# La résolution de noms

# FQDN CQFD

Tous les internautes vous le diront, l'URL (ou URI) est le gouvernail de la navigation sur le Net. Ça fait déjà au moins trois sigles à expliquer :

- **FQDN :** Full Qualified Domain Name
  Le nom complet d'un hôte, sur l'Internet, c'est-à-dire de la machine jusqu'au domaine, en passant par les sous-domaines.
- **URL** : Uniform Resource Locator
  C'est la méthode d'accès à un document distant. Un lien hypertexte avec une syntaxe de la forme:
  <Type de connexion>://<FQDN>/[<sous-répertoire>]/.../<nom du document>
  Exemple: http://www.ac-aix-marseille.fr/bleue/francais/nouveau.htm
  - http: Hyper Text Transfert Protocol
  - www.ac-aix-marseille.fr: FQDN du serveur de pages personnelles
  - /bleue/francais/: arborescence de répertoires
  - nouveau.htm: nom du document.
- **URI** : Universal Resource Identifier.
  On ne va pas chipoter, c'est la même chose que l'URL. Le W3C (World Wide Web Consortium), garant de l'universalité de l'Internet, voudrait voir abandonner URL au profit d'URI. Notez la très subtile divergence de sens, qui vaut bien, le changement.

## La part des choses

Donc, ne confondons pas tout, un FQDN est significatif d'un hôte sur l'Internet (un serveur la plupart du temps), alors qu'un URI définit l'accès à un document sur un serveur. L'URI contient donc un FQDN, mais pas seulement.

L'objet de ce chapitre est de donner des détails sur le prodige qui fait qu'un hôte sur le Net peut être retrouvé à partir d'un patronyme, autrement dit, comment un FQDN est converti en adresse IP.

## Les DNS

Domain Name System. Les serveurs DNS sont là pour réaliser cette opération et l'inverse également (retrouver un nom à partir d'une adresse IP). Votre fournisseur d'accès met à votre disposition un serveur de ce type, dont l'adresse IP vous est habituellement donnée de façon automatique lors de la transaction DHCP (voir le chapitre à ce sujet[1]), ou via le protocole PPP.

Nous allons ici décortiquer le fonctionnement d'un tel serveur et même voir comment l'on peut s'en construire un personnel, aussi efficace (sinon plus) que celui de votre FAI.

1 DHCP : http://christian.caleca.free.fr/dhcp/index.html

# Plan du chapitre

# Notions de base

## Débutants

Ce paragraphe est principalement destiné aux internautes qui n'ont encore aucune notion des subtilités des réseaux en général et du Net en particulier. Il n'y a aucun mal à çà, mais c'est tout de même mieux de savoir, surtout lorsque ça ne marche pas comme on le voudrait.

Dans cette partie, nous verrons dans le détail :

- Comment est construit un FQDN.
- Comment la hiérarchie des domaines est organisée.
- A quoi sert toute cette organisation.

## Constatations de base

Il existe sous Windows NT et sous Linux (malheureusement pas à ma connaissance sous windows 95/98), une commande bien pratique: "nslookup". Cette commande "is deprecated" dans les dernières distributions à base de noyau 2.4, ce qui va permettre de devoir se farcir en plus les commandes "host" et "dig", qui font en gros la même chose, mais avec une syntaxe complètement différente. Nous aurons donc la joie de nous les taper toutes...

### Nslookup

**Note: Les exemples sont anciens, ils sont tirés d'une antique Mandrake 7.2 sur une connexion câble, du temps où nous utilisions DHCP. Ca n'enlève rien à la justesse de ces exemples.**

Cette commande est une "usine à gaz", qui peut s'utiliser en mode "ligne de commande" ou en mode "interactif".

Elle permet de contrôler le bon fonctionnement d'un DNS en l'interrogeant de diverses manières. La question la plus simple étant: "Quelle est l'adresse de la machine X" ou encore "Quel est le nom de la machine dont l'adresse est xxx.yyy.zzz.ttt ?"

```
[root@gateway2 /root]# nslookup www.wanadoo.fr
Server:  ns0.mrs.ftci.oleane.com
Address:  62.161.120.11


Non-authoritative answer:
Name:    www.wanadoo.fr
Address:  193.252.19.189
```

Lorsque la question est posée au DNS par défaut, il commence par se présenter (son nom et son adresse), puis répond à la question.
La "Non-authoritative answer" est intéressante, on la met de côté pour l'instant...

Autre exemple :

```
[root@gateway2 /root]# nslookup ca-ol-marseille-1-85.abo.wanadoo.fr
Server:  ns0.mrs.ftci.oleane.com
```

```
Address:  62.161.120.11

Name:    ca-ol-marseille-1-85.abo.wanadoo.fr
Address:  62.161.96.85
```

Le genre de nom simple et facile à retenir dont nous sommes affublés par Wanadoo Câble... Mais en général, tous les FAI affublent leurs clients de noms aussi poétiques.

## Host

Voici la même chose, avec la commande host des distributions à base de 2.4 :

```
[root@gw1 root]# host www.wanadoo.fr
www.wanadoo.fr has address 193.252.19.189
```

Et dans l'autre sens :

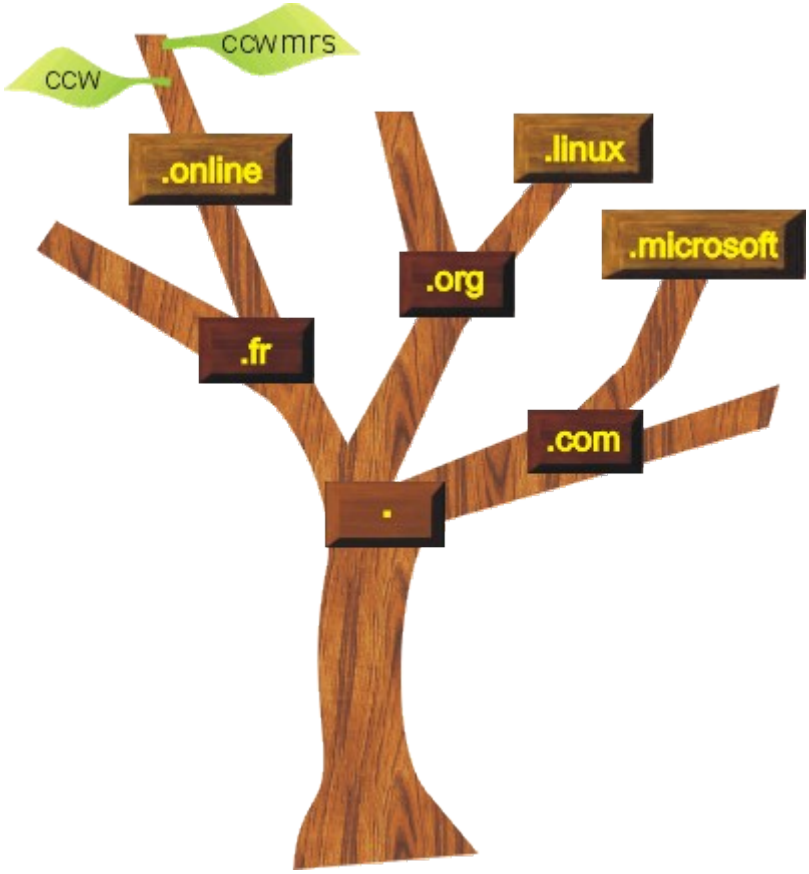
```
[root@gw1 root]# host 193.252.19.189
189.19.252.193.in-addr.arpa domain name pointer www.wanadoo.fr.
```

## Conclusion

Il existe donc un moyen simple d'emploi, qui permet de trouver l'adresse IP d'un hôte lorsque l'on connaît son nom FQDN et réciproquement. Ce même mécanisme est bien évidemment utilisé par votre "browser", votre outil de messagerie, votre outil de consultation de news... Tout protocole du niveau application, qui a besoin de résoudre des FQDNs.

