



Administration de Systèmes d'Information

Luiz Angelo Steffeneel





Cours 3

- ▶ Rappel sur les Services de base
 - ▶ Apache
 - Configuration
 - Serveurs Virtuels
 - Restriction d'accès
 - ▶ Serveur Mail
 - Serveur SMTP
 - POP3/IMAP/Webmail
 - ▶ DNS
 - ▶ DHCP





Serveur Web APACHE





Motivations

- ▶ Pourquoi Apache est devenu un standard ?
 - ▶ Coût nul
 - ▶ Code source disponible et modifiable
 - ▶ www.apache.org
 - ▶ Structuration modulaire
 - ▶ modules pour php, LDAP, sécurité, authentification, etc.
 - ▶ Stabilité et performance
 - ▶ la moitié des sites internet utilise Apache
- ▶ Le serveur Apache 2 répond actuellement pour environ 50% des sites web existants (les valeurs varient selon les sources)



APACHE – Installation sous Ubuntu / Debian

- ▶ `apt-get install apache2`
- ▶ `daemon : apache2`
 - ▶ port 80 (http)
 - ▶ port 443 (https)
- ▶ `script de démarrage : apache2`
 - ▶ `/etc/init.d/apache2 start`
- ▶ **Alternative : Installer LAMP (Linux Apache MySQL PHP)**
 - ▶ **`apt-get install apache2 php5-mysql libapache2-mod-php5 mysql-server`**
 - ▶ **ou `tasksel install lamp-server`**



Configuration de Base

- ▶ Lancer le serveur Apache
 - ▶ `/etc/init.d/apache2 start`
- ▶ Vérifier que Apache tourne correctement
 - ▶ `ps -aux | grep httpd` (vérifie si le serveur a été chargé)
 - ▶ naviguer sur <http://localhost> (affiche une page si correct)
 - ▶ le répertoire par défaut des fichiers est `/var/www/`
- ▶ Recharger les fichiers de configuration (après modification)
 - ▶ `/etc/init.d/apache2 reload`
- ▶ Arrêter le serveur
 - ▶ `/etc/init.d/apache2 stop`





Configuration

- ▶ `/etc/apache2/apache2.conf`
- ▶ Autres fichiers et répertoires utilisés (sous `/etc/apache2`)
 - ▶ `httpd.conf`
 - ▶ `ports.conf`
 - ▶ `conf.d/`
 - ▶ `mods-available`
 - ▶ `mods-enabled/`
 - ▶ `sites-available/`
 - ▶ `sites-enabled/`
 - ▶ `proxy-conf`



Édition du fichier apache2.conf

- ▶ À l'aide de votre éditeur préféré (vi, nano, etc.), ouvrir le fichier `/etc/apache2/apache2.conf`
- ▶ On y trouve notamment :
 - ▶ L'utilisateur et le groupe par défaut (détenteur des droits)

```
User www-data  
Group www-data
```
 - ▶ les fichiers par défaut (chargés sans qu'on donne leurs noms)

```
DirectoryIndex index.html index.php index.xhtml
```
 - ▶ le répertoire par défaut des pages des utilisateurs

```
# UserDir public_html
```



Édition du fichier apache2.conf

- ▶ Le journal des erreurs

```
ErrorLog /var/log/apache2/error.log
```

- ▶ Le fichier à afficher en cas d'erreur

```
ErrorDocument 404 /missing.html
```

- ▶ Une sortie formatée des connections

```
CustomLog /var/log/apache2/other.log vhost_combined
```

- ▶ Le fichier où sont définies les ports d'écoute

```
# Include ports listing  
Include /etc/apache2/ports.conf
```



Serveurs Virtuels - Motivation

- ▶ Il est courant d'héberger des dizaines de sites dans un seul serveur. Parfois, ces sites ont des adresses différents (magnum.univ-reims.fr, cosy.univ-reims.fr, ...)
- ▶ Lorsque les requêtes HTTP atteignent notre serveur http, celui-ci va regarder dans ses règles afin de trouver dans quel répertoire il doit se diriger. C'est là que la gestion des **virtual hosts** va intervenir
- ▶ Également, la réponse doit inclure dans l'en-tête http l'adresse virtuelle initialement demandée
- ▶ En conséquence, nous créerons une entrée pour chaque site hébergé sur notre serveur. Cette entrée contiendra le domaine prévu, et le répertoire de redirection





Configuration

- ▶ Configuration de deux sites : test1.com et test2.com
- ▶ Nous allons créer un répertoire pour chaque site sous /var/www
 - ▶ /var/www/test1
 - ▶ /var/www/test2
- ▶ Pour chaque site nous allons créer un fichier de configuration sous le répertoire */etc/apache2/sites-available*



Configuration de Base

- ▶ Créer deux fichiers (un par serveur virtuel) avec au moins ces informations

```
<VirtualHost IP:PORT>  
  ServerName NOM  
  DocumentRoot CHEMIN  
</VirtualHost>
```

- ▶ La balise `<VirtualHost IP:PORT>`
 - ▶ Indique quel est l'IP et port d'écoute (* ou *:80 généralement)
- ▶ La balise `ServerName NOM`
 - ▶ Indique le nom du serveur virtuel (utilisé pour filtrer les requêtes)
- ▶ La balise `DocumentRoot CHEMIN`
 - ▶ Indique le chemin des pages de ce serveur virtuel



Fichier de configuration pour www.test1.com

- `<VirtualHost *>`
ServerAdmin postmaster@test1.com
ServerName www.test1.com
ServerAlias test1.com
DocumentRoot /var/www/test1/
ErrorLog /var/www/test1/logs/error.log
ErrorDocument 404 /var/www/test1/erreur.html
LogLevel warn
`</VirtualHost>`



