un fichier de verrouillage local est créé en fonction du nom de fichier de destination et de toute valeur contenue dans la variable d'environnement globale `LOCKEXT`.

Vous pouvez aussi spécifier le nom du fichier de verrouillage local à utiliser avec cette recette après le symbole des deux points (:).

13.4.2.4. Conditions et actions spéciales

Des caractères spéciaux utilisés avant les conditions de recettes et et avant les actions de Procmail modifient la façon selon aquelle elles sont interprétées.

Les caractères suivants peuvent être utilisés après le symbole `*` au début de la ligne de condition d'une recette :

- `!` — Placé dans la ligne de condition, ce caractère inverse la condition, de sorte que la concordance sera désormais établie seulement si la condition ne satisfait pas le message.
- `<` — Vérifie si la taille du message est inférieure à un nombre d'octets spécifié.
- `>` — Vérifie si la taille du message est supérieure à un nombre d'octets spécifié.

Les caractères suivants sont utilisés pour exécuter des actions spéciales :

- `!` — Placé dans la ligne d'action, ce caractère indique à Procmail de réacheminer le message vers les adresses électroniques spécifiées.
- `$` — Renvoie à une variable définie précédemment dans le fichier `rc`. Cette option est généralement utilisée pour définir une boîte aux lettres commune à laquelle diverses recettes feront référence.
- `|` — Démarre un programme spécifié afin qu'il traite le message.
- `{` and `}` — Construit un bloc d'imbrication, utilisé pour contenir des recettes supplémentaires devant être appliquer aux messages satisfaisant les conditions.

Si aucun caractère spécial n'est utilisé au début de la ligne d'action, Procmail considère que la ligne d'action spécifie la boîte aux lettres où les messages doivent être déposés.

13.4.2.5. Exemples de recettes

Procmail est certes un programme extrêmement flexible, mais en raison de cette flexibilité, la création d'une recette Procmail de toutes pièces peut être une tâche difficile pour de nouveaux utilisateurs.

Le meilleure façon de développer les capacités nécessaires pour créer les conditions des recettes Procmail consiste à bien comprendre la notion d'expressions régulières et à examiner de nombreux exemples élaborés par d'autres. Une explication exhaustive des expressions

régulières va au-delà de la portée de cette section. La structure des recettes Procmail ainsi que des exemples de recettes Procmail sont disponibles à différents endroits sur Internet (notamment à l'adresse suivante : http://www.iki.fi/era/procmail/links.html). En examinant ces exemples de recettes, il est possible d'acquérir des connaissances solides sur la bonne utilisation et sur l'adaptation des expressions régulières. En outre, des informations élémentaires sur des règles d'expressions régulières de base se trouvent dans la page de manuel de `grep`.

Les simples exemples reproduits ci-dessous illustrent la structure 'élémentaire de recettes Procmail et peuvent servir de base pour des conceptions plus élaborées.

Une recette élémentaire ne contient pas forcément de conditions, comme le montre l'exemple ci-dessous :

```
:0:
new-mail.spool
```

La première ligne spécifie qu'un fichier de verrouillage local doit être créé, mais n'indique aucun nom. Procmail utilise donc le nom du fichier de destination et y ajoute la valeur spécifiée dans la variable d'environnement `LOCKEXT`. Étant donné qu'aucune condition n'est spécifiée, tous les messages satisfont cette recette et sont par conséquent placés dans le fichier spoule unique appelé `new-mail.spool` qui se trouve dans le répertoire spécifié par la variable d'environnement `MAILDIR`. Un AGC peut ensuite visualiser les messages dans ce fichier.

Une recette de base, telle que celle-ci, peut être placée à la fin de tous les fichiers `rc` afin que les messages soient acheminés vers un emplacement par défaut.

L'exemple ci-dessous illustre l'etablissement de la concordance avec des messages d'une adresse électronique spécifique et le dépôt de ces derniers dans la corbeille.

```
:0
* ^From: spammer@domain.com
/dev/null
```

Dans cet exemple, tout message envoyé par `spammer@domain.com` est acheminé vers le périphérique`/dev/null` où il est supprimé.

Avertissement

Assurez-vous que les règles fonctionnent bien comme vous le désirez avant d'acheminer les messages concernés vers `/dev/null` afin qu'ils soient supprimés de façon permanente. Si les conditions de votre recette retiennent accidentellement des messages qui ne devraient pas l'être et qu'ils disparaissent sans laisser de trace, il est alors difficile de résoudre tout problème lié à la règle.

Une solution plus appropriée consiste à pointer l'action de la recette vers une boîte aux lettres spéciale qui peut être vérifiée de temps en temps, afin de voir si elle contient de fausses concordances. Une fois convaincu qu'aucun message ne fait l'objet d'une concordance accidentelle, supprimez la boîte aux lettres et établissez l'action de façon à ce qu'elle envoie les messages vers `/dev/null`.

La recette ci-dessous retient les messages envoyés depuis une liste de diffusion spécifique et les place dans un dossier déterminé.

```
:0:
* ^(From|CC|To).*tux-lug
tuxlug
```

Tout message envoyé depuis la liste de diffusion `tux-lug@domain.com` est automatiquement placé dans la boîte aux lettres `tuxlug` pour le AGC. Notez que la condition dans cet exemple a une concordance avec le message si l'adresse électronique de la liste de diffusion se trouve sur l'une des lignes suivantes : `From` (De), `CC` ou `To` (À).

13.4.2.6. Filtres de spam

Puisque Procmail est appelé par Sendmail, Postfix et Fetchmail lors de la réception de nouveaux messages, il peut être utilisé comme un outil puissant pour combattre le pourriel.

Le combat contre le pourriel est encore plus efficace lorsque Procmail est utilisé de concert avec SpamAssassin. En effet, grâce à une double action ces deux applications peuvent rapidement identifier des messages-pourriel, les trier et les détruire.

SpamAssassin recours à une analyse de l'en-tête et du texte, à des listes noires et à des bases de données de localisation de spam ainsi qu'à une analyse bayesienne auto-organisatrice de pourriel pour identifier et étiqueter rapidement et précisément tout pourriel (aussi appelé spam).

Pour un utilisateur local, la meilleure façon d'utiliser SpamAssassin consiste à insérer la ligne suivante vers le haut du fichier `~/.procmailrc` :

```
INCLUDERC=/etc/mail/spamassassin/spamassassin-default.rc
```

Le programme `/etc/mail/spamassassin/spamassassin-default.rc` contient une simple règle Procmail permettant d'activer SpamAssassin pour tout courrier électronique reçu. Si un message est reconnu comme étant un pourriel, il est étiqueté en tant que tel dans l'en-tête et la mention suivante est ajoutée au sujet :

```
*****SPAM*****
```

Le corps du message de l'email est précédé d'un compte-rendu des éléments ayant justifié le diagnostic de spam.

Pour classer les emails étiquetés en tant que pourriel, il est possible d'utiliser une règle semblable à celle reproduite ci-dessous :

```
:0 Hw
* ^X-Spam-Status: Yes
spam
```

Selon cette règle, tous les messages étiquetés en tant que spam dans l'en-tête sont rangés dans une boîte aux lettres nommée `spam`.

Étant donné que SpamAssassin est un script Perl, il sera peut-être nécessaire d'utiliser le démon binaire SpamAssassin (`spamd`) et l'application client (`spamc`) sur des serveurs très sollicités. Pour configurer SpamAssassin de la sorte, il est nécessaire d'avoir un accès super-utilisateur à l'hôte.

Pour lancer le démon `spamd`, tapez la commande suivante en étant connecté en tant que super-utilisateur (ou root) :

```
/sbin/service spamassassin start
```

Pour lancer le démon SpamAssassin lors du démarrage du système, utilisez un utilitaire initscript, comme l'**Outil de configuration des services** (`system-config-services`), pour activer le service `spamassassin`.

Pour configurer Procmail afin qu'il utilise l'application client SpamAssassin au lieu du script Perl, placez le ligne suivante vers le haut du fichier `~/.procmailrc`. Pour une configuration s'appliquant à tout le système, placez cette dernière dans `/etc/procmailrc` :

```
INCLUDERC=/etc/mail/spamassassin/spamassassin-spamc.rc
```

13.5. Ressources supplémentaires

Ci-dessous figure une liste de la documentation supplémentaire relative aux applications de messagerie.

13.5.1. Documentation installée

- Les paquetages `sendmail` and `sendmail-cf` contiennent des informations sur la manière de configurer Sendmail.
    - `/usr/share/doc/sendmail/README.cf` — Contient entre autres des informations sur `m4`, sur les emplacements des fichiers pour Sendmail, sur les boîtes de messagerie prises en charge et sur les façons d'accéder à des fonctionnalités avancées.

  En outre, les pages de manuel de `sendmail` et `aliases` contiennent des informations utiles sur les différentes options de Sendmail et sur la configuration adéquate du fichier Sendmail, `/etc/mail/aliases`.

- `/usr/share/doc/postfix-<version-number>` — Contient de nombreuses informations sur la manière de configurer Postfix. Remplacez `<version-number>` par le numéro de version de Postfix.
- `/usr/share/doc/fetchmail-<version-number>` — Contient une liste complète des fonctions Fetchmail dans le fichier `FEATURES` et un document `FAQ` d'introduction. Remplacez `<version-number>` par le numéro de version de Fetchmail.
- `/usr/share/doc/procmail-<version-number>` — Contient un fichier `README` qui offre un aperçu de Procmail, un fichier `FEATURES` qui explore toutes les fonctions du programme et un fichier `FAQ` qui offre les réponses à de nombreuses questions de configuration courantes. Remplacez `<version-number>` par le numéro de version de Procmail.

Lors de l'apprentissage de la manière selon laquelle Procmail fonctionne et de la façon de créer de nouvelles recettes, les pages de manuel suivantes sont d'une aide très précieuse :

- `procmail` — Offre un aperçu du fonctionnement de Procmail et des étapes de filtrage du courrier.
- `procmailrc` — Explique le format de fichier `rc` utilisé pour créer des recettes.
- `procmailex` — Donne des exemples pratiques utiles de recettes Procmail.
- `procmailsc` — Explique la technique de weighted scoring utilisée par Procmail pour vérifier s'il y a concordance entre une recette donnée et un message.
- `/usr/share/doc/spamassassin-<version-number>/` — Contient de nombreuses informations sur SpamAssassin. Remplacez `<version-number>` par le numéro de version du paquetage `spamassassin`.

13.5.2. Sites Web utiles

- http://www.redhat.com/mirrors/LDP/HOWTO/Mail-Administrator-HOWTO.html — Fournit un aperçu du fonctionnement du courrier électronique et examine les solutions et configurations possibles de messagerie électronique, tant du côté serveur que client.
- http://www.redhat.com/mirrors/LDP/HOWTO/Mail-User-HOWTO/ — Examine le courrier électronique du point de vue de l'utilisateur, analyse diverses applications client de messagerie très utilisées et offre une introduction sur des sujets variés, tels que les alias, le réacheminement, la réponse automatique, les listes d'adresses, les filtres de courrier et le pourriel.
- http://www.redhat.com/mirrors/LDP/HOWTO/mini/Secure-POP+SSH.html — Explique une façon de télécharger du courrier POP en utilisant SSH avec le réacheminement de port, afin que les mots de passe et les messages soient transférés de manière sécurisée.
- http://www.sendmail.net/ — Contient des informations récentes, entrevues et articles relatifs à Sendmail, notamment un aperçu détaillé des nombreuses options disponibles.
- http://www.sendmail.org/ — Offre une explication technique très détaillée des fonctions de Sendmail et des exemples de configuration.
- http://www.postfix.org/ — Représente la page d'accueil du projet Postfix contenant de nombreuses informations sur Postfix. La liste de diffusion représente une excellente source d'informations.
- http://catb.org/~esr/fetchmail/ — Représente la page d'accueil de Fetchmail, comprenant un manuel en ligne et un forum aux questions (FAQ) complet.
- http://www.procmail.org/ — Représente la page d'accueil de Procmail, avec des liens menant à diverses listes d'adresses de participants dédiées à Procmail, de même que de nombreux documents FAQ.
- http://www.ling.helsinki.fi/users/reriksso/procmail/mini-faq.html — Constitue un excellent FAQ sur Procmail, offrant des conseils en matière de résolution de problèmes, des informations sur le verrouillage de fichiers et l'utilisation de caractères génériques (aussi appelés wildcards).
- http://www.uwasa.fi/~ts/info/proctips.html — Contient de nombreux conseils rendant l'utilisation de Procmail plus aisée. Ce site inclut des instructions sur la manière de tester les fichiers `.procmailrc` et d'utiliser le marquage de Procmail pour décider si une action donnée doit être exécutée ou non.
- http://www.spamassassin.org/ — Représente le site officiel du projet SpamAssassin.