Pour les personnes curieuses, la suite va expliquer comment ça marche et même comment on peut se débrouiller tout seul pour réaliser ce genre d'opérations.

# L'exemple à suivre

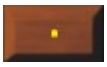



Lorsque l'on doit organiser des éléments qui présentent une certaine cohérence entre eux, la nature nous a donné un excellent exemple à suivre sous la forme de l'arbre.

Les informaticiens ne se sont pas privés d'utiliser cette organisation dans de très nombreux domaines.

Le plus connu des utilisateurs est certainement celui de l'organisation des répertoires sur la mémoire de masse.

Les domaines de l'Internet sont organisés de la même manière.




Au départ, il existe un domaine "racine" qui est sous-entendu. Il n'est jamais indiqué dans les FQDN, mais il existe, au même titre que le "\" dans l'arborescence Windows ou le "/" dans celle d'Unix.


Il existe ensuite un certain nombre de domaines "génériques", le plus connu étant ".com", mais il n'est pas le seul. Tous les pays en ont un, ".fr" étant le notre.

Ces domaines sont appelés TLD (Top Level Domains).

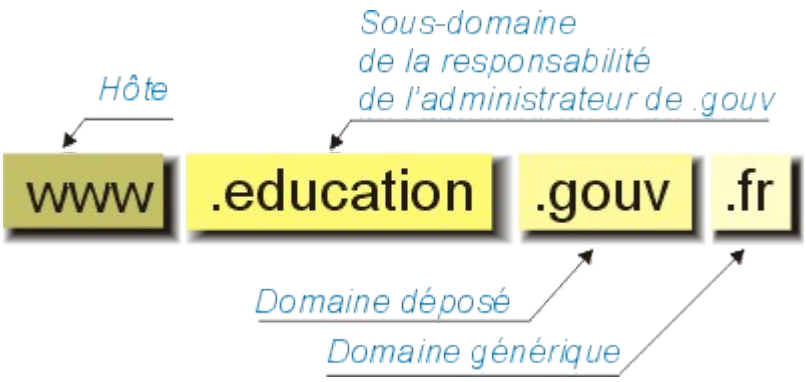
La création et l'emploi de ces domaines génériques ne s'est pas pas toujours faite avec le plus grand discernement. Les premiers ont été créés par les américains, certains d'entre eux sont encore à leur usage exclusif comme ".gov" ou ".mil" Par la suite, les domaines nationaux ont été créés, y compris pour les Etats-Unis, mais chaque pays est libre d'organiser son domaine générique comme il l'entend. Les domaines ".org", ".net" et ".com" sont utilisés dans tous les pays.


Viennent ensuite les noms de domaines réservés par les gestionnaires de sites. Ces noms de domaines doivent être déposés auprès de l'organisme responsable du domaine générique demandé et doivent bien évidemment être uniques dans le domaine générique en question.
Ces domaines sont gérés par leur propriétaire, qui est libre d'y intégrer autant de sous-domaines qu'il le souhaite.
Exemple : "eu.microsoft.com" ou encore "education.gouv.fr"
Dans les deux cas, il ne s'agit pas d'hôtes mais bien de sous-domaines


Viennent enfin les "feuilles" qui sont les hôtes; exemples :
ccwmrs.online.fr
www.education.gouv.fr
www.eu.microsoft.com

## Conclusions

- La partie la plus à gauche représente toujours un hôte.
- La partie la plus à droite représente toujours un domaine générique (TLD).
- Entre les deux, les éventuels sous-domaines et le domaine déposé de l'entité concernée




# Le serveur Domain Name System

Votre fournisseur d'accès met à votre disposition un outil pour traduire les noms FQDN en adresses IP. Chez Wanadoo , les DNS fournis aux clients sont généralement 193.252.19.3 et 193.252.19.4. Ce n'est pas une règle absolue, principalement pour les câblés.

Ce Serveur DNS travaille de manière "récursive" . Autrement dit, vous n'avez pas de questions à vous poser (sauf lorsqu'il ne fonctionne plus). Vous (ou votre logiciel) demandez de traduire un nom et vous attendez sagement la réponse, il s'occupe de tout. Le mécanisme est complètement transparent. Il est mis en route, par exemple, lorsque vous tapez quelque chose comme "http://www.google.fr" dans la barre d'adresse de votre navigateur.

Comme l'adresse de votre DNS vous est donnée avec le bail que vous accorde le DHCP (ou PPP), vous n'avez généralement même pas besoin de connaître son adresse IP.

Où les choses se compliquent un peu, c'est si vous avez par exemple réalisé la passerelle Linux entre le modem et votre (ou vos) ordinateur(s) personnel(s). Dans ce cas, la machine Linux est bien documentée sur l'adresse du DNS, mais pas votre ou vos postes qui sont, eux, configurés en "dur". Ils ont donc besoin de connaître une adresse de DNS pour pouvoir fonctionner.

Deux solutions sont alors possibles :

- Vous allez sur votre machine Linux, un petit coup de "nslookup" ou de "host -v" vous

indiquera l'adresse du DNS, il ne vous restera plus qu'à la fournir à votre ou à vos postes de travail du réseau privé.

- Vous prenez votre courage à deux mains, vous lisez attentivement tout ce qui est dit dans ce chapitre et vous construisez sur votre passerelle Linux votre propre serveur DNS :-)

# Notions avancées

## Pour les curieux

Cette partie s'adresse à ceux qui ont déjà quelques notions sur l'adressage IP et le rôle des DNS (la lecture et sa compréhension de la partie pour les débutants suffit) et qui sont curieux de savoir comment ça fonctionne de plus près.

Nous allons ici découvrir comment fonctionne un serveur DNS "lambda", tel que celui que nous pourrions construire, tel que celui de votre FAI.

### Mise en garde

Vous imaginez bien qu'avec le nombre de millions d'hôtes référencés dans le monde, construire un système de résolution de noms à l'échelle planétaire qui fonctionne et qui soit perpétuellement à jour, n'est pas un petit travail. Il va donc falloir s'attendre à ce que le processus soit un peu compliqué, mais nous verrons qu'il est en fait extrêmement logique.

## Rappels sur les "TLD"

Souvenons-nous que juste au dessous de la racine "." se trouvent un certain nombre de "Top Level Domains" comme ".com", ".net", ".fr" etc.