Fichier de configuration pour intranet.test1.com

- `<VirtualHost *>`
ServerAdmin postmaster@test1.com
ServerName intranet.test1.com
DocumentRoot /var/www/intranet/
ErrorLog /var/www/intranet/logs/error.log
ErrorDocument 404 /var/www/intranet/erreur.html
LogLevel warn
`</VirtualHost>`



L'heure de la vérité

- ▶ On sauvegarde les fichiers
- ▶ Pour activer le domaine nous faisons un lien symbolique dans le répertoire sites-enabled :

```
sudo a2ensite test1.com
```

```
sudo a2ensite test2.com
```

- ▶ On redémarre apache2 :
 - ▶ `/etc/init.d/apache2 restart`
- ▶ Et on peut accéder à notre répertoire :
 - ▶ `http://www.test1.com/`



Protéger l'accès à une page web

► Utilisation du contrôle **htaccess**

- 1 – modifier la configuration du serveur pour permettre le contrôle via htaccess

```
<VirtualHost *>
```

```
ServerAdmin postmaster@test1.com
```

```
ServerName intranet.test1.com
```

```
DocumentRoot /var/www/intranet/
```

```
<directory /var/www/intranet/>
```

```
    AllowOverride AuthConfig
```

```
    Order deny,allow
```

```
</directory>
```

```
    ErrorLog /var/www/intranet/logs/error.log
```

```
    ErrorDocument 404 /var/www/intranet/erreur.html
```

```
    LogLevel warn
```

```
</VirtualHost>
```



Créer un fichier de mots de passe

- ▶ Pour protéger un répertoire avec un mot de passe il faut :
 - ▶ Créer un fichier avec les mots de passe
 - ▶ `htpasswd -c /chemin/passwd user` (création du fichier)
 - ▶ `htpasswd /chemin/passwd autreuser` (rajouter un user)
 - ▶ Options : **-m (MD5) ou -d (crypt)** ou `-p` (plain text)
 - ▶ Rajouter les règles d'authentification dans le fichier `.htaccess` qui est dans le répertoire concerné

AuthType Basic

AuthName "Restricted Files"

(Following line optional)

AuthBasicProvider file

AuthUserFile /usr/local/apache/passwd/passwords

Require user rbowen



Groupe d'utilisateurs

- ▶ Créer un fichier groupes avec le format

GroupName: rbowen dpitts sungo rshersey

- ▶ Créer un fichier avec les mots de passe des utilisateurs
- ▶ Rajouter les règles d'authentification dans le fichier .htaccess qui est dans le répertoire concerné

AuthType Basic

AuthName "By Invitation Only"

Optional line:

AuthBasicProvider file

AuthUserFile /usr/local/apache/passwd/passwords

AuthGroupFile /usr/local/apache/passwd/groups

Require group GroupName

- ▶ Autre option est de rajouter uniquement "Require valid-user"
 - ▶ accepte uniquement les users avec un mot de passe



Questions de Sécurité

- ▶ Mots de passe stockés en format texte
 - ▶ option : utilisation d'une base de données

AuthBasicProvider dbm

AuthDBMUserFile /www/passwords/passwd.dbm

- ▶ Transmission du mot de passe en clair
 - ▶ mot de passe transmis à chaque lecture de page ou d'image d'un répertoire protégé
 - ▶ option : la méthode d'authentification **AuthType Digest** utilise la méthode d'hachage MD5
 - ▶ créer le fichier de mots de passe avec l'outil **htdigest**



.htaccess avec la méthode Digest

- ▶ Création du fichier de mots de passe

```
root# htdigest -c .passwd 'zone restreinte' user
```

- ▶ Puis modification de .htaccess

```
AuthType Digest
```

```
AuthName "zone restreinte"
```

```
AuthDigestDomain /intranet/ http://intranet.teste1.com
```

```
AuthDigestProvider file
```

```
AuthUserFile /var/www/intranet/.senhas
```

```
Require valid-user
```



En cas de problèmes

- ▶ Lisez les messages d'erreur
- ▶ Cherchez dans `/var/log/apache2/...` la description des problèmes
- ▶ Google ;)
 - ▶ <http://doc.ubuntu-fr.org/apache2>
 - ▶ http://doc.ubuntu-fr.org/tutoriel/virtualhosts_avec_apache2





Serveur Mail



Pourquoi gérer son propre serveur de courriers ?

- ▶ Éviter que le courrier interne passe par le FAI ou autre
 - ▶ Choix de l'interconnexion à des sites distants
 - ▶ Sécurité des données
 - ▶ Pas de limite de quotas
 - ▶ Filtrage sur mesure des courriers
-
- INCONVENIENTS :
 - Demande un niveau de compétences élevé
 - vis-à-vis des utilisateurs
 - vis-à-vis de la communauté Internet





Les fonctionnalités requises

- ▶ Echange de Mails dans l'entreprise et vers l'extérieur
- ▶ Consultation accessible par WebMail
- ▶ Protection contre les virus
- ▶ Filtrage des courriers non sollicités





Les besoins

- ▶ De préférence une IP Fixe
- ▶ Une connexion Internet Haut-Débit
- ▶ Un nom de domaine associé à votre IP
- ▶ Le protocole SMTP
- ▶ Les protocoles POP et IMAP
- ▶ Un routeur/NAT si le serveur est sur une IP privée



POSTFIX : le MTA

- ▶ Le MTA est l'agent de transfert des courriers
- ▶ POSTFIX est disponible : www.postfix.org
- ▶ En tant que MTA, POSTFIX ne fournit aucune fonctionnalité de récupération des courriers par les utilisateurs, il ne fournit que le protocole SMTP
- ▶ L'architecture
 - ▶ POSTFIX est composé de plusieurs processus (daemons)
 - ▶ Chacun de ces daemons à une fonction bien précise et distincte
 - ▶ Le daemon nommé **master** assure la gestion des différents processus



POSTFIX : La livraison des messages

- ▶ POSTFIX est doté de plusieurs agents de livraison de messages (MDA) :
 - ▶ Une partie client SMTP chargée de router les messages vers les autres serveurs via le protocole du même nom
 - ▶ Un agent de livraison LOCAL qui livre les messages aux utilisateurs locaux du système
 - ▶ Un agent VIRTUAL si le système héberge des boîtes aux lettres d'utilisateurs virtuels (Utilisateurs sans comptes Shell)
 - ▶ Des agents pour le traitement des erreurs, des files, etc.