13.5.3. Livres sur le sujet

- *Sendmail* de Bryan Costales avec Eric Allman et al ; O'Reilly & Associates — Une bonne référence pour Sendmail, écrite avec l'aide du créateur original de Delivermail et Sendmail.
- *Removing the Spam : Email Processing and Filtering* de Geoff Mulligan ; Addison-Wesley Publishing Company — Un livre examinant les diverses méthodes utilisées par les administrateurs de messagerie ayant recours à des outils établis, tels que Sendmail et Procmail, pour gérer les problèmes causés par le pourriel.
- *Internet Email Protocols : A Developer's Guide* de Kevin Johnson ; Addison-Wesley Publishing Company — Un recueil d'informations détaillées sur les principaux protocoles de messagerie et la sécurité offerte par ceux-ci.
- *Managing IMAP* de Dianna Mullet et Kevin Mullet ; O'Reilly & Associates — Une explication des étapes nécessaires à la configuration d'un serveur IMAP.
- *Guide de sécurité de Red Hat Enterprise Linux* ; Red Hat, Inc. — Le chapitre intitulé *Sécurité de serveur* explique différentes manières de sécuriser Sendmail et d'autres services.

Section 14 Installation d'un service Web-mail

Agent Utilisateurs de Messagerie basés sur *HTTP*

Table of Contents

Présentation
Architecture générale du service
Installation et configuration OpenWebmail
    Préparation de la machine
    Installation d'OpenWebmail
    Configuration de l'application OpenWebmail
    Test de l'environnement
    Configuration de l'environnement utilisateur
    Test et environnement OpenWebmail
Application

14.1 Présentation

Il est préférable d'avoir réalisé les ateliers sur les serveurs HTTP, SMTP et DNS avant de commencer celui-ci.

Le service Web-mail permet l'utilisation d'un service de messagerie à partir d'un client Web comme mozilla. Cette interface est intéressante, car contrairement à un client pop3 standard qui serait configuré pour rapatrier les courriers sur la machine locale, ceux-ci, vont rester sur le serveur. Vous pouvez les consulter à partir de n'importe quel poste pourvu qu'il dispose d'un navigateur. Vous aurez ensuite tout le loisir de les récupérer avec votre client de messagerie préféré si vous en utilisez un.

Il existe un très grand nombre de serveur Web-mail, et écris dans des langages très différents. Une étude a été réalisée par le CRU http://www.cru.fr/http-mail/, mais parmi les principaux on peut citer IMP écris en PHP et qui s'appuie sur la librairie horde. Il existe aussi OpenWebmail qui lui est écris en Perl. Ces deux produits existent en paquets Debian, on utilisera pour le TP OpenWebmail, mais ces deux outils comportent chacun de nombreuses qualités, le choix devra se faire en fonction du degré d'intégration que vous souhaiterez obtenir avec vos autres applications.

14.2 Architecture générale du service

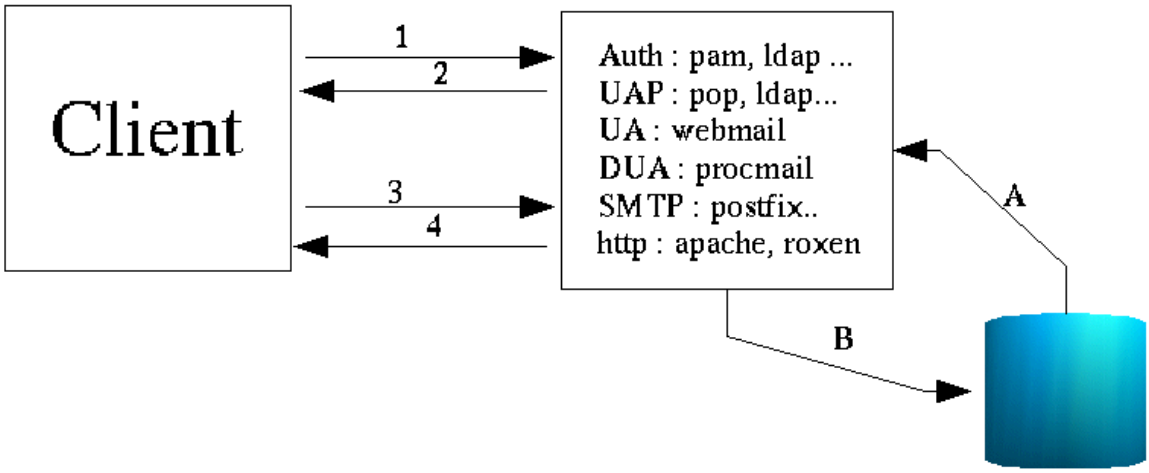


Figure 14.1. Architecture globale d'un service Web-mail

1. En 1 et 2, le client passe par une phase préalable d'authentification, il faut donc un service correspondant sur le serveur. Cela peut être pris directement en charge par le service Web-mail ou par un service extérieur (pam, ldap par exemple). (Nous utiliserons l'authentification pam).
2. En 3 et 4, le dialogue s'effectue en un navigateur et un serveur HTTP. Le dialogue peut s'effectuer dans un canal SSL ou TLS. Le suivi de session peut être réalisé à l'aide de cookie par exemple. (Nous utiliserons Apache comme serveur HTTP).
3. En A et B, le service Web-mail utilise une base de données pour les boîtes aux lettres des utilisateurs, pour les dossiers (`inbox`, `outbox`, `trash`...) et la conservation des courriers. Certains Web-mail s'appuient sur des bases de type MySQL, PostgreSQL... ou simplement sur une arborescence de répertoires dans le `HOME_DIRECTORY` de l'utilisateur. (Nous n'utiliserons pas de SGBD/R).
4. Sur le serveur, il faut activer un protocole de traitement du courrier (pop3, pop3s, imap, imaps...). Nous utiliserons imap et pop3.
5. Vous aurez également besoin d'un service SMTP pour le traitement des courriers sortants (nous utiliserons postfix) et d'un service de livraison (DUA) (nous utiliserons procmail) pour délivrer les courriers entrants.

14.3 Installation et configuration OpenWebmail

Vous allez installer OpenWebmail mais auparavant il est nécessaire de s'assurer du bon fonctionnement de certains services. (smtp, mail, procmail, apache...)

14.3.1 Préparation de la machine

Suivez la procédure ci-dessous pour préparer la machine.

Configuration générale

On considère la configuration suivante, vous adapterez les noms, adresses IP et autres paramètres à votre configuration. Il n'y a pas de DNS.

```
Nom d'hôte : freeduc-sup ($NAME)
Nom FQDN de la machine : freeduc-sup.foo.org ($FQDN)
Adresse de réseau : 192.168.0.0
Adresse de la machine : 192.168.0.2
Adresse de la passerelle par défaut : 192.168.0.254
```

14.3.1.1 Test de la résolution de nom

Vérifier que la résolution de nom fonctionne parfaitement. Les commandes : **ping `$NAME`** et **ping `$FQDN`** doivent répondre correctement.

```
# Exemple de fichier /etc/hosts
127.0.0.1                    freeduc-sup   freeduc-sup.foo.org   localhost
localhost.localdomain
192.168.0.2    freeduc-sup  freeduc-sup.foo.or
```

14.3.1.2 Le service Apache

Activez le service apache. Il ne doit pas y avoir de message d'erreur au lancement, notamment sur la résolution de nom. Vérifiez également le bon fonctionnement avec une requête sur : http://localhost

14.3.1.3 Le service SMTP (Postfix)

Pour configurer postfix, utilisez la commande **`dpkg-reconfigure` *`postfix`***, vous prendrez site internet. Les valeurs par défaut doivent normalement fonctionner.

Ouvrez le fichier `/etc/postfix/main.cf`, vérifiez qu'il correspond à celui-ci, au besoin modifiez le :

```
# Fichier de configuration de Postfix
# Adaptez vos noms d'hôtes et vos noms de machines

# see /usr/share/postfix/main.cf.dist for a commented, fuller
# version of this file.

# Do not change these directory settings - they are critical to Postfix
# operation.
command_directory = /usr/sbin
daemon_directory = /usr/lib/postfix
program_directory = /usr/lib/postfix

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
setgid_group = postdrop
biff = no

myhostname = freeduc-sup.foo.org
mydomain = foo.org
myorigin = $myhostname
inet_interfaces = all
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = $myhostname, localhost.$mydomain
relayhost =
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
```

Une fois cela réalisé, activez ou relancez le service.

```
/etc/init.d/postfix start
/etc/init.d/postfix reload
```

14.3.1.4 Activation des services imap

Cela s'effectue dans le fichier `inet.conf`. Il faudra adapter si vous utilisez xinetd. Cela dépend de la distribution de GNU/Linux que vous utilisez. Vous pouvez également utiliser la commande **`dpkg-reconfigure uw-imapd`**. Dans ce cas, prenez imap2 (qui correspond à imap4 (? ;-))) et imaps.

Extrait d'un exemple de configuration de `inetd.conf` :

```
#:MAIL: Mail, news and uucp services.
imap2   stream  tcp     nowait  root    /usr/sbin/tcpd /usr/sbin/imapd
```

Adaptez votre fichier de configuration, puis relancer inetd avec la commande :

```
/etc/init.d/inetd restart
```

Vérifiez que les ports sont bien ouverts avec la commande **netstat**. Vous devez avoir les ports 25 (pop3)si vous l'avez activé et 143 (imap) ouverts.

```
netstat -atup | grep LISTEN
tcp     0      0 *:netbios-ssn       *:*      LISTEN      269/smbd
tcp     0      0 *:imap2             *:*      LISTEN      263/inetd
tcp     0      0 *:sunrpc            *:*      LISTEN      151/portmap
tcp     0      0 *:ssh               *:*      LISTEN      278/sshd
tcp     0      0 *:ipp               *:*      LISTEN      290/cupsd
tcp     0      0 *:smtp              *:*      LISTEN      899/master
```

Nous voyons imap2, ligne 2, pris en charge par inetd.

```
root@freeduc-sup:/home/mlx# netstat -natup | grep LISTEN
tcp   0      0 0.0.0.0:139   0.0.0.0:*       LISTEN      269/smbd
tcp   0      0 0.0.0.0:143   0.0.0.0:*       LISTEN      263/inetd
tcp   0      0 0.0.0.0:111   0.0.0.0:*       LISTEN      151/portmap
tcp   0      0 0.0.0.0:22    0.0.0.0:*       LISTEN      278/sshd
tcp   0      0 0.0.0.0:631   0.0.0.0:*       LISTEN      290/cupsd
tcp   0      0 0.0.0.0:25    0.0.0.0:*       LISTEN      899/master
```

Ici l'option `-n` de **netstat** nous indique les numéros de ports utilisés.

Remarque : si vous souhaitez utiliser un client pop, vous devrez activer également le protocole pop.

14.3.1.5 Test des services

A ce stade, il nous est possible de tester complètement le service de messagerie. Vous pouvez indifféremment utiliser un client comme kmail ou Mozilla. Le plus simple est d'utiliser le client mail.

Suivez la procédure ci-dessous :

1. Créez deux comptes utilisateurs alpha et beta qui serviront pour les tests avec la commande **adduser**.
2. Testez avec la commande **mail** que l'envoie de courrier se déroule correctement. Les commandes :

```
mail alpha
mail alpha@freeeduc-sup
mail alpha@freeduc-sup.foo.org
```

doivent fonctionner correctement.

14.3.2 Installation d'OpenWebmail

Si vous utilisez la freeduc-sup, OpenWebmail n'est pas installé. Utilisez la commande :

```
apt-get install openwebmail
```

La commande installera également 3 paquets supplémentaires qui correspondent aux dépendances puis lancera la procédure de configuration.