Des organismes de gestion de ces TLD exitent, ils permettent d'enregistrer des noms de domaines dans les TLD comme par exemple "wanadoo.fr" ou "voila.fr". En France, c'est le NIC France qui s'en occupe. Ces organismes doivent s'assurer de l'unicité d'un nom de domaine à l'intérieur d'un TLD donné.

Lorsqu'un domaine est déclaré, toutes les machines appartenant à ce domaine seront référencées sur le ou les serveurs DNS de l'organisme qui a déposé le nom de domaine. Le NIC France se contentant normalement de permettre de retrouver les adresses IP de ces serveurs de noms.

Le rôle des serveurs racine (Root Servers) consiste à permettre de retrouver les adresses des serveurs que l'on vient de décrire. Comme ceci n'est pas forcément très clair, voyons çà sur un exemple.

## Le processus

- Nous partons d'un DNS racine, pour lui demander s'il connaît des DNS qui savent répondre pour le TLD recherché,
- nous interrogeons alors l'un de ces DNS pour trouver un DNS qui sache répondre pour le domaine d'entreprise donné,
- enfin, nous interrogeons l'un de ces DNS, pour qu'il résolve le FQDN recherché.

Les serveurs racine sont connus. Nous pouvons trouver leur liste, par exemple, ici : ftp://ftp.rs.internic.net/domain/named.root
Ce fichier est maintenu par l'InterNIC, organisme qui regroupe d'autres organismes comme le NIC

France.

A l'heure où ces lignes sont écrites, ce fichier contient ceci :

```
;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC
;       under anonymous FTP as
;           file                /domain/named.root
;           on server           FTP.INTERNIC.NET
;
;       last update:    Nov 5, 2002
;       related version of root zone:   2002110501
;
;
; formerly NS.INTERNIC.NET
;
.                        3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
;
; formerly NS1.ISI.EDU
;
.                        3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000      A     128.9.0.107
;
; formerly C.PSI.NET
;
.                        3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.      3600000      A     192.33.4.12
;
; formerly TERP.UMD.EDU
;
.                        3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.      3600000      A     128.8.10.90
;
; formerly NS.NASA.GOV
;
.                        3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.      3600000      A     192.203.230.10
;
; formerly NS.ISC.ORG
;
.                        3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.      3600000      A     192.5.5.241
;
; formerly NS.NIC.DDN.MIL
;
.                        3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.      3600000      A     192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
;
.                        3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.      3600000      A     128.63.2.53
;
; formerly NIC.NORDU.NET
;
.                        3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.      3600000      A     192.36.148.17
;
; operated by VeriSign, Inc.
;
.                        3600000      NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.      3600000      A     192.58.128.30
;
; housed in LINX, operated by RIPE NCC
;
.                        3600000      NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.      3600000      A     193.0.14.129
```

```
;
; operated by IANA
;
.                       3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.     3600000      A     198.32.64.12
;
; housed in Japan, operated by WIDE
;
.                       3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.     3600000      A     202.12.27.33
; End of File
```

Normalement, l'un quelconque de ces serveurs suffit à démarrer sa recherche.

# Recherche du DNS de ac-aix-marseille.fr

Dans cet exemple, nous allons essayer de trouver les adresses et les noms des serveurs DNS capables de nous renseigner sur les hôtes du domaine "ac-aix-marseille.fr".

## Faisons-le avec nslookup :

| | |
|---|---|
| `Microsoft(R) Windows NT(TM)`<br>`(C) Copyright 1985-1996 Microsoft Corp.` | L'exemple est pris à partir d'une console sous Windows NT 4.<br>Cette machine est connectée par l'intermédiaire d'une passerelle sous LINUX. |
| `D:\>nslookup`<br>`Serveur par défaut : gateway1.maison.mrs`<br>`Address: 192.168.0.250` | On démarre NSLOOKUP en mode interactif.<br>Le DNS par défaut est celui qui est spécifié dans les options du protocole TCP/IP. Ici un DNS personnel construit sur la passerelle LINUX. |
| `> server g.root-servers.net`<br>`Serveur par défaut : g.root-servers.net`<br>`Address: 192.112.36.4` | Cette commande signifie à nslookup d'utiliser le serveur spécifié, ici l'un des "root-servers" choisi au hasard. |
| `> set q=ns` | Cette commande indique à nslookup que l'on va s'intéresser aux champs de type "NS" (Name Servers). |
| `> fr.`<br>`Serveur: g.root-servers.net`<br>`Address: 192.112.36.4`<br><br>`Réponse de source secondaire :`<br>`fr nameserver = NS1.NIC.fr`<br>`fr nameserver = NS3.NIC.fr`<br>`fr nameserver = DNS.INRIA.fr`<br>`fr nameserver = NS2.NIC.fr`<br>`fr nameserver = NS.EU.NET`<br>`fr nameserver = DNS.PRINCETON.EDU`<br>`fr nameserver = NS-EXT.VIX.COM`<br>`fr nameserver = DNS.CS.WISC.EDU`<br><br>`NS1.NIC.fr internet address = 192.93.0.1`<br>`NS3.NIC.fr internet address = 192.134.0.49`<br>`DNS.INRIA.fr internet address = 193.51.208.13`<br>`NS2.NIC.fr internet address = 192.93.0.4`<br>`NS.EU.NET internet address = 192.16.202.11`<br>`DNS.PRINCETON.EDU internet address = 128.112.129.15`<br>`NS-EXT.VIX.COM internet address = 204.152.184.64`<br>`DNS.CS.WISC.EDU internet address = 128.105.2.10` | Posons la question :<br>Quels sont les serveurs de noms qui savent donner des informations sur le TLD "fr." ?<br><br>Et voilà, nous disposons de la liste des serveurs qui pourront nous renseigner sur les domaines existant dans le TLD "fr." |

```
> server dns.inria.fr
Serveur par défaut :  dns.inria.fr
Address:  193.51.208.13
```

Passons donc sur l'un de ces serveurs...

```
> ac-aix-marseille.fr.
Serveur:  dns.inria.fr
Address:  193.51.208.13

Réponse de source secondaire :
ac-aix-marseille.fr     nameserver = dnse.ac-aix-marseille.fr
ac-aix-marseille.fr     nameserver = cianame.ac-clermont.fr

dnse.ac-aix-marseille.fr internet address = 195.83.252.200
cianame.ac-clermont.fr   internet address = 194.254.204.31
```

Et interrogeons-le sur le domaine ac-aix-marseille.fr.

Nous obtenons les DNS qui nous renseigneront sur les hôtes du domaine ac-aix-marseille.fr.

## Faisons le avec host :

host, contrairement à nslookup, n'est pas une commande interactive, son usage est quelque peu différent, mais les résultats devraient être les mêmes.

```
[root@gw1 root]# host -t ns fr. 192.112.36.4
```

Nous demandons à 192.112.36.4 quels sont les "name servers" (-t ns) qui gèrent le TLD fr (fr.)

```
Using domain server:
Name: 192.112.36.4
Address: 192.112.36.4#53
Aliases:

fr name server NS3.DOMAIN-REGISTRY.NL.
fr name server DNS.CS.WISC.EDU.
fr name server NS1.NIC.fr.
fr name server NS3.NIC.fr.
fr name server DNS.INRIA.fr.
fr name server NS2.NIC.fr.
fr name server DNS.PRINCETON.EDU.
fr name server NS-EXT.VIX.COM.
```

La réponse arrive.