POSTFIX : Installation

- ▶ Comme tous les logiciels sous Linux, POSTFIX peut-être installé de différents façons :
 - ▶ A partir des sources
 - ▶ A partir d'un paquetage propre à la distribution de Linux utilisée
 - ▶ apt-get install postfix (Debian / Ubuntu)



POSTFIX : Configuration

- ▶ Les fichiers de configuration sont au format texte et sont donc éditables facilement avec un simple éditeur.
- ▶ Les principaux fichiers sont :
 - ▶ **master.cf**
 - ▶ **main.cf**
- ▶ Ils se trouvent dans le dossier :
 - ▶ **/etc/postfix**
- ▶ Après chaque modification de ces fichiers, la configuration de postfix doit être rechargée :
 - ▶ **postfix reload**



POSTFIX : main.cf

- ▶ les informations sur votre domaine :
 - ▶ mydomain = a203.net
 - ▶ myhostname = debian-prof
 - ▶ myorigin = \$mydomain ou \$myhostname
 - ▶ mydestination = \$mydomain, \$myhostname, localhost.\$mydomain
 - ▶ relayhost =[smtp.monfai.com
 - ▶ mynetworks = 192.168.1.0/24, 172.16.0.0/16



POSTFIX : Mailbox ou Maildir

- ▶ Par défaut POSTFIX livre les messages locaux dans des fichiers au format Mailbox (mbox)
- ▶ Ces fichiers portent le nom de l'utilisateur de destination et se situent dans le dossier :
 - ▶ `/var/mail` ou `/var/spool/mail`
- ▶ Il existe UN seul fichier par boîtes aux lettres
- ▶ Une autre alternative est celle du format **QMail**
 - ▶ Il utilise plusieurs répertoires et un fichier par message
 - ▶ C'est le format nécessaire pour l'accès IMAP



POSTFIX : Mailbox ou Maildir

- ▶ Pour changer le format de boîte aux lettres, on utilise la commande **Home_mailbox** dans le fichier `main.cf` :
 - ▶ `Home_mailbox = Maildir` → utilise le format Mbox
 - ▶ `Home_mailbox = Maildir/` → utilise le format QMail
 - ▶ La différence est le /
- ▶ On peut aussi modifier l'endroit où la boîte d'arrivée sera stockée
 - ▶ `Home_mailbox = Maildir/`
 - ▶ `Home_mailbox = Mail/Maildir/`
- ▶ La commande **maildirmake** permet de créer les boîtes dans les répertoires utilisateurs



Lire le courrier : POP et IMAP

- ▶ POSTFIX ne fournissant que le protocole SMTP, il est nécessaire d'installer d'autres logiciels pour les utilisateurs puissent récupérer leurs messages :
 - ▶ courier-imap
 - ▶ courier-pop



Lire le courrier : WEBMAIL

- ▶ La solution Webmail présente plusieurs avantages, aussi bien du côté client que du côté administration
 - ▶ Une seule installation à réaliser : Celle de l'application WebMail sur le serveur
 - ▶ La possibilité de pouvoir interroger ses mails de n'importe quel endroit de la planète à partir d'un simple navigateur
- ▶ Elle peut présenter aussi quelques inconvénients :
 - ▶ Nécessite un espace disque plus important sur le serveur car les utilisateurs y laisseront la totalité de leurs messages
 - ▶ Pas de possibilité de consulter ses messages si l'utilisateur n'est pas connecté au réseau
 - ▶ La gestion des pièces jointes de tailles importantes est parfois impossible



Lire le courrier : Webmail

- ▶ Exemples de serveur Webmail :
 - ▶ SQUIRRELMAIL
 - ▶ HORDE
- ▶ L'installation nécessite généralement :
 - ▶ Un serveur Apache
 - ▶ Php
- ▶ Généralement les produits sont extensibles et de nombreux plug-ins sont fournis
- ▶ Comme tout les applications WebMail, des connexions sécurisées SSL sont possibles



Filtrage des messages : PROCMAIL

- ▶ Procmail fournit des règles permettant d'appliquer des traitements particuliers à tous les messages arrivant sur le système. Il permet entre autres :
 - ▶ De rediriger les messages vers des listes de distribution
 - ▶ De rediriger les messages vers certaines boîtes aux lettres en fonction de critères définis
 - ▶ De supprimer certains messages à l'arrivée
 - ▶ De mettre en place un service de répondeur automatique en cas d'absence prolongée
 - ▶ De passer le traitement du courrier à une autre application :
 - ▶ Antivirus (ClamAV, ...)
 - ▶ Antispams (Spamassassin, ...)