14.3.3 Configuration de l'application OpenWebmail

Vous pouvez à tout moment reconfigurer l'application avec **dpkg-reconfigure *openwebmail***. Prenez comme option :

```
authentification -> auth_pam.pl
langage -> fr
```

14.3.4 Test de l'environnement

Pour tester l'environnement vous avez deux liens :

```
http://localhost/openwebmail
```

qui vous place sur un espace documentaire

```
http://localhost/cgi-bin/openwebmail/openwebmail.pl
```

qui lance l'application proprement dite et vous amène sur la première fenêtre de login

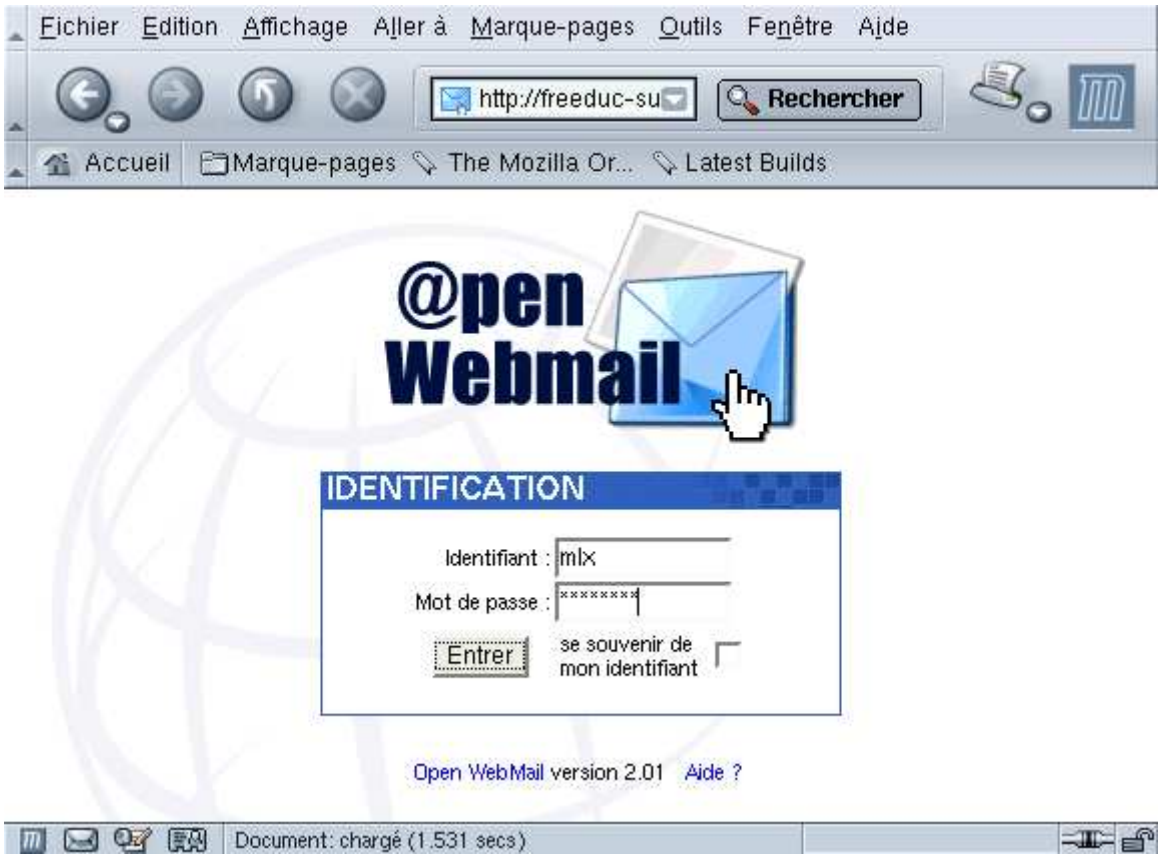


Figure 14.2. Ouverture de session sur un Web-mail

Il vous sera possible de créer un serveur Web virtuel pour avoir par exemple : " openwebmail.freeduc-sup.org ".

14.3.5 Configuration de l'environnement utilisateur

A la première session, la personne reçoit une invite lui permettant de configurer son environnement et ses paramètres particuliers, comme son adresse de réponse, modifier son mot de passe... Ces paramètres sont modifiables à tout moment.

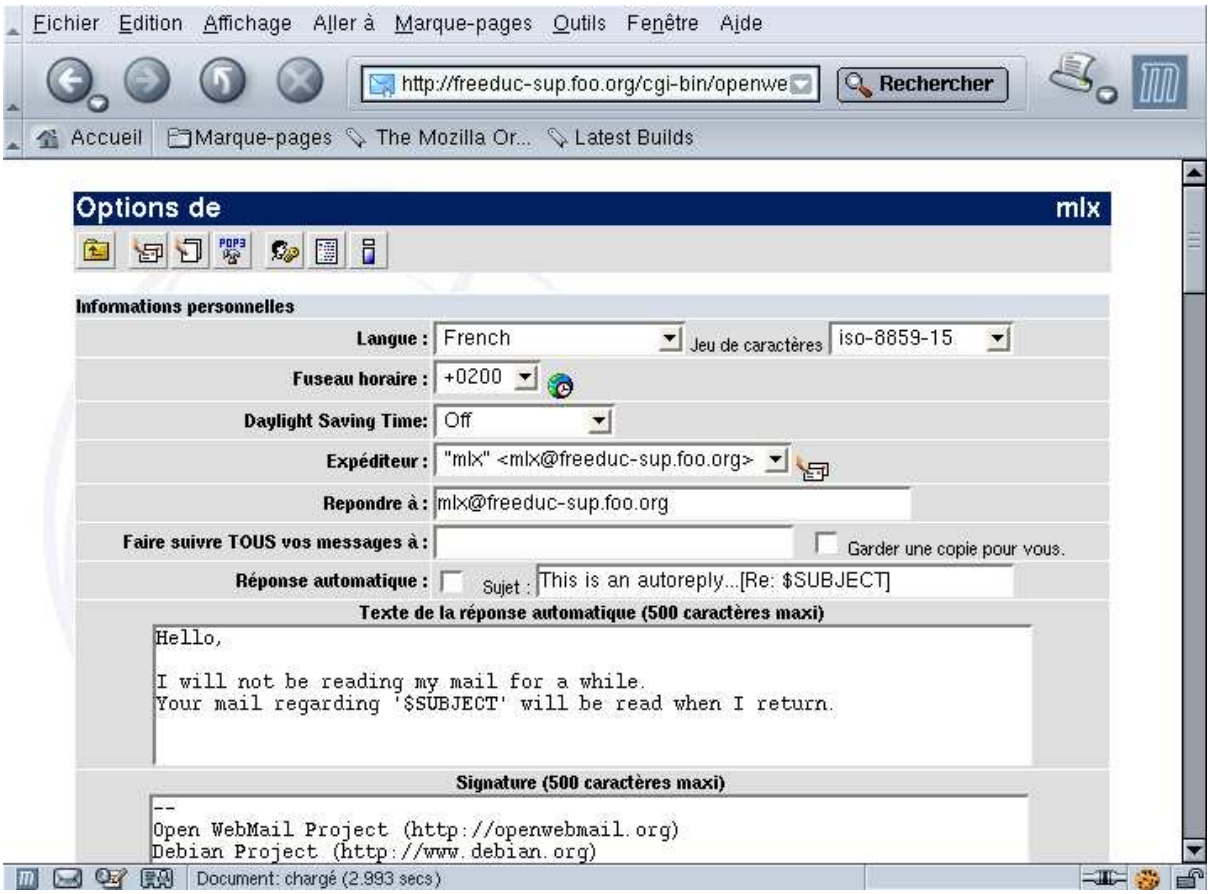


Figure 14.3. Configuration de l'environnement utilisateur

14.3.6 Test et environnement OpenWebmail

A partir de ce moment, l'environnement complet est disponible. L'utilisateur dispose également d'un calendrier et d'une documentation en ligne.

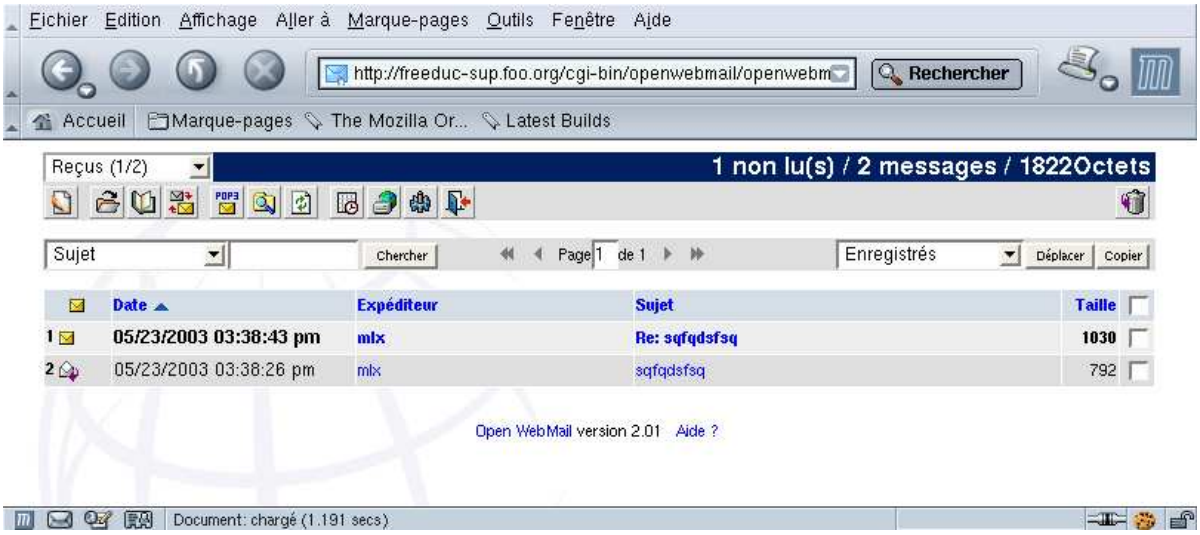


Figure 14.4. Voir ses messages

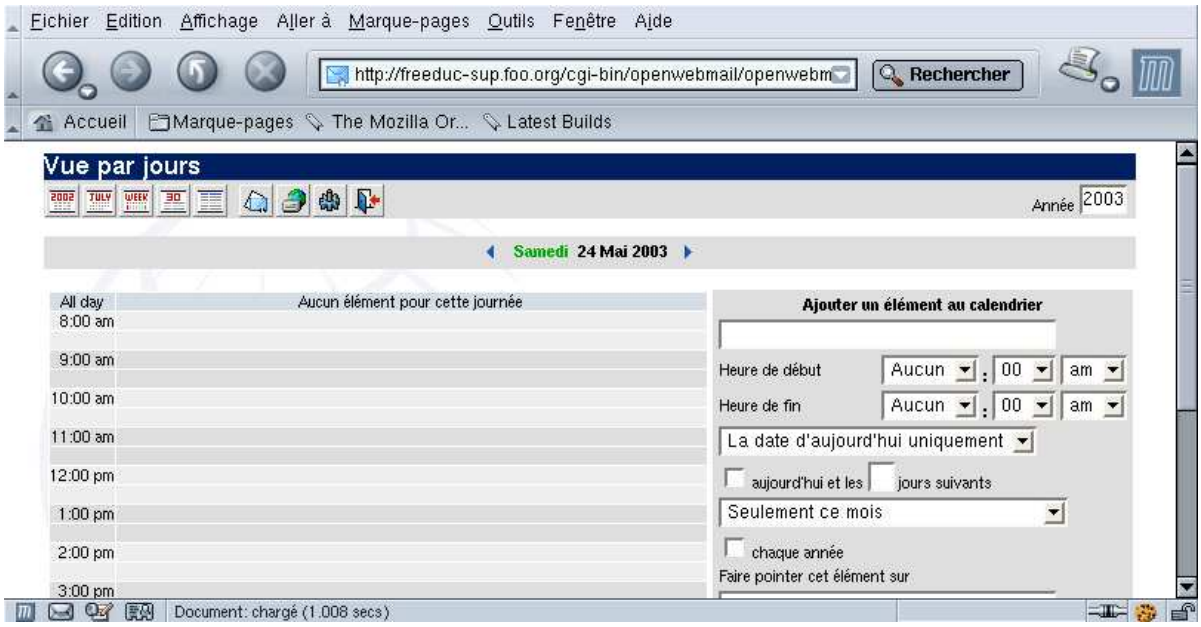


Figure 14.5. Le calendrier

Figure 14.6. L'aide en ligne

14.4 Application

1. Configurez et vérifiez le bon fonctionnement des services de résolution de nom, apache, postfix.
2. Installez et configurez OpenWebmail.
3. Testez le fonctionnement OpenWebmail.

Section 15 Protocole LDAP (Lightweight Directory Access Protocol)

Le protocole *Lightweight Directory Access Protocol* (ou *LDAP*) est en fait un ensemble de protocoles ouverts utilisés pour accéder à des informations stockées localement sur un réseau. Il est basé sur le standard *X.500* pour le partage de répertoires, mais est moins complexe et moins gourmand en ressources, d'où la référence à LDAP sous le terme "*X.500 Lite*." Le standard X.500 est un répertoire qui contient des informations hiérarchisées et organisées pouvant inclure des renseignements tels que des noms, des adresses et des numéros de téléphone.