C'est sympa, nous avons les noms de DNS, mais pas leurs adresses...

```
[root@gw1 root]# host ns1.nic.fr. 192.112.36.4
```

Nous demandons donc au même DNS de bien vouloir nous indiquer l'IP de ns1.nic.fr.

```
Using domain server:
Name: 192.112.36.4
Address: 192.112.36.4#53
Aliases:

ns1.nic.fr has address 192.93.0.1
```

Ce qu'il fait sans rechigner, puis qu'on le lui demande poliment.

```
[root@gw1 root]# host -t ns ac-aix-marseille.fr. 192.93.0.1
```

Nous demandons maintenant à ns1.nic.fr (192.93.0.1) de nous indiquer les DNS (-t ns) qui savent répondre pour le domaine d'entreprise ac-aix-marseille.fr

```
Using domain server:
Name: 192.93.0.1
Address: 192.93.0.1#53
Aliases:

ac-aix-marseille.fr name server dnse.ac-aix-marseille.fr.
ac-aix-marseille.fr name server cianame.ac-clermont.fr.
```

Ce qu'il fait, toujours en n'indiquant que les noms.

```
[root@gw1 root]# host dnse.ac-aix-marseille.fr. 192.93.0.1
```

Nous aimerions avoir l'adresse IP de dnse.ac-aix-marseille.fr, par exemple :

```
Using domain server:
Name: 192.93.0.1
Address: 192.93.0.1#53
Aliases:

dnse.ac-aix-marseille.fr has address 195.83.252.200
```

et voilà.

### Conclusions

Cette petite manipulation nous permet de comprendre comment, en partant d'un des root-servers de l'Internic, il est possible de retrouver le ou les DNS capables de renseigner sur un domaine donné. Cette façon de faire, si elle ne présente aucun intérêt pour l'internaute "moyen" (sans aucune connotation péjorative de ma part) permet de comprendre comment va fonctionner un serveur DNS de type "récursif" comme celui que vous procure normalement votre fournisseur d'accès; cette méthode permet également de dépister certaines pannes d'inaccessibilité à un serveur. Nous le verrons d'ailleurs dans un "sniff" gigantesque un peu plus loin...

# Recherche habituelle de l'adresse d'un hôte

## Le DNS par défaut

Simple... Vous vous adressez à votre DNS par défaut :

```
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

D:\>nslookup www.ac-aix-marseille.fr
Serveur: gateway1.maison.mrs
Address: 192.168.0.250

Nom : copernic.crdp.ac-aix-marseille.fr
Address: 194.254.139.4
Aliases: www.ac-aix-marseille.fr
```

Et, s'il est normalement constitué, il vous donne la réponse. Allez tiens, encore un coup pour le plaisir :

```
D:\>nslookup www.ac-aix-marseille.fr
Serveur: gateway1.maison.mrs
Address: 192.168.0.250

Réponse de source secondaire :
Nom : copernic.crdp.ac-aix-marseille.fr
Address: 194.254.139.4
Aliases: www.ac-aix-marseille.fr
```

Tiens, cette fois-ci, il y a quelque chose en plus: "Réponse de source secondaire:"! On se le remet de côté pour plus tard...

Bien entendu, nous pourrions faire la même chose avec host, sans préciser de DNS. C'est alors le DNS par défaut qui sera interrogé :

```
[root@gw1 root]# host www.ac-aix-marseille.fr
www.ac-aix-marseille.fr has address 195.83.252.87
```

Tant qu'on y est, une petite variation sur le thème "host" : le paramètre v (verbose) :

```
[root@gw1 root]# host -v www.ac-aix-marseille.fr
Trying "www.ac-aix-marseille.fr"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11747
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 1

;; QUESTION SECTION:
;www.ac-aix-marseille.fr. IN A

;; ANSWER SECTION:
www.ac-aix-marseille.fr. 604765 IN A 195.83.252.87
```

```
;; AUTHORITY SECTION:
ac-aix-marseille.fr. 603884 IN NS cianame.ac-clermont.fr.
ac-aix-marseille.fr. 603884 IN NS dnse.ac-aix-marseille.fr.

;; ADDITIONAL SECTION:
dnse.ac-aix-marseille.fr. 603884 IN A 195.83.252.200

Received 126 bytes from 127.0.0.1#53 in 11 ms
```

Pour être verbeux... Mais nous avons quelques informations qui peuvent intéresser :

- Le DNS qui a répondu (dernière ligne 127.0.0.1, c'est mon DNS local),
- les DNS qui font autorité pour le domaine ac-aix-marseille.fr,
- en prime, l'IP de dnse.ac-aix-marseille.fr.

Bien, mais on ne va pas rester en plan sur la recherche que nous avons faite en partant d'un root-server...

## Jusqu'au bout de la recherche

Continuons la manip jusqu'au bout...

(Je rappelle que nous avions démarré nslookup en mode interactif, que nous étions parti d'un root-server choisi au hasard dans la liste distribuée par l'Internic, et que nous avions trouvé deux serveurs de noms pour le domaine ac-aix-marseille.fr.

Prenons-en un, presque au hasard :

| | |
|---|---|
| `> server dnse.ac-aix-marseille.fr`<br>`Serveur par défaut : dnse.ac-aix-marseille.fr`<br>`Address: 195.83.252.200` | passons maintenant sur l'un des serveurs DNS du domaine ac-aix-marseille.fr: |
| `> set q=a` | Le type de question est maintenant : donner l'adresse de... |
| `> www.ac-aix-marseille.fr.` | du serveur web de l'académie |
| `Serveur: dnse.ac-aix-marseille.fr`<br>`Address: 195.83.252.200`<br><br>`Nom : copernic.crdp.ac-aix-marseille.fr`<br>`Address: 194.254.139.4`<br>`Aliases: www.ac-aix-marseille.fr` | Et la réponse arrive. On remarque d'ailleurs que le véritable nom du serveur est : |

Et avec host :

| | |
|---|---|
| `[root@gw1 root]# host www.ac-aix-marseille.fr 195.83.252.200` | allons jusqu'au bout pour obtenir de dnse.ac-aix-marseille.fr (195.83.252.200) l'IP du serveur web de l'académie : |
| `Using domain server:`<br>`Name: 195.83.252.200`<br>`Address: 195.83.252.200#53`<br>`Aliases:`<br><br>`www.ac-aix-marseille.fr has address 195.83.252.87` | C'est terminé. |

**Ce qui va être fait maintenant n'est pas à faire pour s'amuser ! Ce genre de commande consomme inutilement des ressources sur le serveur DNS visé. A réserver à l'administrateur du DNS (auprès duquel je m'excuse, c'était juste à des fins pédagogiques).**
**D'ailleurs, l'administrateur dispose de la possibilité d'interdire ce genre de requête, comme nous le verrons dans la construction d'un DNS, et lorsque ce n'est pas interdit, ça ne veut pas**