DNS



Résolution de Noms

- ◆ Adresses dans un réseau IP composés de 32 bits
- ◆ Représentation "facilitée" avec le format décimal pointé
 - ◆ 192.168.10.56
 - ◆ 200.18.42.1
- ◆ Ce format d'adresses est encore trop difficile pour les utilisateurs
- ◆ **Solution** : associer des noms aux adresses IP
 - ◆ 194.57.105.10 ↔ www.univ-reims.fr



Résolution de Noms sous ARPANet

- ▶ L'ARPANET des années 80 est constitué d'une centaine d'ordinateurs reliés
 - ▶ Structure suffisamment simple à gérer
- ▶ Un unique fichier hosts.txt rassemble les correspondances entre nom d'hôte et adresse IP
 - ▶ Le fichier hosts.txt est stocké sur le SRI-NIC (Stanford)
- ▶ Régulièrement, les machines téléchargent par FTP la nouvelle version du fichier

- NET : 10.0.0.0 : ARPANET :
- NET : 128.10.0.0 : PURDUE-CS-NET :
- GATEWAY : 10.0.0.77, 18.10.04 :
- MIT-GW.ARPA,
- MIT-GATEWAY : PDP-11 :
- MOS : IP/GW, EGP :
- HOST : 26.0.0.73, 10.0.0.51
- SRI-NIC.ARPA, SRI-NIC, NIC :
- DEC-2060 : TOPS-20 :
- TCP/TELNET, TCP/SMTP
- TCP/TIME, TCP/FTP
- TCP/ECHO, ICMP :
- HOST : 10.2.0.11 : SU-TAC.ARPA,
- SU-TAC : C/30 : TAC : TCP :m





Inconvénients

- ◆ La taille du fichier hosts.txt augmente avec le nombre d'hôtes
- ◆ En 1983, le réseau amorçe son expansion exponentielle
- ◆ La fréquence des mises-à-jours des tables devient proportionnelle au nombre de machines
- ◆ La consommation de bande passante est proportionnelle au carré du nombre d'hôtes



/etc/hosts

- Dans les machines actuelles, nous avons toujours un fichier hosts
- Utilisation
 - Créer des alias “locaux” pour les services les plus utilisés
 - Garder une base de données locale, indépendante du FAI
 - Tester le comportement de certains logiciels
 - Apache avec serveurs virtuels
- Exemple
 - 127.0.0.1 localhost, machine1.test.com, machine2.test.com
 - 10.22.1.15 printer
 - 255.255.255.255 broadcasthost
 - ::1 localhost
 - fe80::1%lo0 localhost



DNS : Domain Name System

- ◆ La croissance de l'Internet dans les années 80 et le déploiement du protocole SMTP (e-mail) motivent la définition du DNS
- ◆ Première spécification : RFC882, RFC883 (1983)
 - ◆ **DNS = Schéma de Nommage**
+ Système de base de données Distribué
- ◆ Système décentralisé de gestion de noms et d'adresses
 - ◆ Base de données distribuée, avec caches locaux
- ◆ Organisation hiérarchique