Comme X.500, LDAP organise des informations d'une manière hiérarchique en utilisant des répertoires. Ces répertoires peuvent stocker diverses informations et peuvent même être utilisés d'une manière semblable au service d'informations réseau (ou NIS de l'anglais Network Information Service), permettant à tout un chacun d'accéder à son compte depuis une machine quelconque présente sur un réseau sous LDAP.

Dans la plupart des cas, LDAP sert d'annuaire téléphonique virtuel, permettant aux utilisateurs d'accéder facilement aux coordonnées d'autres utilisateurs. Mais le protocole LDAP est beaucoup plus flexible qu'un annuaire téléphonique traditionnel car il peut renvoyer un demandeur vers d'autres serveurs LDAP de par le monde, fournissant ainsi un référentiel d'informations global et improvisé. À l'heure actuelle cependant, le protocole LDAP est plus généralement utilisé au sein de grandes organisations comme des universités, des services gouvernementaux et des entreprises du secteur privé.

Le protocole LDAP est un système client/serveur. Le serveur peut utiliser diverses bases de données pour stocker un répertoire, chacune d'elles étant optimisée de façon à permettre des opérations de consultation rapides et en grande quantité. Lorsqu'un client LDAP se connecte à un serveur LDAP, il peut soit consulter un répertoire, soit y apporter des modifications. Lors de l'arrivée d'une requête, le serveur y répond localement ou la renvoie à un serveur LDAP de niveau supérieur qui aura lui la réponse. Si l'application cliente tente de changer des informations dans un répertoire LDAP, le serveur vérifie d'abord que l'utilisateur est bien autorisé à effectuer des changements et ensuite ajoute ou met à jour les informations.

Ce chapitre décrit la configuration et l'utilisation de OpenLDAP 2.0, une implémentation Open Source des protocoles LDAPv2 et LDAPv3.

15.1. Pourquoi utiliser LDAP ?

Le principal avantage du protocole LDAP réside dans la possibilité de réunir les informations concernant toute une organisation dans un lieu central. Par exemple, plutôt que de gérer des listes d'utilisateurs pour chaque groupe au sein d'une organisation, LDAP peut être utilisé comme un répertoire central accessible sur tout le réseau. De plus, puisque LDAP prend en charge les fonctions Secure Sockets Layer (SSL) et Transport Layer Security (TLS), des données confidentielles peuvent être protégées contre toute intrusion.

LDAP prend aussi en charge diverses bases de données parallèles pour y enregistrer des répertoires. Ainsi, les administrateurs disposent de la flexibilité nécessaire pour déployer la base de données la plus adaptée au type d'informations que le serveur doit disséminer. De

plus, comme LDAP comporte une interface de programmation d'application (ou API de l'anglais Application Programming Interfaces) bien définie, le nombre d'applications compatibles avec LDAP est vaste et croissant aussi bien en quantité qu'en qualité.

15.1.1. Caractéristiques d'OpenLDAP

OpenLDAP comprend un certain nombre de caractéristiques importantes parmi lesquelles figurent :

- *Prise en charge de LDAPv3* — OpenLDAP prend en charge SASL (de l'anglais Simple Authentication and Security Layer), TLS (Transport Layer Security) et SSL (Secure Sockets Layer) entre autres améliorations. De nombreux changements apportés au protocole depuis LDAPv2 visent à augmenter la sécurité de LDAP.
- *Prise en charge de IPv6* — OpenLDAP prend en charge le protocole de la prochaine génération, Internet Protocol version 6.
- *LDAP sur IPC* — OpenLDAP peut communiquer au sein d'un système en utilisant IPC (de l'anglais interprocess communication). Il en résulte une sécurité améliorée car il n'est plus nécessaire de communiquer à travers un réseau.
- *Mise à jour de C API* — Améliore la manière dont les programmeurs se connectent aux serveurs de répertoires LDAP et les utilisent.
- *Prise en charge de LDIFv1* — Grâce à cette prise en charge, OpenLDAP 2.0 est pleinement compatible avec la version 1 du format LDIF (ou LDAP Data Interchange Format).
- *Amélioration du serveur autonome LDAP* — À présent le serveur inclut entre autres un système de contrôle d'accès mis à jour, un pool de conversation et des outils plus performants.

15.2. Terminologie de LDAP

Toute discussion concernant LDAP nécessite une compréhension élémentaire d'un certain nombre de termes spécifiques à LDAP :

- *entrée* — Correspond à une seule unité dans un répertoire LDAP. Chaque entrée (ou entry) est identifiée par son *Nom distinctif ou DN* (de l'anglais Distinguished Name) unique.
- *attributs* — Des attributs (ou attributes en anglais) sont des éléments d'information directement associés à une entrée. Par exemple, une organisation pourrait être représentée en tant qu'une entrée LDAP. Parmi les attributs associés à l'organisation, on pourrait avoir son numéro de fax, son adresse, etc. Des personnes peuvent également être représentées par des entrées dans un répertoire LDAP, avec par exemple des attributs courants tels que le numéro de téléphone de la personne en question et son adresse électronique.

  Certains attributs sont obligatoires, tandis que d'autres sont facultatifs. Une définition de classe d'objets (*objectclass*) détermine les attributs spécifiques qui sont obligatoires pour chaque entrée. Des définitions de classes d'objets figurent dans différents fichiers schéma contenus dans le répertoire `/etc/openldap/schema/`.

- *LDIF* — Le format d'échange de données *LDAP Data Interchange Format* (LDIF) est un format de texte ASCII pour les entrées LDAP. Les fichiers qui échangent des données avec des serveurs LDAP doivent avoir le format LDIF. Une entrée LDIF ressemble à l'extrait ci-dessous :

```
[<id>]
dn: <distinguished name>
<attrtype>: <attrvalue>
<attrtype>: <attrvalue>
<attrtype>: <attrvalue>
```

- Toute entrée peut contenir autant de paires `<attrtype>: <attrvalue>` quil n'est nécessaire. Une ligne blanche indique que l'entrée est terminée.

Attention

Toutes les paires `<attrtype>` et `<attrvalue>` *doivent* être définies dans un fichier de schéma correspondant pour pouvoir utiliser ces informations.

Toute valeur figurant entre un `<` et un `>` représente une variable que vous pouvez paramétrer lorsqu'une nouvelle entrée LDAP est créée. Toutefois, cette règle ne s'applique pas à `<id>`. Cet élément `<id>` représente un nombre paramétré par l'application utilisée pour modifier l'entrée.

LDAP fournit un ensemble d'outils.

1. un protocole permettant d'accéder à l'information contenue dans l'annuaire,
2. un modèle d'information définissant l'organisation et le type des données contenues dans l'annuaire,
3. un modèle de nommage définissant comment l'information est organisée et référencée
4. un modèle fonctionnel qui définit comment accéder à l'information,
5. un modèle de sécurité qui définit comment accéder aux données et comment celles-ci sont protégées. OpenLDAP est souvent configuré avec SASL (Simple Authentication and Security Layer), qui permet les transactions cryptées avec les protocoles fonctionnant en mode connecté.
6. un modèle de duplication qui définit comment la base est répartie entre serveurs,
7. des APIs pour développer des applications clientes,
8. LDIF, (Ldap Data Interchange Format) un format d'échange de données.

15.2.1 Le protocole

Un protocole d'accès aux données, qui décrit comment ajouter, modifier, supprimer des données dans la base de donnée, quels protocoles de chiffrement (kerberos, ssl...), et quels mécanismes d'authentification sont utilisés. Ce protocole est utilisé dans la relation client/serveur, mais également entre serveurs (serveur/serveur) car une base de données LDAP peut être répartie.

15.2.2 Le modèle de données

15.2.2.1 Le Directory Information Tree

LDAP fournit un modèle d'organisation des données. Ces données sont organisées sous forme hiérarchique. L'arbre est nommé Directory Information Tree (DIT). Le sommet (racine), contient le "suffixe". Chaque noeud représente une "entrée" ou "Directory Entry Service" (DSE). Les données sont stockées sur un format de base de données hiérarchique de type "dbm". Ce format est différent des bases de données relationnelles, conçues pour supporter de multiples mises à jour. DBM est conçu pour supporter peu de mises à jour, mais de nombreuses consultations.

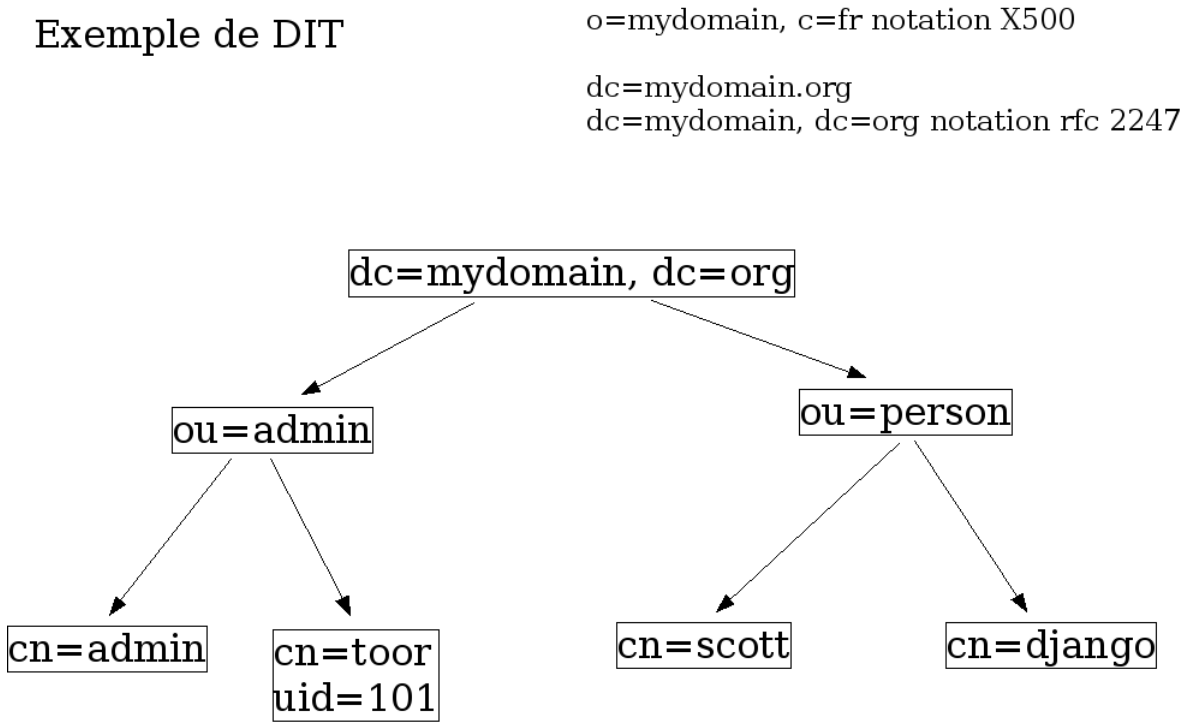


Figure 15.1. LDAP : le DIT Directory Information Tree

15.2.2.2 Classes d'objets, objets, attributs et schéma

Une entrée (DSE) dnas le DIT correspond à un objet abstrait (organisation, ressource) ou concret (personne, équipement...). Les objets possèdent une description dans une "classe d'objet". Une classe d'objet donne une représentation modélisée des objets qu'elle représente en caractérisant tous les attributs des objets.

Certaines classes dobjet ont fait l'objet d'une normalisation et sont réutilisables. Elles sont définies par un nom, un OID (Object Identifier), la liste des attributs (facultatifs ou obligatoires), et, pour chaque attribut, un type. Le type est lié à la nature (essentiellement texte ou binaire) des attributs utilisés.

Une classe d'objet est définie par un nom, un OID (Object IDentifier), la liste des attributs (facultatifs et obligatoires), un type. Le type est lié à la nature des attributs utilisés.

Chaque objet est composé d'attributs en fonction des types d'attributs décrits dans les classes d'objets. Un attribut est généralement un couple clé/valeur, mais peut être caractérisé par un nom, un OID, s'il est mono ou multi-évalué, un indicateur d'usage (facultatif/obligatoire), un format (voir par exemple pour les images).

Les OID sont normalisés par la RFC2256 et sont issus d'un schéma X500. Les OID sont tenus à jour par l'IANA Internet Assigned Numbers Authority. Un OID est une séquence de chiffres séparés par un "." point (Exemple 1.2.3.4) qui permet d'identifier de façon unique un élément du schéma LDAP.