**forcément dire que c'est autorisé...**

```
> ls -d ac-aix-marseille.fr.
[dnse.ac-aix-marseille.fr]
 ac-aix-marseille.fr.      SOA     dnse.ac-aix-marseille.fr
 ...
 ac-aix-marseille.fr.      A       195.83.252.20
 ac-aix-marseille.fr.      NS      dnse.ac-aix-marseille.fr
 ac-aix-marseille.fr.      NS      cianame.ac-clermont.fr
...
 copernic.crdp.ac-aix-marseille.fr. A      194.254.139.4
...
 www.ac-aix-marseille.fr. CNAME  copernic.crdp.ac-aix-marseille.fr
...
```

Cette commande permet de lister tous les enregistrements d'une zone (domaine ou partie d'un domaine, comme nous le verrons dans la construction d'un DNS).
Le listing a été volontairement tronqué, d'abord parce qu'il est assez long, ensuite parce qu'il ne présente pas un grand intérêt dans notre propos.

On retrouve bien dans cette liste les informations qui nous intéressaient concernant le serveur web de l'académie.

Notez également les champs qui apparaissent :

- **SOA** (Start Of Authority) Ce champ indique que le serveur concerné (dnse.ac-aix-marseille.fr) est le responsable de la zone ac-aix-marseille.fr (la référence suprême en quelque sorte).
- **A** (Address) indique l'adresse IP correspondant à un nom d'hôte dans la zone.
- **NS** (Name Server) indique que l'hôte en question est un serveur de noms. On retrouve bien les deux serveurs de noms déjà vus plus haut.
- **CNAME** (Canonical Name) C'est la définition d'un alias. La partie de droite n'est d'ailleurs pas une adresse, mais un nom d'hôte déjà défini dans un enregistrement de type A.

Il existe également d'autres types d'enregistrements, comme **MX** qui renseigne sur les serveurs smtp, mais ceci est une autre histoire[2]...

Les mêmes informations auraient pu être extraites avec la commande :

```
host -l ac-aix-marseille.fr 195.83.252.200
```

## Conclusions

- Une recherche toute simple, à partir du serveur de noms par défaut nous donne immédiatement la solution.
- Par ailleurs, en recherchant à partir d'un root-server quels sont les DNS capables d'informer sur le TLD "fr.", puis, à partir de l'un de ces serveurs, quels sont ceux qui peuvent nous informer sur "ac-aix-marseille.fr." et finalement, à partir de l'un de ces serveurs , quelle est l'adresse de l'hôte "www.ac-aix-marseille.fr.", nous avons aussi trouvé la solution...

Dans le second cas, nous avons effectué "à la main" une recherche "itérative", par approches successives, pour aboutir à la solution.

Dans le premier cas, notre DNS par défaut a effectué une recherche analogue de façon transparente pour nous, et nous a servi la réponse "sur un plateau".

Comme c'est tout de même son travail de servir des adresses correspondant aux noms, il garde sa recherche en mémoire au cas où... Et c'est la raison du fameux "Réponse de source secondaire:" !

2 SMTP : http://christian.caleca.free.fr/smtp/index.html

Lorsqu'on lui repose la question une seconde fois, il ressort le résultat qu'il a gardé en cache, s'il estime qu'il est encore d'actualité (mais ceci est encore une autre question que nous verrons plus loin). En fait, comme nous le verrons dans le "sniff", il en profite pour garder en cache toutes les informations qu'il a trouvées dans sa recherche concernant les serveurs de noms intermédiaires, ce qui pourra lui faire gagner du temps si par exemple on lui demandait de résoudre un autre nom du TLD fr. (www.wanadoo.fr par exemple).

Notre DNS par défaut est dit "récursif", c'est à dire qu'il sait faire tout seul une recherche comme nous l'avons menée manuellement. Tous les DNS ne le sont pas et principalement les root-servers.

# Dans l'autre sens...

Il est parfois utile de pouvoir trouver le nom d'un hôte si l'on ne connaît que son adresse IP, bien que l'internaute moyen n'ait normalement pas besoin de cette fonction.

Les DNS savent le faire et une telle recherche peut s'effectuer à la main de façon similaire à une recherche normale, en partant des root-servers.

Prenons un exemple qui marche bien: la recherche du nom correspondant à l'adresse 193.252.19.142 (prise pas du tout au hasard). Dans la pratique, la manip est parfois un peu plus compliquée parce que les serveurs ne renvoient pas toujours les adresses IP avec les noms, mais la démarche est bonne et fonctionne dans tous les cas, elle peut simplement être parfois beaucoup plus longue.

| | |
|---|---|
| `D:\>nslookup`<br>`Serveur par défaut : gateway1.maison.mrs`<br>`Address: 192.168.0.250` | la routine... |
| `> server a.root-servers.net`<br>`Serveur par défaut : a.root-servers.net`<br>`Address: 198.41.0.4` | un root server... |
| `> set q=ptr` | ici, on va demander les enregistrements de type **PTR**. Nous verrons en détail dans la construction d'un DNS à quoi ils correspondent |
| `> 193.252.19.142` | Je pose la question... |
| `Serveur: a.root-servers.net`<br>`Address: 198.41.0.4`<br><br>`193.IN-ADDR.ARPA nameserver = NS.RIPE.NET`<br>`193.IN-ADDR.ARPA nameserver = NS.EU.NET`<br>`193.IN-ADDR.ARPA nameserver = AUTH03.NS.UU.NET`<br>`193.IN-ADDR.ARPA nameserver = NS2.NIC.FR`<br>`193.IN-ADDR.ARPA nameserver = SUNIC.SUNET.SE`<br>`193.IN-ADDR.ARPA nameserver = MUNNARI.OZ.AU`<br>`193.IN-ADDR.ARPA nameserver = NS.APNIC.NET`<br>`NS.RIPE.NET internet address = 193.0.0.193`<br>`NS.EU.NET internet address = 192.16.202.11`<br>`AUTH03.NS.UU.NET internet address = 198.6.1.83`<br>`NS2.NIC.FR internet address = 192.93.0.4`<br>`SUNIC.SUNET.SE internet address = 192.36.125.2`<br>`MUNNARI.OZ.AU internet address = 128.250.1.21`<br>`NS.APNIC.NET internet address = 203.37.255.97` | Et le root server, comme à son habitude, ne me répond pas directement, mais m'envoie une liste de serveurs qui savent résoudre les adresses qui commencent par 193. |
| `> server 192.93.0.4`<br>`Serveur par défaut : ns2.nic.fr`<br>`Address: 192.93.0.4` | J'en choisis un, au hasard... |

```
> 193.252.19.142
Serveur: ns2.nic.fr
Address: 192.93.0.4

252.193.in-addr.arpa nameserver = bow.rain.fr
252.193.in-addr.arpa nameserver = proof.rain.fr
252.193.in-addr.arpa nameserver = ns.ripe.net
252.193.in-addr.arpa nameserver = ns.global-ip.net
bow.rain.fr internet address = 194.51.3.49
proof.rain.fr internet address = 194.51.3.65
ns.ripe.net internet address = 193.0.0.193
ns.ripe.net AAAA IPv6 address = 0:0:0:0:0:ffff:c100:c1
ns.global-ip.net internet address = 194.52.1.10
```