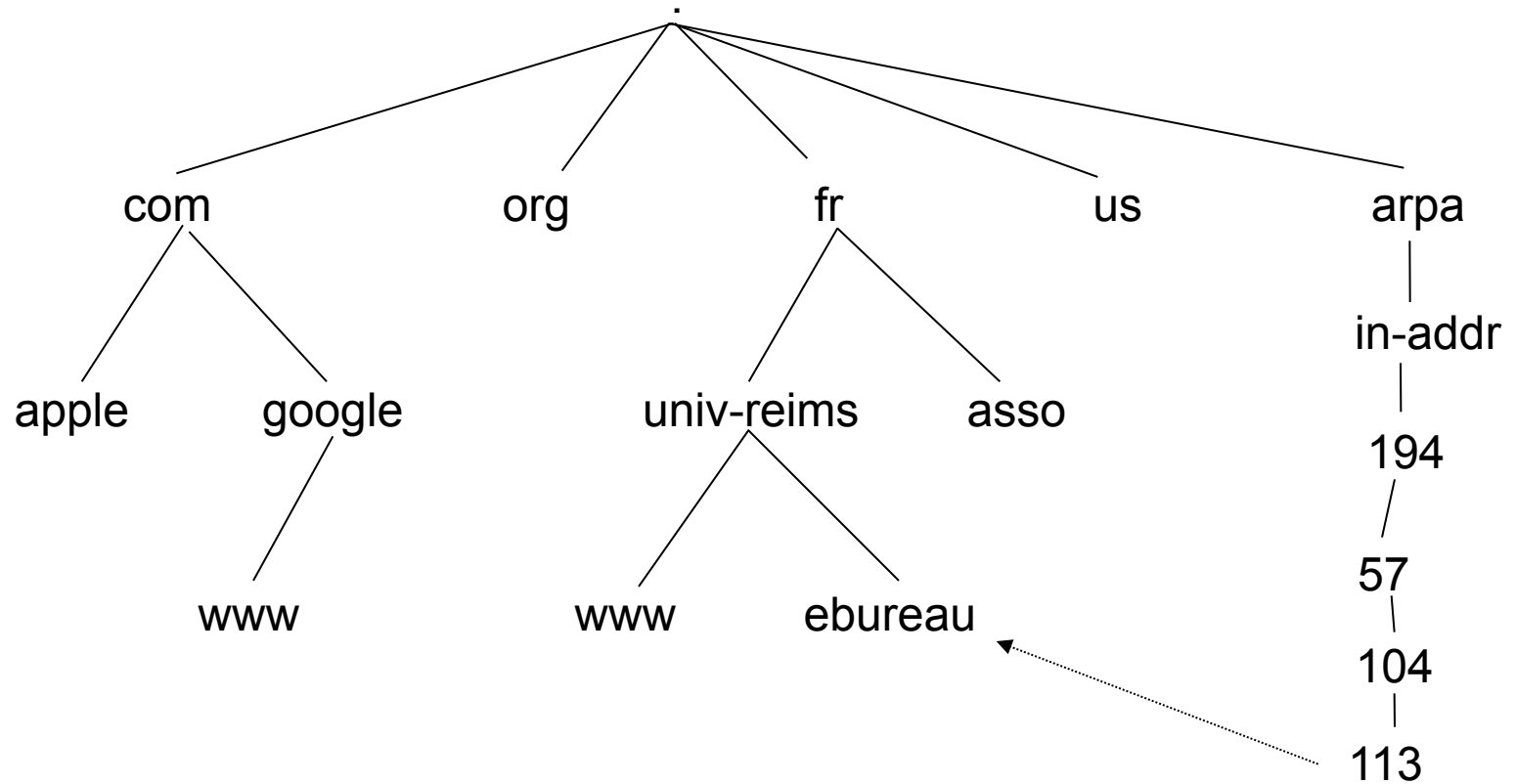


Caractéristiques

- ▶ Arbre de nommage globalement Unique
- ▶ Distribution bijective
 - ▶ un nom d'hôte peut désigner plusieurs adresses ip pour des interfaces différentes et vice-versa
- ▶ Distribution très forte
 - ▶ des données, de l'accès aux données,
 - ▶ de la responsabilité de gestion
- ▶ Motivations premières
 - ▶ [nom de machines -> adresse IP],
 - ▶ [adresse de mail -> adresse des serveurs de mail],
 - ▶ mais de nombreux autres types d'informations



Structure Arborescente des Noms



Les trois rôles d'un DNS

▶ RESOLVER

- ▶ Prend la requête de l'application, le formate dans un paquet UDP, et l'envoie au serveur cache

▶ SERVEUR AUTORITAIRE

- ▶ Contient les informations actuelles placées dans le DNS par le propriétaire du domaine

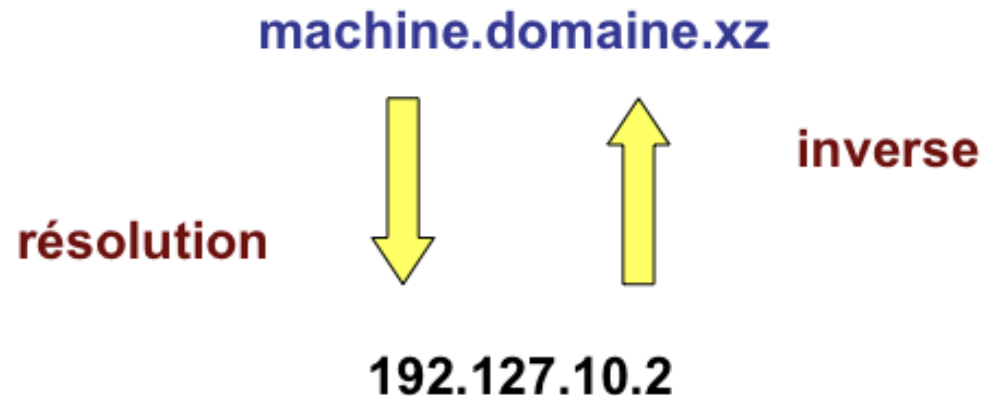
▶ Serveur CACHE

- ▶ Renvoie la réponse si déjà connue
- ▶ Dans le cas contraire, recherche le serveur autoritaire qui a l'information
- ▶ Met le résultat en cache pour les requêtes futures
- ▶ Egalement connu comme serveur récursif



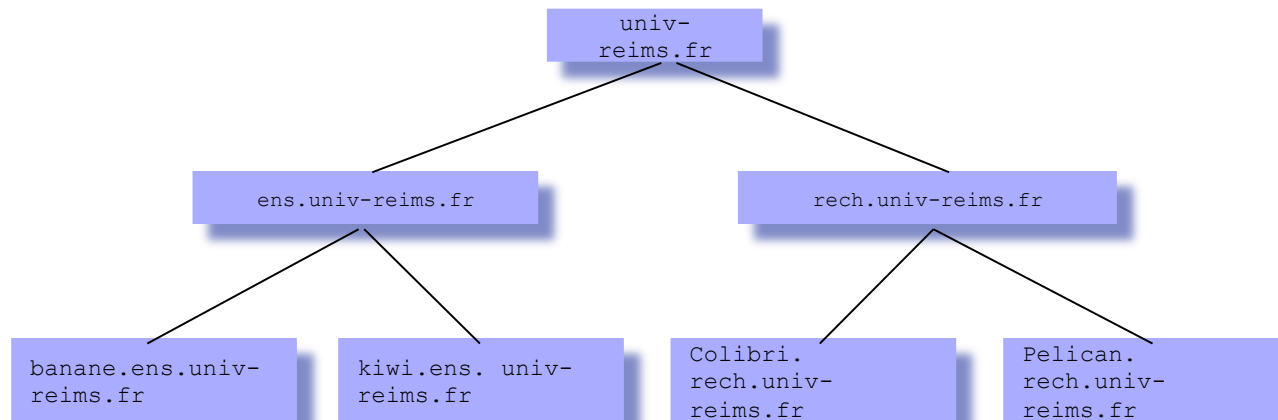
Le protocole DNS

- ▶ Le protocole Domain Name System est un ensemble de règles utilisées par les logiciels pour établir (entre autres choses) la correspondance entre des noms et des adresses
- ▶ Il utilise un protocole de communication client/serveur UDP/TCP sur le port 53



Résolution des noms

- ▶ Résolution par requête
 - ▶ non récursive : le serveur communique au client quel serveur celui-ci doit contacter pour pouvoir faire la résolution
 - ▶ récursive : le serveur communique la requête à un autre serveur. La récursivité se termine quand un serveur pouvant faire la résolution est trouvée
- ▶ Possibilité d'utiliser un cache pour éviter d'encombrer de réseau



Systeme Distribue DNS - Registre

- ◆ Registre :