Exemple de la la classe inetOrgPerson (dépend de organizationalPerson)

```
# The inetOrgPerson represents people who are associated with an
# organization in some way.  It is a structural class and is derived
# from the organizationalPerson which is defined in X.521 [X521].
Objectclass     ( 2.16.840.1.113730.3.2.2  <-------------- OID
  NAME 'inetOrgPerson'
     DESC 'RFC2798: Internet Organizational Person'
     SUP organizationalPerson
       STRUCTURAL
       MAY (
               audio $ businessCategory $ carLicense $ departmentNumber $
               displayName $ employeeNumber $ employeeType $ givenName $
               homePhone $ homePostalAddress $ initials $ jpegPhoto $
               labeledURI $ mail $ manager $ mobile $ o $ pager $
               photo $ roomNumber $ secretary $ uid $ userCertificate $
               x500uniqueIdentifier $ preferredLanguage $
               userSMIMECertificate $ userPKCS12 )
        )

et l'attibut employeeType
# employeeType
# Used to identify the employer to employee relationship.  Typical values
# used will be "Contractor", "Employee", "Intern", "Temp", "External", and
# "Unknown" but any value may be used.
Attributetype ( 2.16.840.1.113730.3.1.4
        NAME 'employeeType'
        DESC 'RFC2798: type of employment for a person'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
Signification des mots réservés :

DESC :  Description
SUP :           Objet parent
MUST :  Attributs requis
MAY :   Attributs possibles
```

Chaque objet de l'annuaire qui correspond à une classe d'objet est décrit par des valeurs en fonction des attributs qui décrivent la classe d'objets. Un attribut est généralement un couple clé/valeur, mais peut être caractérisé par un nom, un OID, indique s'il est mono ou multi-évalué, donne un indicateur d'usage (facultatif/obligatoire), impose un format (par exemple base64 pour les images, utf-8 pour les données).

Les objets sont rattachés obligatoirement à au moins une classe d'objet. Une classe d'objet caractérise ou modélise les objets qui lui seront rattachés. On en déduit qu'un objet ne pourra

pas avoir d'attribut non déclaré dans la classe d'objet. Par contre dans la classe d'objet, un attribut pourra être soit optionnel, soit obligatoire. L'administrateur a déjà des classes d'objets prédéfinies, il a la possibilité d'en définir d'autres.

Chaque objet hérite des attributs des objets dont il hérite. Quand on décrit un objet "fils", on doit décrire tous les liens de parenté avec l'attribut "ObjetClass". Les objets forment une hiérarchie, avec, au sommet, l'objet "top". Exemple pour enrichir l'objet "person" des attributs "technicalPerson".

```
objectClass: top
objectClass: myOrganization
objectClass: person
objectClass: technicalPerson
```

L'ensemble de la description des classes d'objet et des types d'attributs définissent le "schéma".

Une entrée peut appartenir à plusieurs classes d'objets. Les attributs de l'objets sont la réunion des attributs de toutes les classes d'objets :

```
Entry Type        Required Attributes   Optional Attributes
person            commonName (cn)       mail
                  surName (sn)          mobile
                  objectClass           ...

OrganizationUnit  ou                    description
                  objectClass           localisation
                                        ...
Organization      o                     description
                  onjectClass           ...
```

Les OID sont normalisés par la RFC2256 et sont issus d'un schéma X55. Les OID sont tenus à jour par l'IANA Internet Assigned Numbers Authority. Un OID est une séquence de chiffres séparés par un "." point. Exemple 1.2.3.4

Le "dn", ou Distinguished Name, est le chemin absolu de l'entrée dans le DIT à partir de la racine. Par exemple :

```
dc=org, dc=mydomaine, ou=person, uid=toor
```

On peut utiliser aussi un nommage "relatif" par rapport à une position courante. Dans ce cas on utilise le RDN (Relative Distinguished Name).

15.2.2.3 Le format d'échange de donnée LDIF

Le format d'échange permet l'import/export de données des bases, mais sert également pour l'ajout ou la modification. Les données sont en ASCII codées en UTF-8, sauf pour le binaire qui est codé en base64 (images par exemple).

Les fichiers au format LDIF respectent une structure de description des objets et des commandes :

```
Syntaxe générale :
    dn: <distinguished name
    objectClass: <object class
    objectClass: <object class
    ...
    <attribute type:<attribute value
    <attribute type:<attribute value
    ...

Exemple :
    dn: cn= Manon Des Sources, ou= compta, dc=mydomain, dc=org
    objectClass: person
    objectClass: organization
    cn: AN GROSSI
    sn: GROSSI
    givenName: AM
    userPassword: {sha}KDIE3AL9DK
    uid: amg
```

Les fichiers supportent également des commandes pour ajouter, modifier ou supprimer une entrée.

```
    dn: distinguished name
    changetype <identifier
    change operation identifier
    list of attributes...
    ...
    -
    change operation identifier
    list of attributes
    ...
    <identifier :
        add (ajouter une entrée,
        delete (suppression),
        modrdn (modification du RDN),
        modify (modification : add, replace, delete d'un attribut)
```

On utilise le caractère "-" pour séparer 2 instructions. Par exemple :

```
     dn: cn= Morina Fuentes, ou=admin, dc=mydomain, dc=org
     changetype: modify
     add: telephonenumber
     telephonenumber: 05 55 55 55 55
     -
     add: manager
     manager: cn= toor root, ou=admin, dc=mydomain, dc=org
```

15.2.2.4 Les méthodes d'accès

Les opérations de base sont résumées ici :

```
Opération
Search              recherche dans l'annuaire d'objets
Compare             comparaison du contenu de deux objets
Add                 ajout d'une entrée
Modify              modification du contenu d'une entrée
Delete              suppression d'un objet
Rename (Modify DN) modification du DN d'une entrée
Bind                connexion au serveur
```

```
Unbind              deconnexion
```

Les requêtes de type "search" ou "compare" reçoivent des paramètres.

```
Paramètres :
base object :   l'endroit de l'arbre où doit commencer la recherche
scope :         la profondeur de la recherche
size limit :    nombre de réponses limite
time limit :    temps maxi alloué pour la recherche
attrOnly :      renvoie ou pas la valeur des attributs en plus de leur type
search filter : le filtre de recherche
list of attributes : la liste des attributs que l'on souhaite connaître
```

Le scope (Profondeur de recherche)

Le scope définit la profondeur de la recherche dans l'arbre des données. La figure montre la portée d'une recherche ou d'une comparaison en fonction du paramètre scope.

```
search scope=base recherche uniquement dans l'entrée définie
search scope=one recherche dans l'entrée définie et le premier sous-niveau
search scope = subtree, cherche dans toute la sous-arborescence.
```

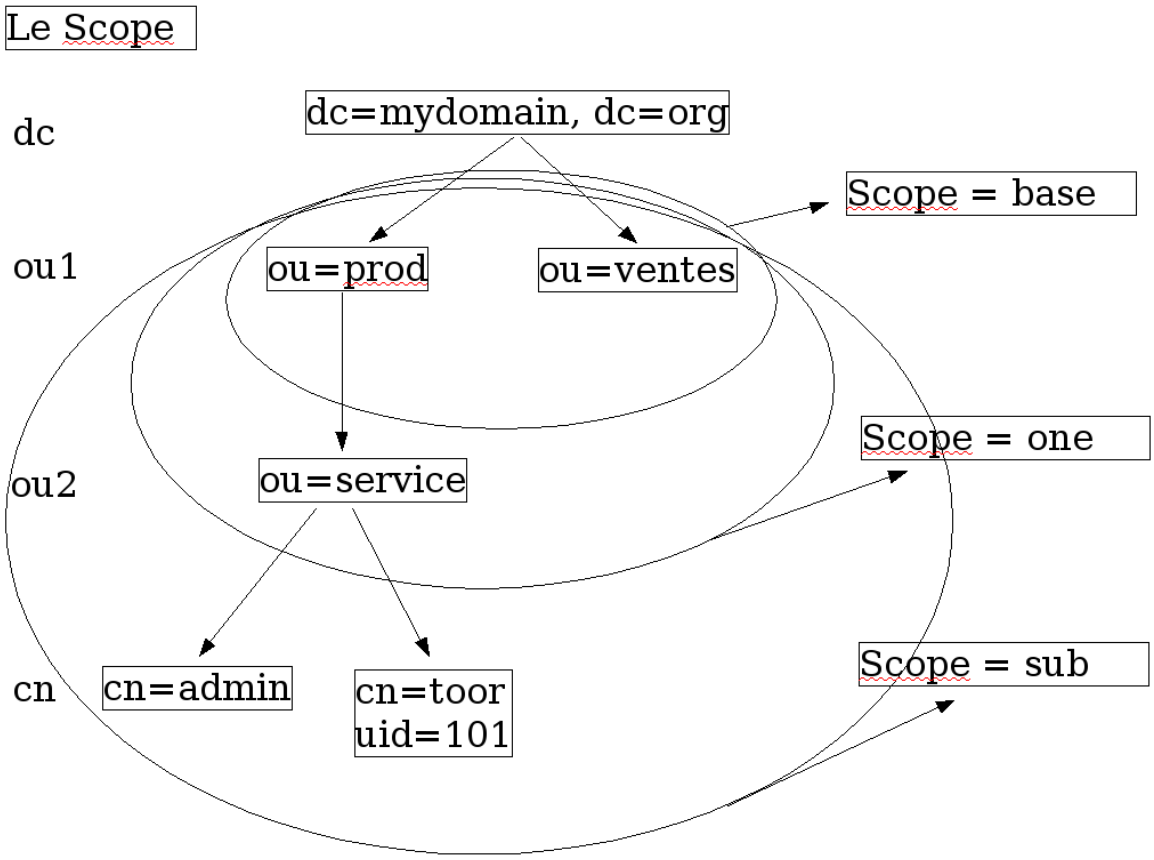


Figure 15.2. LDAP : le scope

search scope=base recherche uniquement dans l'entrée définie

search scope=onelever search, cherche dans tous les noeuds fils rattachés directement au noeud courant

search scope = subtree, cherche dans toute la sous-arborescence.

Les URL LDAP offrent aux clients web un accès aux bases de données :

```
ldap[s]://<hostname>:<port>/<base_dn>?<attributes>?<scope>?<filter>

    <base_dn>   : DN de l'entrée qui est le point de départ de la recherche
    <attributes>: les attributs que l'on veut consulter
    <scope>     : la profondeur de recherche dans le DIT à partir du
    <base_dn>   :"base" | "one" | "sub"
    <filter>    : filtre de recherche, par défaut (objectClass=*)
```

Exemples :

```
    ldap://ldap.netscape.com/ou=Sales,o=Netscape,c=US?cn,\
           tel,mail?scope=sub?(objetclass=person)
    ldap://ldap.point-libre.org/cn=Manon,ou=Contact,o=point-libre.org
```

Lecture de toutes les personnes du service vente :

```
ldap://ldap.netscape.com/ou=Sales,o=Netscape,c=US?cn,tel,mail?scope=sub?(ob
jetclass=person)
```

Lecture des objets personnes d'un annuaire :

```
ldap://localhost:389/??sub?objectclass=person
```

Recherche de Valery Febvre :

```
ldap://ldap.easter-eggs.fr/cn=Valery%20Febvre,\
       ou=Moyens%20Informatiques,dc=easter-eggs,dc=fr
```

Recherche approximative d'une personne :

```
ldap://ldap.easter-eggs.fr/o=easter-eggs,dc=fr?mail,uid,sub?(sn=Febvre)
```

15.3. Démons et utilitaires d'OpenLDAP

La suite de bibliothèques et d'outils OpenLDAP est incluse dans les paquetages suivants :

- `openldap` — Contient les bibliothèques nécessaires pour faire fonctionner le serveur OpenLDAP et les applications clientes.
- `openldap-clients` — Contient les outils de ligne de commande pour visualiser et modifier les répertoires d'un serveur LDAP.
- `openldap-servers` — Contient les serveurs et autres utilitaires nécessaires pour configurer et faire fonctionner un serveur LDAP.

Deux serveurs sont contenus dans le paquetage `openldap-servers` : le *démon autonome LDAP* (`/usr/sbin/slapd`) et le *démon autonome LDAP de réplication de mise à jour* (`/usr/sbin/slurpd`).

Le démon `slapd` est un serveur LDAP autonome, tandis que le démon `slurpd` sert à synchroniser les changements d'un serveur LDAP vers les autres serveurs LDAP du réseau. Le démon `slurpd` n'est utilisé que pour des opérations avec de multiples serveurs LDAP.

Pour effectuer des tâches administratives, le paquetage `openldap-servers` installe les utilitaires suivants dans le répertoire `/usr/sbin/` :

- `slapadd` — Ajoute des entrées d'un fichier LDIF dans un répertoire LDAP. Par exemple, la commande `/usr/sbin/slapadd -l` *`ldif-input`* lit le fichier LDIF *`ldif-input`* contenant les nouvelles entrées.