Et je lui repose la même question. Il me répond bien poliment par une liste de serveurs qui savent résoudre les adresses commençant par 193.252.
Notez qu'en recherche inverse, les adresses sont classées à l'envers. C'est logique, on part du poids le plus fort vers le poids le plus faible.

```
> server 194.51.3.49
51.194.in-addr.arpa nameserver = bow.rain.fr
51.194.in-addr.arpa nameserver = proof.rain.fr
51.194.in-addr.arpa nameserver = ns.ripe.net
51.194.in-addr.arpa nameserver = ns.global-ip.net
51.194.in-addr.arpa nameserver = ns3.nic.fr
bow.rain.fr internet address = 194.51.3.49
proof.rain.fr internet address = 194.51.3.65
ns.ripe.net internet address = 193.0.0.193
ns.ripe.net AAAA IPv6 address = 0:0:0:0:0:ffff:c100:c1
ns.global-ip.net internet address = 194.52.1.10
ns3.nic.fr internet address = 192.134.0.49
Serveur par défaut : [194.51.3.49]
Address: 194.51.3.49
```

Je prends le premier serveur proposé.
Notez que celui-ci, sans que je ne lui demande rien, m'envoie la liste des serveurs qui connaissent les adresses commençant par 194.51 (comme la sienne). Je n'en ai rien à faire ici, mais je peux éventuellement garder cette info dans mes tablettes...

```
> 193.252.19.142
Serveur: [194.51.3.49]
Address: 194.51.3.49

19.252.193.in-addr.arpa nameserver = ns.wanadoo.fr
19.252.193.in-addr.arpa nameserver = ns.wanadoo.com
19.252.193.in-addr.arpa nameserver = ns2.wanadoo.fr
19.252.193.in-addr.arpa nameserver = ns2.wanadoo.com
ns.wanadoo.fr internet address = 193.252.19.10
ns.wanadoo.com internet address = 194.51.238.1
ns2.wanadoo.fr internet address = 193.252.19.11
ns2.wanadoo.com internet address = 194.51.238.2
```

Je pose à ce serveur toujours la même question, il me répond cette fois-ci par la liste des serveurs connaissant les adresses qui commencent par 193.252.19.
(on approche)
A ce stade, on a déjà une idée du domaine dans lequel se trouve notre hôte.

```
> server 194.51.238.2
Serveur par défaut : ns2.wanadoo.com
Address: 194.51.238.2
```

Pour changer un peu, je choisis le dernier de la liste (bienvenue dans wanadoo.com)

```
> 193.252.19.142
Serveur: ns2.wanadoo.com
Address: 194.51.238.2

142.19.252.193.in-addr.arpa name = www.wanadoo.fr
19.252.193.in-addr.arpa nameserver = ns.wanadoo.fr
19.252.193.in-addr.arpa nameserver = ns.wanadoo.com
19.252.193.in-addr.arpa nameserver = ns2.wanadoo.fr
19.252.193.in-addr.arpa nameserver = ns2.wanadoo.com
ns.wanadoo.fr internet address = 193.252.19.10
ns.wanadoo.com internet address = 194.51.238.1
ns2.wanadoo.fr internet address = 193.252.19.11
ns2.wanadoo.com internet address = 194.51.238.2
```

Encore une fois la question et j'obtiens la réponse définitive :
www.wanadoo.fr

Vous avez compris le principe, on ne va pas aussi le faire avec host.

## Conclusions

Une recherche inverse se mène donc de manière similaire à une recherche directe. Si j'avais posé directement la question à mon DNS favori, j'aurais bien entendu obtenu directement la réponse :

```
D:\>nslookup 193.252.19.142
Serveur: gateway1.maison.mrs
Address: 192.168.0.250

Nom : www.wanadoo.fr
Address: 193.252.19.142
```

C'eut été plus simple, mais je n'aurais pas compris comment ça marche...

Pour l'instant, ne nous posons pas trop de questions sur ce "in-addr.arpa" que nous verrons en détail lors de la construction de notre DNS. Il s'agit en fait d'une zone dans laquelle on enregistre les adresses avec le nom d'hôte qui correspond, ceci se faisant en même temps que l'on entre dans la zone "normale" les noms avec l'adresse qui correspond dans un champ de type "A".

Un document fort intéressant[3] et fort complet vous aidera à comprendre tout ça et même plus...

# Le "DNS Round-robin"

Comme c'est une technique qui se développe de plus en plus, autant en dire quelques mots...

Après tout ce que nous avons vu, nous avons probablement compris qu'à un nom correspond une IP et, éventuellement, par le jeu des alias, une IP peut être attribuée à plusieurs noms. Serait-il possible qu'à un nom donné puissent correspondre plusieurs IPs ?

```
~# host news.free.fr
news.free.fr is an alias for news.proxad.net.
news.proxad.net has address 213.228.0.133
news.proxad.net has address 213.228.0.136
news.proxad.net has address 213.228.0.138
news.proxad.net has address 213.228.0.196
news.proxad.net has address 213.228.0.4
news.proxad.net has address 213.228.0.32
news.proxad.net has address 213.228.0.33
news.proxad.net has address 213.228.0.75
```

Est-ce possible ? Pas moins de 8 adresses différentes pour le même nom ? Ce doit être une erreur, recommençons...

```
~# host news.free.fr
news.free.fr is an alias for news.proxad.net.
news.proxad.net has address 213.228.0.33
news.proxad.net has address 213.228.0.75
news.proxad.net has address 213.228.0.133
news.proxad.net has address 213.228.0.136
news.proxad.net has address 213.228.0.138
news.proxad.net has address 213.228.0.196
news.proxad.net has address 213.228.0.4
news.proxad.net has address 213.228.0.32
```

Ce n'est pas une erreur, nous avons bien ici huit IP différentes, et constatez qu'entre les deux interrogations, si les adresses n'ont pas changé, en revanche, l'ordre dans lequel elles ont été données, lui, a changé. En pratique, à chaque requête, le DNS opèrera une permutation circulaire.

Le client utilisera normalement la première réponse dans la liste, deux utilisateurs consécutifs utiliseront donc deux IPs différentes, pour accéder au même service. La charge en terme de flux de données sera naturellement répartie sur les diverses IPs.

A quoi cela sert-il ? Il peut y avoir plusieurs raisons :

- Une grosse machine, très puissante et qui peut fournir un flux de données très important, dispose de plusieurs interfaces réseau de manière à éliminer un goulet d'étranglement au niveau de chaque interface. Disons qu'un serveur pourrait par exemple assurer un flux continu de l'ordre du Giga Bits par seconde. Si son unique interface réseau est une 100 Méga bits par seconde, il y aura un goulet sur cette connexion.