- ◆ Nom-Domaine est un nom absolu de l'espace de nommage DNS (FQDN Fully Qualified Domain Name)
- ◆ CLASSE vaut IN pour internet, CH pour chaos, ...
- ◆ TYPE est le type de données du RR
- ◆ TTL Time To Live définit la durée de vie de l'objet dans les caches, en secondes
- ◆ RDATA est la valeur de l'objet (valeur associé au TYPE)

www.univ-reims.fr. 172800 IN A 194.57.105.10

"**www** a pour **Adresse IN**ternet IPv4 **194.57.105.10**, cette info est valide pour les **172800** secondes (2 jours) à venir"



Systeme Distribue DNS - Registre

- ▶ Quelques types de registre
 - ▶ A – traduction nom->adresse
 - ▶ A6 – traduction nom-> adresse IPv6
 - ▶ PTR – traduction adresse->nom (traduction reverse)
 - ▶ CNAME – alias
 - ▶ SOA (Start Of Authority) - indique l'autorité sur la zone
 - ▶ NS (Name Server) - adresses des serveurs de noms pour le domaine
 - ▶ MX – Mail eXchange (serveur email associé à une adresse)
- ▶ **Commentaires (IMPORTANT)**
 - ▶ les commentaires commencent avec un point-virgule ;
 - ▶ souvent source d'erreur



Registre de type SOA

```
$TTL 38400
```

```
foo.org. IN SOA ns1.foo.org. hostmaster.foo.org. (  
    20001210011      ; numéro de série  
    10800            ; rafraîchissement  
    3600             ; nouvel essai  
    604800           ; Obsolescence après une semaine  
    86400 )          ; TTL minimal de 1 jour
```

- Le nom de la zone (foo.org.) peut être remplacé par un @
- Le numéro de série doit augmenter !!!
- Toujours utiliser des noms FQDN !!!



Registre de type NS

```
foo.org.      IN NS  ns1.foo.org. ; noter le point final "."  
foo.org.      IN NS  ns2.foo.org. ;  
              ; IN signifie enregistrement de type INternet
```

- Le nom de la zone (foo.org.) peut être remplacé par un @
- Le numéro de série doit augmenter !!!
- Toujours utiliser des noms FQDN !!!



Registre de type A, CNAME, PTR

ns1.foo.org.	IN	A	192.168.0.1
ns2.foo.org.	IN	A	192.168.0.2
localhost.foo.org.	IN	A	127.0.0.1
www	IN	CNAME	ns1.foo.org.
ftp	IN	CNAME	ns1.foo.org.
1.0.168.192.in-addr.arpa.	IN	PTR	ns1.foo.org.
2.0.168.192.in-addr.arpa.	IN	PTR	ns2.foo.org.



Recherche dans le DNS

- ▶ DNS est automatiquement utilisé par toute application réseaux (la résolution DNS fait partie des API). On peut aussi y accéder directement :
 - ▶ au niveau du langage de commande (shell)
 - ▶ Commande host
 - ▶ commande dig (Domain Information Groper) - voir man
 - ▶ dig www.google.com
 - ▶ dig -x 194.199.25.39 # -x : adresse vers nom
 - ▶ au niveau des appels systèmes Unix

```
#include <netdb.h>

struct hostent *gethostbyname(const char *name);

struct hostent *gethostbyaddr(const char *addr, int len, 0);
```



Installation d'un Serveur DNS

- ▶ Pour l'installation d'un serveur DNS nous utilisons sous Unix l'application BIND (Berkeley Internet Name Daemon)
 - ▶ `apt-get install bind9 bind9-doc dnsutils`
- ▶ Éditer/créer les fichiers
 - ▶ `/etc/resolv.conf` – indique l'adresse IP du serveur DNS
 - ▶ `/etc/bind/named.conf` – fichier de configuration qui liste les zones (fait le lien avec le fichier des zones)
 - ▶ `/etc/bind/zones` – répertoire avec les fichiers de zones définis
 - ▶ Fichier de votre zone
 - ▶ Fichier avec la zone DNS reverse



Resolv.conf

- ▶ Le fichier `/etc/resolv.conf` indique à Linux où chercher les informations de DNS
- ▶ Exemple :
 - domain mydomain.fr**
 - search mydomain.fr**
 - nameserver 127.0.0.1**
 - nameserver x.x.x.x**
- ▶ Définit votre machine (127.0.0.1) comme le serveur pour le domaine mydomain.fr
- ▶ Remplacez mydomain.fr par votre nom de domaine et x.x.x.x par le serveur de secours





Les Zones

- ◆ Une zone est un sous arbre de l'arbre des noms de domaines sur lesquels un NS possède une information complète
- ◆ Une zone est gérée par une entité administrative particulière
 - ◆ L'autorité sur ce sous-arbre est déléguée
- ◆ La délégation est totale :
 - ◆ libre organisation
 - ◆ changements sans préavis
 - ◆ délégation de sous-zones



Définition d'une Zone de Nommage

- ▶ Une Zone DNS est formellement définie comme une partie connexe de l'arbre de nommage. Elle est donc constituée d'un domaine DNS (racine de la zone) et éventuellement de sous-domaines issus de ce domaine
- ▶ En générale, c'est le serveur AUTORITAIRE qui fait ça
 - ▶ Possibilité d'un serveur "esclave" synchronisé avec le "master"
- ▶ Afin d'enregistrer une zone, nous devons créer des entrées pour deux TYPES d'enregistrement
 - ▶ SOA – Désigne l'autorité pour le domaine
 - ▶ délimite la zone dont le Serveur de Noms est "autorité"
 - ▶ permet de répartir les réponses multiples fondées sur des caches
 - ▶ NS – Indique le Serveur de Noms pour ce domaine