  Important

  Seul le super-utilisateur peut utiliser `/usr/sbin/slapadd`. Toutefois, le serveur de répertoires tourne en tant que l'utilisateur `ldap`. Par conséquent, le serveur de répertoires n'est pas en mesure de modifier un fichier quelconque créé par `slapadd`. Pour résoudre ce problème, après avoir utilisé `slapadd`, tapez la commande suivante :

  ```
  chown -R ldap /var/lib/ldap
  ```

- `slapcat` — Extrait des données d'un répertoire LDAP dans le format par défaut, à savoir le système *Berkeley DB de Sleepycat Software*, et les enregistre dans un fichier LDIF. Par exemple, la commande `/usr/sbin/slapcat -l` *`ldif-output`* renvoie un fichier LDIF nommé *`ldif-output`* qui contient les entrées du répertoire LDAP.
- `slapindex` — Indexe à nouveau le répertoire `slapd` sur la base du contenu actuel. Cet outil devrait être exécuté chaque fois que les options d'indexation de `/etc/openldap/slapd.conf` sont modifiées.
- `slappasswd` — Crée une valeur pour le mot de passe utilisateur crypté devant être utiliser avec `ldapmodify` ou la valeur `rootpw` dans le fichier de configuration de `slapd`, `/etc/openldap/slapd.conf`. Exécutez la commande `/usr/sbin/slappasswd` pour créer le mot de passe.

Avertissement

Vous devez arrêter `slapd` en exécutant la commande `/sbin/service lapd stop` avant d'utiliser `slapadd`, `slapcat` ou `slapindex`. Dans le cas contraire, l'intégrité du répertoire LDAP risque d'être compromise.

Pour obtenir de plus amples informations sur l'utilisation de ces utilitaires, consultez leurs pages de manuel respectives.

Le paquetage `openldap-clients` installe dans `/usr/bin/` des outils permettant d'ajouter, de modifier et de supprimer des entrées dans un répertoire LDAP. Parmi ces outils figurent :

- `ldapadd` — Ajoute des entrées dans un répertoire LDAP en acceptant la saisie d'entrées par le biais d'un fichier ou par une saisie standard ; `ldapadd` est en fait un lien dur vers la commande `ldapmodify -a`.
- `ldapdelete` — Supprime des entrées dans un répertoire LDAP en acceptant la saisie de l'utilisateur à une invite du shell ou par le biais d'un fichier.
- `ldapmodify` — Modifie les entrées dans un répertoire LDAP, acceptant leur saisie par un fichier ou par une saisie standard.
- `ldappasswd` — Définit le mot de passe pour un utilisateur LDAP.
- `ldapsearch` — Recherche des entrées dans un répertoire LDAP en utilisant une invite du shell.

À l'exception de la commande `ldapsearch`, chacun de ces utilitaires est plus facilement utilisé en référencant un fichier contenant les changements à effectuer plutôt qu'en tapant une commande pour chaque entrée devant être modifiée dans le répertoire LDAP. Le format d'un tel fichier est expliqué dans les pages de manuel de chaque utilitaire.

Exemples :

```
ldapsearch  -x -h localhost -b "dc=mydomain,dc=fr" "objectclass=*"
```

Recherche de tous les objets sur l'annuaire de la machine locale, à partir de la racine. (-b indique à partir de quel niveau la recherche doit être exécutée). L'option « -x » indique de ne pas utiliser la méthode d'authentification SASL si elle n'est pas activée.

```
ldapdelete 'cn=Jean Colombani,cn=mydomain,cn=fr'
```

Suppression d'une entrée dans l'annuaire

```
ldapadd -f /tmp/unFichierAuFormatLDIF
```

Ajout dans l'annuaire à partir d'un fichier contenant des données au format LDIF.

Note: Pour éviter d'avoir à préciser à chaque fois certains paramètres (machine, port, annuaire...) il est possible de configurer le fichier de configuration « ldap.conf » qui sera utilisé par les applications clientes.

15.3.1. Filtre de recherche

15.3.1.1. Présentation générale

Un filtre LDAP est comparable à une requête SQL. C'est une chaîne de caractères destinée à être exécutée pour récupérer des entrées d'un annuaire LDAP. Plus précisément elle ne définit que la partie *WHERE* d'une requête SQL: un filtre définit sur quels objets et sur quels attributs doit se faire la recherche.

Un filtre LDAP est donc constitué d'un ensemble d'opérations, portant sur des attributs, combinées avec les opérateurs booléens classiques: *ET*, *OU* et *NON*.

Une opération élémentaire de recherche s'écrit sous la forme : *attribut OPERATEUR valeur*
La forme générale d'un filtre est une combinaison : *(operator(search operation)(search operation)...))*.

La syntaxe complète des filtres LDAP est décrite dans le document [rfc2254], qui fait partie de l'ensemble des RFC définissant la norme LDAP.

15.3.1.2. Les opérations élémentaires

Une opération élémentaire est composée d'un attribut, d'un opérateur de comparaison et d'une valeur. Pour effectuer des filtres sur le type des objets, sur leur classe, il suffit d'utiliser leur objectclass comme un attribut. Au besoin, il est possible d'utiliser l'OID de l'attribut plutôt que son nom.

Les opérateurs de recherche sont les suivants :

Égalité	: =
Approximation	~=
Supérieur ou égal	>=
Inférieur ou égal	<=

S'il n'existe pas d'opérateur : "différent de", "strictement inférieur", ou "strictement supérieur", il est possible de les obtenir à l'aide des opérateurs autorisés, en utilisant un opérateur booléen "NON".

La valeur accepte le caractère '*' afin de permet des recherches sur des parties de chaînes. Ce même caractère, seul, permet de tester la présence d'un attribut. Ce caractère n'est valide qu'avec l'opération d'égalité.

Une opération doit obligatoirement se trouver entre deux parenthèses. Les valeurs qui sont dans la partie droite d'une opération élémentaire ne sont pas entre quote, mais certains caractères doivent être échappés :

Le caractère	**Sa valeur ASCII**	**Le caractère dans un filtre**
*	0x2a	\2a
(	0x28	\28
)	0x29	\29
\	0x5c	\5c
NULL (caractère vide)	0x00	\00

Les opérateurs booléens sont les suivants :

L'opérateur NON	!
L'opérateur ET	&
L'opérateur OU	|

Un opérateur booléen s'applique à toutes les opérations qui suivent jusqu'à l'opérateur suivant.

15.3.1.3. Exemples de filtres simples

Toutes les personnes ayant leur numéro de téléphone renseigné dans la base :

```
(&(objectclass=person)(telephoneNumber=*))
```

Toutes les personnes dont le nom commence par 'A' et n'habitant pas Paris :

```
(&(objectclass=person)(cn=A*)(!(l=Paris)))
```

Toutes les personnes dont le nom ressemble à Febvre (Faivre, Fèvre, Lefebvre, ...) :

```
(&(objectclass=person)(cn~=febvre))
(&(objectclass=person)(cn=*f*vre))
```

15.3.1.4. Les filtres étendus

En plus des attributs constituant une entrée d'un annuaire, il est possible, grâce aux filtres étendus, de considérer les éléments du DN comme faisant partie aussi de l'entrée elle même, lors de la recherche. La syntaxe est la suivante :

```
attribut:dn:=value
```

Par exemple le filtre *(ou:dn:=users)* récupérera non seulement toutes les entrées qui ont un attribut ou qui a pour valeur *users*, mais aussi toutes les entrées dont le DN contient un attribut ou avec la valeur *users*.

L'autre possibilité offerte par les filtres étendus est de modifier la règle de comparaison sur un attribut. La syntaxe est la suivante :

```
attribut:oid-matching-rule:=value
```

Par exemple, si un attribut a une règle de comparaison par défaut qui est insensible à la case, mais que l'on veut faire une recherche avec une valeur précise, qui tienne compte de la case, il faut modifier la règle de comparaison par défaut.

a. Changement de règle de comparaison dans un filtre

Dans cet exemple, on rend l'opération d'égalité sensible à la case

```
(cn:2.5.13.5:=Mickey)
```

Les filtres étendus permettent aussi de spécifier une règle de comparaison, sans spécifier d'attribut. La recherche s'effectue alors sur tous les attributs qui acceptent cette règle. On peut aussi étendre cette recherche au DN. Les syntaxes sont les suivantes:

```
:oid-matching-rule:=value
:dn:oid-matching-rule:=value
```

Tous les filtres ne marchent qu'avec l'opérateur d'égalité, et celui-ci s'écrit :=, lieu de = dans le cas de filtres simples.

Il n'est pas permis non plus d'utiliser le caractère * pour faire des recherches sur des sous chaînes avec les filtres étendus.

15.3.2. NSS, PAM et LDAP

Outre les paquetages OpenLDAP, Red Hat Enterprise Linux comprend un paquetage nommé `nss_ldap` qui améliore la capacité de LDAP à s'intégrer aussi bien dans un environnement Linux que dans tout autre environnement UNIX.

Le paquetage nss_ldap fournit les modules suivants :

- /lib/libnss_ldap-<*glibc-version*>.so
- /lib/security/pam_ldap.so

Le paquetage nss_ldap fournit les modules suivants pour les architectures Itanium ou AMD64 :

- /lib64/libnss_ldap-<*glibc-version*>.so
- /lib64/security/pam_ldap.so

Le module libnss_ldap-<*glibc-version*>.so permet aux applications de rechercher des utilisateurs, des groupes, des hôtes et d'autres informations en utilisant un répertoire LDAP via l'interface *Nameservice Switch* (ou NSS) de glibc (remplacez <*glibc-version*> par la version de libnss_ldap utilisée). NSS permet aux applications de s'authentifier en utilisant LDAP de concert avec le service de noms NIS et les fichiers plats d'authentification.

Le module pam_ldap permet aux applications fonctionnant avec PAM d'authentifier les utilisateurs en utilisant les informations stockées dans un répertoire LDAP. Les applications fonctionnant avec PAM comprennent la connexion console, les serveurs de messagerie POP et IMAP ainsi que Samba. En déployant un serveur LDAP sur votre réseau, toutes ces applications peuvent, pour leur authentification, utiliser la même combinaison identifiant d'utilisateur/mot de passe, ce qui simplifie considérablement l'administration.

15.3.3. PHP4, LDAP et le Serveur HTTP Apache

Red Hat Enterprise Linux comprend un paquetage avec un module LDAP pour le langage de scripts PHP côté serveur.

Le paquetage php-ldap ajoute la prise en charge LDAP au langage de script PHP4 avec intégration HTML grâce au module /usr/lib/php4/ldap.so. Ce module permet aux scripts PHP4 d'accéder aux informations stockées dans un répertoire LDAP.

Red Hat Enterprise Linux est vendu avec le module mod_authz_ldap pour le Serveur HTTP Apache. Ce module utilise la forme courte du nom distinct d'un sujet et le fournisseur du certificat SSL client afin de déterminer le nom distinct de l'utilisateur au sein d'un répertoire LDAP. Il est également capable d'autoriser des utilisateurs selon les attributs de l'entrée du répertoire LDAP de cet utilisateur, en déterminant l'accès aux ressources sur la base des privilèges dont disposent l'utilisateur et du groupe sur ces dernières et en refusant l'accès aux utilisateurs dont les mots de passe ont expiré. Le module mod_ssl est nécessaire lorsque le module mod_authz_ldap est utilisé.

Important

Le module mod_authz_ldap n'authentifie pas un utilisateur auprès d'un répertoire LDAP à l'aide d'un mot de passe crypté. Cette fonctionnalité est fournie par le module expérimental nommée mod_auth_ldap qui n'est pas inclus dans Red Hat Enterprise Linux. Pour obtenir de plus amples informations sur le statut de ce module, reportez-vous au site Web de l'organisation Apache Software Foundation à l'adresse suivante : http://www.apache.org/.

15.3.4. Applications client LDAP

Il existe des clients LDAP graphiques prennant en charge la création et la modification de répertoires, mais ces applications ne sont *pas* incluses dans Red Hat Enterprise Linux. C'est le cas de l'application **LDAP Browser/Editor** (Navigateur/éditeur LDAP) — Cet outil basé sur Java est disponible en ligne à l'adresse suivante : http://www.iit.edu/~gawojar/ldap/.

La plupart des autres clients LDAP accèdent aux répertoires en lecture-seule et les utilisent pour référencer, et non pas modifier, les informations relatives à toute l'entreprise. Parmi ces applications figurent Sendmail, **Mozilla**, **Gnome Meeting** et **Evolution**.