3 http://www.freenix.fr/unix/linux/HOWTO/DNS-HOWTO-4.html

- La charge du service est trop importante pour un unique serveur. Plusieurs serveurs, configurés de façon identique, vont répondre aux requêtes des clients, la charge étant naturellement répartie entre ces serveurs.

Cette technique est de plus en plus employée. Autres exemples, pour ne pas faire de jaloux :

```
~# host smtp.free.fr
smtp.free.fr has address 213.228.0.169
smtp.free.fr has address 213.228.0.176
smtp.free.fr has address 213.228.0.44
smtp.free.fr has address 213.228.0.62
```

Et :

```
~# host smtp.wanadoo.fr
smtp.wanadoo.fr has address 193.252.22.72
smtp.wanadoo.fr has address 193.252.22.73
smtp.wanadoo.fr has address 193.252.22.74
smtp.wanadoo.fr has address 193.252.22.75
smtp.wanadoo.fr has address 193.252.22.76
smtp.wanadoo.fr has address 193.252.22.77
```

# Construire un DNS

## Pourquoi pas...

Vous pouvez souhaiter monter un DNS local pour plusieurs raisons :

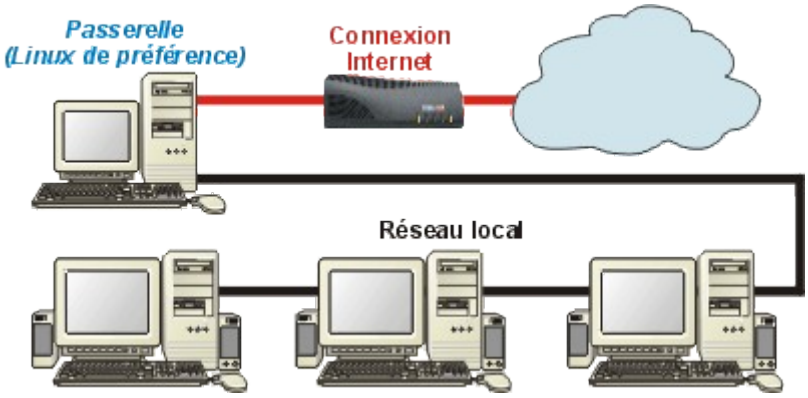
- Vous disposez sur votre LAN de services Intranet du genre serveur web, ftp, messagerie. Dans ce cas, il vous faut un système de résolution de noms.
- Vous ne voulez pas vous embêter avec la configuration des clients de votre LAN, vous préférez leur indiquer comme DNS votre propre système de résolution.
- Vous voulez juste vous amuser avec Bind, pour la beauté du geste.

Il existe un moyen rudimentaire de fournir à chaque hôte du réseau un système de résolution de noms à partir du fichier "hosts" (ne confondez pas avec la commande de Linux). Chaque hôte doit disposer d'un tel fichier, à jour. C'est relativement simple à faire si le fichier ne contient que quelques références et que les postes sont peu nombreux, c'est autre chose si le réseau local prend de l'ampleur. Et ça ne résoudra pas le problème de la résolution sur Internet.

Et puis, c'est tellement simple à monter, un DNS :)

## Un DNS ?

Fixons les esprits. Nous sommes dans la configuration suivante : celle qui intéresse la majorité des galériens du haut débit, le réseau local pouvant aller d'un à un certain nombre d'hôtes, 6 dans mon cas, chacun le (ou les) sien(s), c'est le meilleur moyen d'être tranquille...




Dans ce type de configuration, un DNS monté sur la passerelle Linux peut présenter un intérêt s'il sert à la fois :

- A résoudre les noms sur le réseau local, évitant ainsi d'avoir à maintenir des fichiers hosts sur chaque poste.
- A résoudre aussi les noms sur l'Internet, ce qui permet de configurer les clients du réseau local avec cet unique DNS pour toutes les résolutions de noms. Seul petit problème, les méthodes de travail de Wanadoo qui perturbent... Mais nous verrons comment détourner cet inconvénient.

Il est clair que si votre installation est réduite à un poste directement connecté au modem, construire un DNS dessus n'a plus aucune utilité.

# Mode opératoire

Dans un premier temps, nous allons construire un simple cache pour l'Internet. Ce DNS ne résoudra pas les noms des hôtes du réseau privé, mais il le fera pour les noms Internet.

Dans un second temps, nous ajouterons une zone qui permettra de résoudre les noms dans le réseau local. Il faudra alors déclarer un domaine "bidon", avec tous les risques que ça comporte, à cause d'éventuelles interférences avec un domaine déclaré d'Internet.

# Construction du cache

Nous allons utiliser BIND qui est fait pour çà. Nous partons également du principe que BIND a été installé correctement. Ce n'est pas une chose compliquée, toutes les distributions fournissent les paquetages nécessaires (bind et bind-utils pour Mandrake).

Le mode opératoire décrit ici est testé sur une distribution MANDRAKE 7.0, mais les versions suivantes le supportent également.

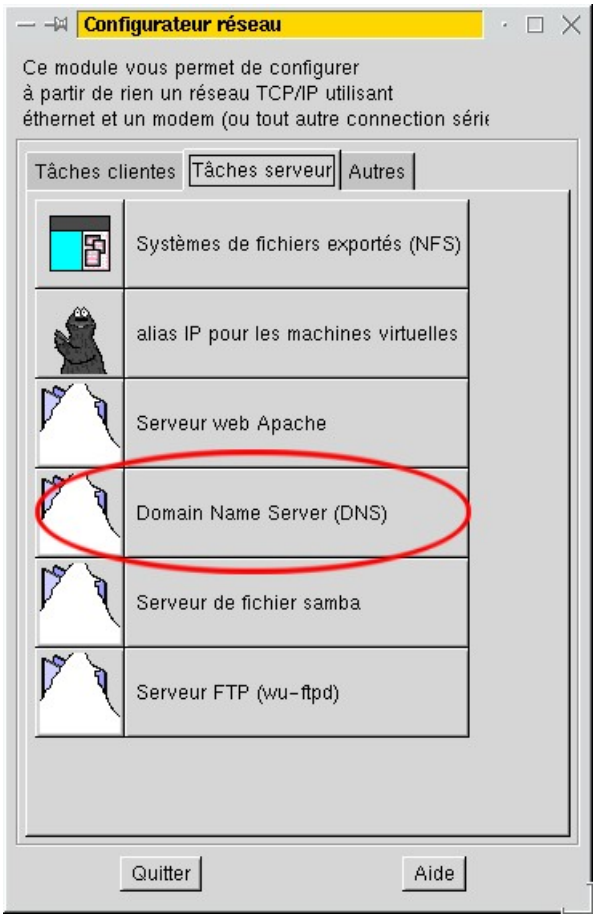
## Configuration avec linuxconf

linuxconf est fait pour RedHat. Il fonctionne sur Mandrake, mais vous risquez d'avoir quelques sueurs au moment de la mise à jour des configurations. En effet, des messages parfois incohérents risquent d'apparaître. Nous verrons alors ce qu'il y a lieu de faire.