Exemple de Réseau

- ▶ Supposons un réseau avec deux machines
 - ▶ un serveur DNS (102.253.253.1)
 - ▶ dns.mydomain.fr
 - ▶ un serveur pour WWW et MAIL (102.253.253.2)
 - ▶ www.mydomain.fr
 - ▶ mail.mydomain.fr
- ▶ Ce réseau présente des "alias" pour le serveur www/mail
- ▶ C'est de la responsabilité du serveur DNS, l'autorité pour la zone, de bien référencer les machines dans le domaine



Fichier /etc/bind/zones/ mydomain.db

- ▶ Fichier contenant les registres SOA et NS pour votre zone

```
; fichier pour la zone "mydomain"  
mydomain.fr. IN SOA dns.mydomain.fr. admin.mydomain.fr. (  
    2006081401 ; numéro série pour mise à jour  
    10800 ; mise à jour dans 3 heures  
    3600 ; nouvelle tentative après 1h  
    604800 ; expire après 1 semaine  
    38400 ; minimum TTL 1 semaine  
)
```

```
@      IN      NS      dns.mydomain.fr.  
dns    IN      A       102.253.253.1  
mail   IN      A       102.253.253.2  
@      IN      MX      10      mail.mydomain.fr.  
www    IN      CNAME   mail
```



Fichier /etc/bind/zones/ mydomain.rev

- ▶ Ce fichier contient les registres de pointeur PTR qui permettent la résolution reverse de votre adresse IP

```
@      IN      SOA      mydomain.fr. admin.mydomain.fr. (
      1      ; Serial
      604800 ; Refresh
      86400  ; Retry
      2419200 ; Expire
      604800 ) ; Default TTL

@      IN      NS      dns.mydomain.fr.
1.253.253.102.in-addr.arpa.  IN  PTR      dns.mydomain.fr.
2.253.253.102.in-addr.arpa.  IN  PTR      www.mydomain.fr.
```



Fichier /etc/named.conf

- ▶ Maintenant, nous allons faire la liaison du fichier de configuration named.conf aux fichiers de description des zones
- ▶ Rajouter à la fin du fichier /etc/named.conf

```
zone "mydomain.fr" {  
    type master;  
    file "zones/mydomain.db";  
};  
zone "253.102.in-addr.arpa" {  
    type master;  
    file "zones/mydomain.rev";  
};
```



Équilibrage de Charge

- ▶ L'équilibrage de charge peut être faite entre plusieurs serveurs :

```
www      IN      A      102.253.253.1  
www      IN      A      102.253.253.10  
www      IN      A      102.253.253.100
```



Délégation de zones

- ▶ Afin de "relier" votre serveur DNS au reste de l'Internet il faut qu'un serveur supérieur (zone mère) délègue l'autorité sur votre zone
 - ▶ On veut déléguer la zone "intra.mydomain.fr" à un serveur localisé sur "102.253.254.1"
- ▶ Il faut déléguer la zone intra et la zone "reverse"



Délégation de zones - mère

- On rajoute à mydomain.db (côté mère)
 - enregistrement qui décrit le serveur de nom de la zone intra
 - On voit deux serveurs DNS (dont celui de la zone principale)

```
intra.mydomain.fr.    IN      NS      ns.intra.mydomain.fr.
```

- enregistrement des adresses IP du serveur de nom
 - C'est la "colle" qui permet la résolution de ns.intra.mydomain.fr

```
ns.intra.mydomain.fr.  IN      A      102.253.254.1
```



Délégation de zones - fille

- On rajoute à `intra.db` (côté fille)
 - enregistrement qui décrit le serveur de nom de la zone `intra`
 - On voit deux serveurs DNS (dont celui de la zone principale)

```
IN      NS      dns.mydomain.fr.
```

- enregistrement des adresses IP du serveur de nom
 - C'est la "colle" qui permet la résolution de `dns.mydomain.fr`

```
dns.mydomain.fr.  IN      A      102.253.253.1
```



Délégation de zones - reverse

- La délégation de la zone reverse fonctionne différemment car elle dépende de la zone in-addr.arpa
 - Dans la pratique, les domaines sont limités à /24
 - On rajoute à 253.102.in-addr.arpa :

intra.mydomain.fr.

IN NS ns.intra.mydomain.fr.

1.254.253.102.in-addr.arpa

86400 IN PTR ns.intra.mydomain.fr.



Serveur Cache

- ▶ Pour rendre plus rapide votre réseau, vous pouvez définir un serveur cache seulement
- ▶ Pas besoin d'être un serveur autoritaire
- ▶ Il suffit d'éditer le fichier `/etc/bind/named.conf.options` :

[...]

```
forwarders {  
    1.2.3.4;  
    5.6.7.8;  
};
```

[...]

- ▶ Et de réinitialiser le service bind9 :
 - ▶ `sudo /etc/init.d/bind9 restart`





DHCP



Problèmes de gestion avec IP

- ▶ La Gestion des adresses IP
 - ▶ Les adresses IP doivent être unique
 - ▶ Nécessité d'une liste d'ordinateurs avec leurs adresses IP respectives
- ▶ La Gestion des principaux paramètres IP
 - ▶ Masques de sous-réseaux
 - ▶ Adresses IP du gateway
 - ▶ Serveurs DNS