15.4. Fichiers de configuration d'OpenLDAP

Les fichiers de configuration d'OpenLDAP sont installés dans le répertoire `/etc/openldap/`. Ci-dessous figure une brève liste des répertoires et fichiers les plus importants :

- `/etc/openldap/ldap.conf` — Ce fichier est le fichier de configuration pour toutes les applications *clientes* qui utilisent les bibliothèques OpenLDAP telles que `ldapsearch`, `ldapadd`, Sendmail, **Evolution** et **Gnome Meeting**.
- `/etc/openldap/slapd.conf` — Ce fichier est le fichier de configuration du démon `slapd`.
- Le répertoire `/etc/openldap/schema/` — Ce sous-répertoire contient le schéma utilisé par le démon `slapd`.

Remarque

Si le paquetage `nss_ldap` est installé, il crée un fichier nommé `/etc/ldap.conf`. Ce fichier est utilisé par les modules PAM et NSS fournis par le paquetage `nss_ldap`.

15.5. Répertoire `/etc/openldap/schema/`

Le répertoire `/etc/openldap/schema/` contient des définitions LDAP précédemment placées dans les fichiers `slapd.at.conf` et `slapd.oc.conf`. Le fichier `/etc/openldap/schema/redhat/` contient des schémas personnalisés distribués par Red Hat pour Red Hat Enterprise Linux.

Toutes les *définitions de syntaxe d'attribut* et *définitions de la classe d'objet* sont maintenant placées dans des fichiers schéma différents. Ces derniers sont référencés dans `/etc/openldap/slapd.conf` en utilisant les lignes `include`, comme dans l'exemple ci-dessous :

```
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/nis.schema
include         /etc/openldap/schema/rfc822-MailMember.schema
include         /etc/openldap/schema/redhat/autofs.schema
```

Attention

Ne modifiez aucun élément schéma défini dans les fichiers schéma installés par OpenLDAP.

Ceci étant, vous pouvez étendre le schéma utilisé par OpenLDAP afin de prendre en charge d'autres types d'attributs et classes d'objets en utilisant comme guide les fichiers schéma par défaut. Pour ce faire, créez un fichier `local.schema` dans le répertoire `/etc/openldap/schema`. Référencez ce nouveau schéma dans `slapd.conf` en ajoutant la ligne suivante en dessous de vos lignes schéma `include` par défaut :

```
include         /etc/openldap/schema/local.schema
```

Ensuite, définissez vos nouveaux types d'attributs et classes d'objets dans le fichier `local.schema`. Beaucoup d'organisations utilisent les types d'attributs et classes d'objet existants dans les fichiers schéma installés par défaut et ajoutent de nouvelles classes d'objets dans le fichier `local.schema`.

L'extension d'un schéma pour qu'il corresponde à des besoins spécialisés est une tâche complexe qui dépasse la portée du présent chapitre. Pour obtenir de plus amples d'informations sur le sujet, consultez l'adresse suivante : http://www.openldap.org/doc/admin/schema.html.

15.6. Aperçu de la configuration d'OpenLDAP

Cette section fournit une présentation rapide des opérations à accomplir pour installer et configurer un annuaire OpenLDAP (aussi appelé répertoire). Pour obtenir de plus amples informations sur le sujet, reportez-vous aux URL suivantes :

- http://www.openldap.org/doc/admin/quickstart.html — Le guide rapide pour commencer (*Quick-Start Guide*) sur le site Web d'OpenLDAP.
- http://www.redhat.com/mirrors/LDP/HOWTO/LDAP-HOWTO.html — Le document *LDAP Linux HOWTO* du Projet de documentation Linux (Linux Documentation Project) en miroir sur le site de Red Hat.

Ci-dessous figurent les étapes de base pour créer un serveur LDAP :

1. Installez les RPM d'`openldap`, `openldap-servers` et `openldap-clients`.
2. Éditez le fichier `/etc/openldap/slapd.conf` afin de spécifier le domaine et le serveur LDAP.
3. Lancez `slapd` à l'aide de la commande :

   ```
   /sbin/service ldap start
   ```

4. Après avoir configuré LDAP, utilisez `chkconfig`, `ntsysv` ou l'**Outil de configuration des services** pour configurer LDAP de façon à le lancer au démarrage. Pour de plus amples informations sur la configuration des services, consultez le chapitre intitulé *Contrôle de l'accès aux services* du *Guide d'administration système de Red Hat Enterprise Linux*.
5. Ajoutez des entrées à un répertoire LDAP à l'aide de `ldapadd`.
6. Utilisez `ldapsearch` afin de vérifier si `slapd` accède correctement aux informations.

7. À ce stade, le répertoire LDAP devrait fonctionner correctement et peut donc être configuré avec des applications compatibles avec LDAP.

15.6.1. Édition de `/etc/openldap/slapd.conf`

Afin d'utiliser le serveur LDAP `slapd`, modifiez son fichier de configuration, `/etc/openldap/slapd.conf` de façon à spécifier le domaine et le serveur corrects.

La ligne de `suffix` nomme le domaine pour lequel le serveur LDAP fournira les informations et devrait être changée comme suit :

```
suffix          "dc=your-domain,dc=com"
```

de façon à refléter votre nom de domaine. Par exemple :

```
suffix          "dc=example,dc=com"
```

L'entrée `rootdn` est le *Nom distinct* (ou *DN* selon l'acronyme anglais) pour un utilisateur dont l'activité n'est pas limitée par les paramètres de contrôle d'accès ou de limites administratives définis pour toute opération sur le répertoire LDAP. L'utilisateur `rootdn` peut être considéré comme le super-utilisateur pour le répertoire LDAP. Dans le fichier de configuration, modifiez la ligne `rootdn` pour changer la valeur par défaut comme dans l'exemple suivant :

```
rootdn          "cn=root,dc=example,dc=com"
```

Si vous avez l'intention de peupler le répertoire LDAP sur le réseau, modifiez la ligne `rootpw` — en remplaçant la valeur par défaut par une chaîne de mot de passe cryptée. Afin de créer une chaîne de mots de passe cryptée, tapez la commande suivante :

```
slappasswd
```

Lorsque le système vous le emandera, saisissez et confirmez un mot de passe. Le programme affiche alors à l'invite du shell, le mot de passe crypté résulant de la commande.

Ensuite, copiez le mot de passe crypté que vous venez de créer dans `/etc/openldap/slapd.conf` sur une des lignes `rootpw` et supprimez le signe dièse (`#`).

Une fois cette modification apportée, la ligne devrait ressembler à l'exemple reproduit ci-dessous :

```
rootpw {SSHA}vv2y+i6V6esazrIv70xSSnNAJE18bb2u
```

Avertissement

Les mots de passe LDAP, y compris la directive `rootpw` spécifiée dans `/etc/openldap/slapd.conf`, sont envoyés sur le réseau en *texte clair*, à moins que le cryptage TLS ne soit activé.

Pour permettre le cryptage TLS, passez en revue les commentaires figurant dans

`/etc/openldap/slapd.conf` et consultez la page de manuel de `slapd.conf`.

Pour une sécurité accrue, la directive `rootpw` devrait être désactivée après avoir peuplé le répertoire LDAP. Pour ce faire, ajoutez un signe dièse devant cette directive (#).

Si vous utilisez l'outil de ligne de commande `/usr/sbin/slapadd` localement pour peupler le répertoire, il n'est pas nécessaire d'utiliser la directive `rootpw`.

Important

Seul le super-utilisateur peut utiliser `/usr/sbin/slapadd`. Toutefois, le serveur de répertoires tourne en tant que l'utilisateur `ldap`. Par conséquent, le serveur de répertoires ne peut modifier aucun fichier créé par `slapadd`. Pour résoudre ce problème, après avoir utilisé `slapadd` tapez la commande ci-dessous :

```
chown -R ldap /var/lib/ldap
```

Remarque

Les autorisations d'accès nécessitent une remarque.
L'accès par défaut est "read", mais il est possible d'affiner. Par exemple avec des règles d'écriture comme:

```
access to <what> [ by <who> <none | compare | search | read | write>]
# Donne un accés en écriture pour le manager du domaine
access to * by dn="cn=Manager,dc=mydomain,dc=fr" write
# Donne un accés en lecture à tout le monde sur la base
access to * by * read
# Donne un accés en écriture sur un attribut pour le manager
#                en lecture pour les autres.
access to attr=uid
                  by dn="manager,dc=mydomain,dc=fr" write
                  by * none
```

Le nombre d'options est très important, utilisez la commande "**man slapd.conf**".

15.7. Configuration d'un système pour l'authentification avec OpenLDAP

Cette section donne un bref aperçu de la manière de configurer l'authentification des utilisateurs à l'aide d'OpenLDAP. À moins d'être un expert d'OpenLDAP, vous aurez probablement besoin de plus de documentation que vous n'en trouverez ici.

Installez les paquetages LDAP nécessaires

Commencez par vérifier que les paquetages appropriés sont installés aussi biensur le serveur LDAP et que sur les machines LDAP clientes. Le serveur LDAP nécessite le paquetage `openldap-servers`.

Les paquetages `openldap`, `openldap-clients` et `nss_ldap` doivent être installés sur tous les ordinateurs clients LDAP.

Éditez des fichiers de configuration

- Sur le serveur LDAP, éditez le fichier `/etc/openldap/slapd.conf` pour vous assurer qu'il correspond bien aux éléments spécifiques de votre organisation.
- Sur les ordinateurs clients, `/etc/ldap.conf` et `/etc/openldap/ldap.conf` doivent contenir le bon serveur et les bonnes informations de la base de recherche pour l'organisation.

  Pour ce faire, lancez l'**Outil de configuration d'authentification** graphique (`system-config-authentication`) et sélectionnez **Activer le support LDAP** sous l'onglet **Informations utilisateur**.

  Vous pouvez également modifier ces fichiers manuellement.
- Sur les ordinateurs clients, le fichier `/etc/nsswitch.conf` doit être édité afin de pouvoir utiliser LDAP.

  Pour ce faire, lancez l'**Outil de configuration d'authentification** (`system-config-authentication`) et sélectionnez **Activer le support LDAP** sous l'onglet **Informations utilisateur**.

  Si vous éditez `/etc/nsswitch.conf` manuellement, ajoutez `ldap` aux lignes appropriées.

  Comme par exemple :

```
passwd: files ldap
shadow: files ldap
group: files ldap
```

15.7.1. PAM et LDAP

Pour faire en sorte que des applications compatibles avec PAM standards utilisent LDAP pour l'authentification, exécutez l'**Outil de configuration d'authentification** (`system-config-authentication`) et sélectionnez **Activer le support LDAP** sous l'onglet **Authentification**. Pour obtenir davantage d'informations sur la configuration de PAM, consultez les pages de manuel de PAM.

15.7.2. Migration de vos anciennes informations d'authentification vers le format LDAP

Le répertoire `/usr/share/openldap/migration/` contient un ensemble de scripts shell et Perl pour la migration des informations d'authentification vers un format LDAP.

Remarque

Perl doit être installé sur votre système pour que vous puissiez utiliser ces scripts.

Tout d'abord, modifiez le fichier `migrate_common.ph` de manière à ce qu'il reflète le domaine approprié. La valeur par défaut du domaine DNS par défaut devrait être modifiée de manière semblable à l'extrait suivant :

```
$DEFAULT_MAIL_DOMAIN = "example";
```

La base par défaut devrait également être changée, pour ressembler à ceci :

```
$DEFAULT_BASE =
"dc=example,dc=com";
```

Le travail de migration d'une base de données d'utilisateurs vers un format lisible par LDAP incombe à un groupe de scripts de migration installés dans le même répertoire. À l'aide du Tableau 15-1, déterminez le script à utiliser pour la migration de base de données d'utilisateurs.

Exécutez le script approprié en fonction du service de noms existant.

Les fichiers `README` et `migration-tools.txt` du répertoire `/usr/share/openldap/migration/` fournissent davantage de renseignements sur la migration d'informations.

Service de noms existant	LDAP fonctionne-t-il ?	Script à utiliser
Fichiers `/etc`	oui	`migrate_all_online.sh`
Fichiers `/etc`	non	`migrate_all_offline.sh`
NetInfo	oui	`migrate_all_netinfo_online.sh`
NetInfo	non	`migrate_all_netinfo_offline.sh`
NIS (YP)	oui	`migrate_all_nis_online.sh`
NIS (YP)	non	`migrate_all_nis_offline.sh`

Tableau 15-1. Scripts de migration LDAP

15.8. Ressources supplémentaires

Les ressources suivantes fournissent des informations supplémentaires sur LDAP. Il est fortement recommandé de les consulter, en particulier le site Web d'OpenLDAP et le HOWTO LDAP, avant de configurer LDAP sur un ou plusieurs systèmes.