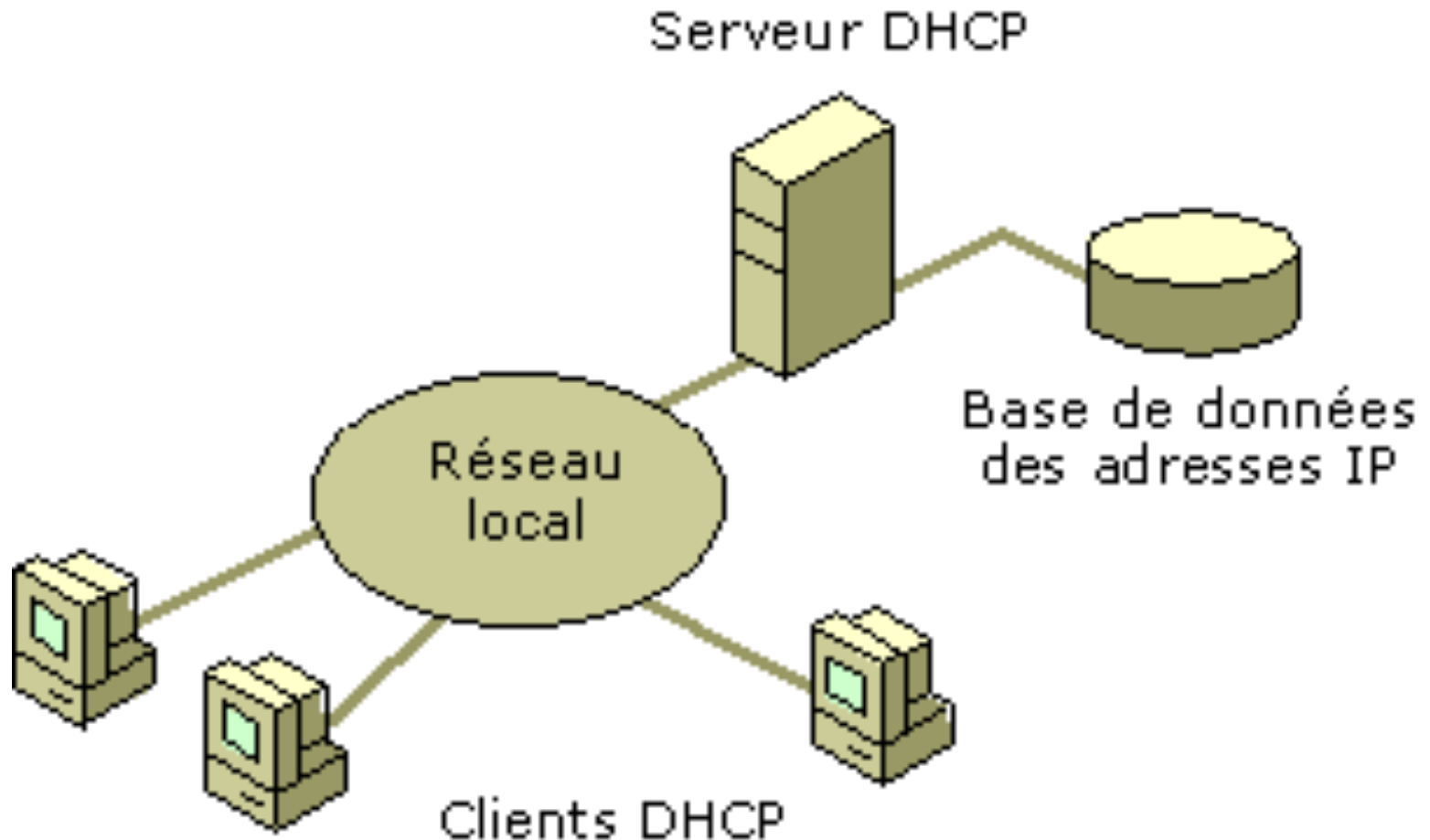


DHCP ?

- ▶ Dynamic Host Configuration Protocol
- ▶ Extension du protocole BOOTP
- ▶ Bâti sur un modèle client-serveur utilisant UDP
- ▶ Composé de deux parties :
 - ▶ Un protocole
 - ▶ Un mécanisme de création d'adresses
- ▶ DHCP permet :
 - ▶ Allocation dynamique des adresses IP et des noms d'hôte
 - ▶ Utilisation automatique de la plupart des paramètres de réseau
 - ▶ Maintenance des adresses IP en cours grâce au concept de "bail d'adresses IP"
 - ▶ Aide à la récupération de paramètres de réseau valides sur un système déplacé d'un réseau géré par DHCP à un autre



Schéma classique





Fonctionnement

- ▶ Modèle client-serveur
- ▶ Le client :
 - ▶ Vient de démarrer et réclame sa configuration.
- ▶ Le serveur :
 - ▶ détient la politique d'attribution des configurations IP.
 - ▶ envoie une configuration donnée pour une durée donnée, appelé bail à un client donné



Le Bail ?

- ▶ Définit par le serveur DHCP
- ▶ C'est l'intervalle de temps pendant lequel un client peut utiliser une adresse IP qui lui a été affectée
- ▶ Demande de renouvellement de l'adresse IP à $T1=1/2*\text{Bail}$
 - ▶ Si échec du renouvellement, nouvelle demande à $T2=0.875*\text{Bail}$
 - ▶ Si nouvelle échec, à expiration du bail, le client libère l'adresse IP attribué



Les messages transmis

- ▶ Plusieurs types de messages DHCP transmis via UDP
- ▶ Spécifié dans l'option 'type du message DHCP' de la trame DHCP
- ▶ Comme un seul "aller-retour" n'est pas suffisant pour une configuration complète
 - ▶ Plusieurs messages sont nécessaires pour une configuration
- ▶ Le client utilise le port 68, le serveur le port 67



Configuration d'un serveur DHCP

► Configuration File: /etc/dhcp3/dhcpd.conf

```
subnet 172.28.0.0 netmask 255.255.0.0 {
    option routers          172.28.1.254;
    option subnet-mask     255.255.0.0;
    option domain-name     "etudiant.univ-reims.fr";
    option domain-name-servers 172.31.1.1;
    range 172.28.4.2 172.28.7.254;
    default-lease-time 7200;
    max-lease-time 10800;
    host tc1 {
        hardware ethernet 00:80:64:1A:E9:14;
        fixed-address 172.28.120.32;
    }
}
```