15.8.1. Documentation installée

- `/usr/share/docs/openldap-<numéro-version>` — Contient un document `README` général ainsi que des informations diverses.
- Pages de manuel de LDAP — Il existe un nombre de pages de manuel pour les diverses applications et fichiers de configuration utilisés avec LDAP. La liste suivante contient certaines des pages de manuel les plus importantes.

Applications client

- `man ldapadd` — Décrit comment ajouter des entrées à un répertoire LDAP.
- `man ldapdelete` — Décrit comment supprimer des entrées dans un répertoire LDAP.
- `man ldapmodify` — Décrit comment modifier des entrées dans un répertoire LDAP.
- `man ldapsearch` — Décrit comment rechercher des entrées dans un répertoire LDAP.
- `man ldappasswd` — Décrit comment définir ou modifier le mot de passe d'un utilisateur de LDAP.

Applications serveur

- `man slapd` — Décrit les options de ligne de commande pour le serveur LDAP.
- `man slurpd` — Décrit les options de ligne de commande pour le serveur de réplication LDAP.

Applications administratives

- `man slapadd` — Décrit les options de ligne de commande utilisées pour ajouter des entrées dans une base de données `slapd`.
- `man slapcat` — Décrit les options de ligne de commande utilisées pour générer un fichier LDIF à partir d'une base de données `slapd`.
- `man slapindex` — Décrit les options de ligne de commande utilisées pour régénérer un index selon le contenu d'une base de données `slapd`.
- `man slappasswd` — Décrit les options de ligne de commande utilisées pour générer des mots de passe d'utilisateur pour les répertoires LDAP.

Fichiers de configuration

- `man ldap.conf` — Décrit le format et les options disponibles au sein du fichier de configuration pour les clients LDAP.
- `man slapd.conf` — Décrit le format et les options disponibles au sein du fichier de configuration référencé aussi bien par les applications serveur de LDAP (`slapd` et `slurpd`) que par les outils administratifs de LDAP (`slapadd`, `slapcat` et `slapindex`).

15.8.2. Sites Web utiles

- http://www.openldap.org/ — Page d'accueil du projet de l'organisation OpenLDAP. Ce site Web contient de nombreuses informations sur la configuration d'OpenLDAP ainsi que sur la feuille de route future et sur les changements apportés à la version.
- http://www.redhat.com/mirrors/LDP/HOWTO/LDAP-HOWTO.html — Document HOWTO complet, pertinent et mis à jour sur LDAP.
- http://www.padl.com/ — Développeurs de `nss_ldap` et `pam_ldap` entre autres outils LDAP utiles.
- http://www.kingsmountain.com/ldapRoadmap.shtml — Feuille de route LDAP de Jeff Hodges contenant des liens vers différents Forums aux questions (FAQ) et des informations importantes concernant le protocole LDAP.
- http://www.newarchitectmag.com/archives/2000/05/wilcox/ — Examen utile de la gestion des groupes sous LDAP.
- http://www.ldapman.org/articles/ — Articles offrant une bonne introduction à LDAP, y compris des méthodes pour concevoir une arborescence de répertoires et personnaliser des structures de répertoire.

15.8.3. Livres sur le sujet

- *OpenLDAP by Example* de John Terpstra et Benjamin Coles ; Prentice Hall.
- *Implementing LDAP* de Mark Wilcox ; Wrox Press, Inc.
- *Understanding and Deploying LDAP Directory Services* de Tim Howes et al. ; Macmillan Technical Publishing

TP Installation/Configuration d'un annuaire LDAP sur serveur GNU/Linux

Objectifs :

– Installer un annuaire LDAP sur un PC serveur GNU/Linux (Mandriva).

– Visiter les principaux fichiers de configuration utiles à LDAP.

– Utiliser l'annuaire LDAP depuis un poste client GNU/Linux, ou Windows.

1 Installation du serveur LDAP et des utilitaires (Mandriva)

a - Installation

- Utiliser rpm –qa pour trouver les paquets *rpm* liés au service ldap.
- Installer le paquet serveur ldapservers, le paquet client openldapclients , les paquets nss_ldap et pam_ldap avec la commande rpm.
- Visualiser les fichiers des paquets installés.

2 Configuration d'un serveur LDAP

On va configurer un serveur LDAP, pour une entité fictive dont le DNS serait tpldap.tn. Le but est de modifier les champs database, suffix, rootdn et rootpw.

a - Configuration : fichier slapd.conf

Localiser et éditer le fichier slapd.conf situé dans /etc/openldap.

- Déduire du nom de domaine un suffix utilisable pour l'annuaire, conforme aux recommandations de l'IETF
- En choisissant admin comme nom d'administrateur de l'annuaire, déduire le DN de root (rootdn)
- On utilisera bdb (Berkeley DB) comme type de base de données
- Utiliser la commande : *slappasswd -h '{SSHA}' -s secret -v*

  pour générer le mot de passe crypté cprrespondant à secret. Ce mot de passe crypté sera utilisé pour le champ rootpw.

Fichier /etc/openldap/slapd.conf :

```
include /usr/share/openldap/schema/core.schema
...
include /usr/share/openldap/schema/nis.schema
include /usr/share/openldap/schema/openldap.schema
```

```
include /usr/share/openldap/schema/autofs.schema
include /usr/share/openldap/schema/samba.schema
...
include /usr/share/openldap/schema/dnszone.schema
include /usr/share/openldap/schema/dhcp.schema
include /etc/openldap/schema/local.schema
include /etc/openldap/slapd.access.conf
access to dn.subtree="dc=tpldap,dc=tn"
by group="cn=Replicator,ou=Group,dc=tpldap,dc=tn"
by users read
by anonymous read
limits group="cn=Replicator,ou=Group,dc=tpldap,dc=tn"
size=unlimited
time=unlimited
pidfile /var/run/ldap/slapd.pid
argsfile /var/run/ldap/slapd.args
modulepath /usr/lib/openldap
TLSCertificateFile /etc/ssl/openldap/ldap.pem
TLSCertificateKeyFile /etc/ssl/openldap/ldap.pem
TLSCACertificateFile /etc/ssl/openldap/ldap.pem
database bdb
suffix "dc=tpldap,dc=tn"
rootdn "cn=admin,dc=tpldap,dc=tn"
rootpw {SSHA}pY6bNv0FBHvRcvVWG3YmmHd40OPxSSO1
directory /var/lib/ldap
checkpoint 256 5
index objectClass,uid,uidNumber,gidNumber,memberuid eq
index cn,mail,surname,givenname eq,subinitial
```

b - lancement du serveur

- Chercher dans le manuel l'option permettant de lancer l'exécutable stand-alone LDAP sous une identité donnée.
- Vérifier l'identité sous laquelle est lancée l'exécutable stand-alone LDAP (piste : consulter le fichier de démarrage du service ldap dans /etc/init.d).
- Démarrer le service ldap et vérifier les messages syslog.
- Extraire (grep) de la commande ps la ligne concernant le service ldap.
- Rechercher dans le fichier /etc/services le numéro officiel du port ldap.

3 Initialisation d'un annuaire

a - Création des objets préliminaires :

Pour pouvoir accéder aux objets de l'annuaire, il faut créer le suffixe de la base (ici, c'est le dn : dc=tpldap,dc=tn). Éditer le fichier init.ldif comme suit :

```
dn:dc=tpldap,dc=tn
objectClass:top
objectClass:dcObject
objectClass:organization
dc:tpldap
o:tpldap
```

- Ajouter les informations du fichier init.ldif avec la commande ldapadd :

  ```
  ldapadd –x –W –D "cn=admin,dc=tpldap,dc=tn" –f init.ldif
  ```

  (On donne le mot de passe correspondant à celui déclaré par le champ rootpw du fichier ldap.conf).

- Vérifier le fonctionnement de l'annuaire avec la commande ldapwhoami :

  ```
  ldapwhoami –x –W –D "cn=admin,dc=tpldap,dc=tn"
  ```

- On peut maintenant rajouter les informations concernant l'utilisateur admin. Éditer le fichier admin.ldif comme suit :

```
dn: cn=admin,dc=tpldap,dc=tn
objectClass:simpleSecurityObject
objectClass:organizationalRole
cn:admin
description:LDAP Administrator
userPassword:{SSHA}pY6bNv0FBHvRcvVWG3YmmHd40OPxSSO1
```

Remarque : le userPassword est identique à celui du fichier ldap.conf.

4 Migration des informations passwd/shadow/group dans un annuaire LDAP

Le but est d'utiliser l'annuaire LDAP à la place des fichiers /etc/passwd, /etc/group et /etc/shadow pour utiliser un mécanisme d'identification/authentification LDAP (dans ce cas, LDAP joue le même rôle que NIS). Pour tester, on va créer un utilisateur *testldap*. Les informations de connexion de cet utilisateur seront migrées sous LDAP, puis effacées des fichiers /etc/passwd, /etc/group et /etc/shadow. On pourra alors vérifier que le système LDAP permet bien la connexion de cet utilisateur.

- Créer un utilisateur *testldap* : `useradd testldap`.
- Lui affecter un mot de passe avec la commande : `passwd testldap`.

a - Récupération des outils de migration

- Rapatrier l'archive avec la commande :

  wget http://www.padl.com/download/MigrationTools.tgz.

- Développer l'archive : tar xvzf MigrationTools.tgz
- Aller dans le répertoire MigrationTools47, et éditer le fichier migrate_common.ph pour donner les bonnes valeurs aux variables DEFAULT_MAIL_DOMAIN et DEFAULT_BASE.
- Créer une version filtrée des fichiers *passwd* et *group* en ne retenant que les informations de *testldap*, et les mettre dans des fichiers *passwd* et *group* :

```
grep testldap /etc/group > group
grep testldap /etc/passwd > passwd
```

- Lancer les scripts migrate_group.pl et migrate_passwd.pl en prenant bien soin d'utiliser les versions raccourcies des fichiers passwd et group :

```
./migrate_group.pl group group.ldif
./migrate_passwd.pl passwd passwd.ldif
```

- Examiner les fichiers LDIF créés.

b - Création des objets préliminaires

Pour pouvoir enregistrer les utilisateurs, il faut créer les objets (de type organizationalUnit) People et Group. Éditer le fichier init2.ldif comme suit :

```
dn: ou=Group,dc=tpldap,dc=tn
objectclass: organizationalUnit
ou: Group
description: Groupes
dn: ou=People,dc=tpldap,dc=tn
objectclass: organizationalUnit
ou: People
description: People
```

- Ajouter les informations du fichier init2.ldif avec la commande ldapadd :

```
ldapadd –x –W –D "cn=admin,dc=tpldap,dc=tn" –f init2.ldif
```

- Ajouter les informations des fichiers *group.ldif* et *passwd.ldif* avec la commande *ldapadd* :

```
ldapadd –x –W –D "cn=admin,dc=tpldap,dc=tn" –f group.ldif

ldapadd –x –W –D "cn=admin,dc=tpldap,dc=tn" –f passwd.ldif
```

- Vérifier le contenu de l'annuaire avec la commande

```
ldapsearch –x –W –D "cn=admin,dc=tpldap,dc=tn" –b "dc=tpldap,dc=tn"
```

- Supprimer l'utilisateur testldap : userdel testldap.

c - Configuration du pc en client LDAP

La dernière opération consiste à configurer le PC en client LDAP, de façon qu'il puisse se servir de ce service pour permettre la connexion des utilisateurs.

Fichier /etc/nsswitch.conf : à modifier pour rajouter le service ldap

```
passwd: files nis ldap
shadow: files nis ldap
group: files nis ldap
```

Fichier /etc/ldap.conf : description du serveur LDAP et de l'annuaire utilisés par le service PAM du client. Le minimum à renseigner est :

```
host 192.168.x.y

base dc=tpldap,dc=tn
```

Si besoin, éditer le reste du fichier pour mentionner la branche dc=tpldap,dc=tn dans les lignes non commentées qui le nécessitent.

Fichier /etc/openldap/ldap.conf : description du serveur LDAP et de l'annuaire utilisés par le client. Le minimum à renseigner est :

```
BASE dc=tpldap, dc=tn

HOST 127.0.0.1

URI ldap://127.0.0.1/
```

On va configurer le service ssh pour qu'il utilise l'identification/authentification LDAP :

- éditer le fichier /etc/ssh/sshd.conf, et mettre le champ UsePAM à yes,
- copier le fichier /usr/share/doc/pam_ldap-170/pam.d/ssh à l'emplacement /etc/pam.d/sshd
- tester une connexion par ssh sous le compte testldap. : ssh testldap@localhost par exemple.