Assurez-vous d'abord que linuxconf est correctement configuré. Le fichier /etc/conf.linuxconf doit contenir au paragraphe [base] la ligne :

```
module.list 1 dnsconf
```

Si ce n'est pas le cas, ajoutez-la et redémarrez linuxconf.

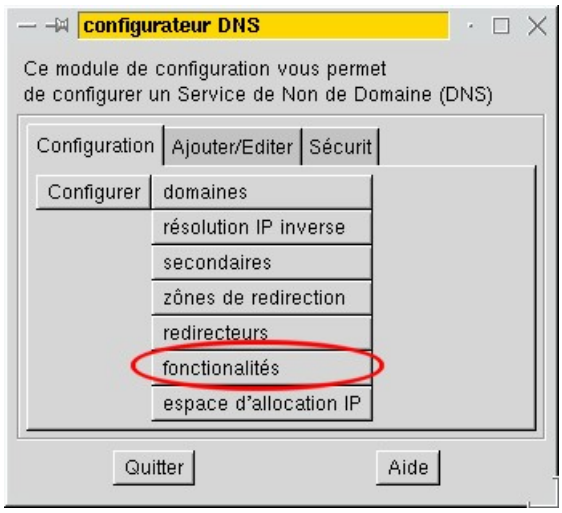



En mode graphique (à la rigueur en mode texte) démarrez linuxconf et choisissez le bouton "Réseau".

Sur l'onglet "Tâches serveur" doit apparaître la fonction "Domain Name Server". Cliquez dessus.

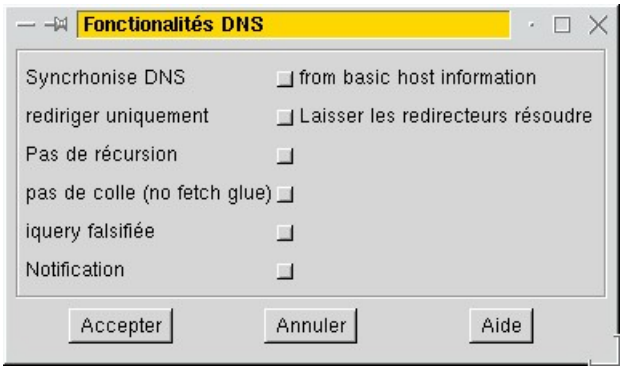
Elle n'y est pas ? assurez-vous que :

- BIND est bien installé.
  - Le fichier /etc/conf.linuxconf contient la ligne : "module.list 1 dnsconf" dans le paragraphe [base]. Cette ligne peut ne pas être présente si vous avez installé bind après linuxconf. Il suffit alors de l'ajouter à la main.



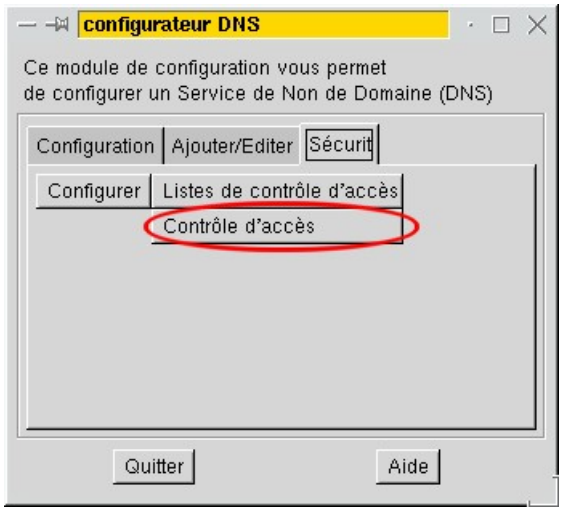

Pour l'instant, nous construisons juste un cache.

Seul le bouton "fonctionnalités" nous intéresse.




Ne cédez pas à la tentation...

Aucun bouton n'a besoin d'être enfoncé.




Nous allons tout de suite régler quelques problèmes de sécurité.

Choisissez l'onglet du même nom, cliquez sur "Contrôle d'accès"...

Nous supposons que le réseau privé est de la classe 192.168.0.0. Nous indiquons ici que les transferts ne seront autorisés que pour les machines dont l'IP est de cette classe.

Même chose pour Autoriser les requêtes...

Quittez linuxconf.

C'est ici que de curieux messages risquent d'apparaître. Si linuxconf vous signale que certains services ne tournent pas et qu'il faut les démarrer, à priori, c'est faux. Vous n'avez manipulé que Bind et il n'y a aucune raison que le reste du système ait été altéré. Vérifiez à la main si le coeur vous en dit... (ps aux, par exemple). Dans un tel cas, dites à linuxconf de ne rien faire, puis rechargez named avec un :

```
/etc/init.d/named reload
```

## Autres moyens

Linuxconf n'est pas forcément très pratique à manipuler, principalement sur les distributions Mandrake. De plus, cet outil semble abandonné dans les dernières distributions. Essayez plutôt :

- Webmin[4], un utilitaire à tout faire via HTTP (ou, encore mieux, HTTPS). Il dispose d'un module pour Bind 8 qui suffit à réaliser ce que l'on veut faire ici, même si nous utilisons un bind 9. Webmin est suffisamment intuitif.
- De configurer les fichiers de bind avec un éditeur de texte. Ce n'est pas très compliqué, ça l'est tout de même un peu plus que d'utiliser linuxconf ou webmin, à cause de la syntaxe très stricte à utiliser.

## Contrôle des fichiers de configuration

Voilà. C'est presque fini, juste quelques vérifications. Pour les faire, on va mettre les mains dans le cambouis, (entendez par là que l'on va aller voir dans les fichiers de configuration)...

BIND utilise quelques fichiers de configuration en mode texte, qui sont mis à jour par linuxconf (ou Webmin).

### named.conf

Ce fichier est le premier que BIND va lire. Nous y trouvons déjà des informations intéressantes...

```
options {
        directory "/var/named";
        notify no;
        allow-transfer{
                192.168.0.0/24;
        };
        allow-query{
                192.168.0.0/24;
        };
};
```

Au paragraphe "options", on trouve :

- L'emplacement des autres fichiers de configuration. Ici, ce sera le répertoire /var/named.
- Notify no indique que ce serveur travaillera pour son compte. Cette option est utile lorsque plusieurs DNS doivent se synchroniser entre eux. Ce ne sera pas notre cas.
- On retrouve enfin les règles de sécurité que nous avions posées dans LINUXCONF.

```
zone "." {
        type hint;
        file "root.cache";
};
```

La zone "." est fondamentale pour ce que l'on veut réaliser. C'est elle qui permettra à BIND d'interroger les root-servers. Le fichier root.cache dont on a déjà parlé contient les adresses de ces root servers. Il doit être périodiquement mis à jour, nous verrons çà plus loin.

```
zone "0.0.127.IN-ADDR.ARPA"{
        type master;
        file "127.0.0";
};
```

Cette zone est une zone de recherche inverse pour les adresses de type 127.0.0. Ces adresses sont utilisées exclusivement en interne par la pile TCP/IP et correspondent à l'hôte "localhost"

### root.cache

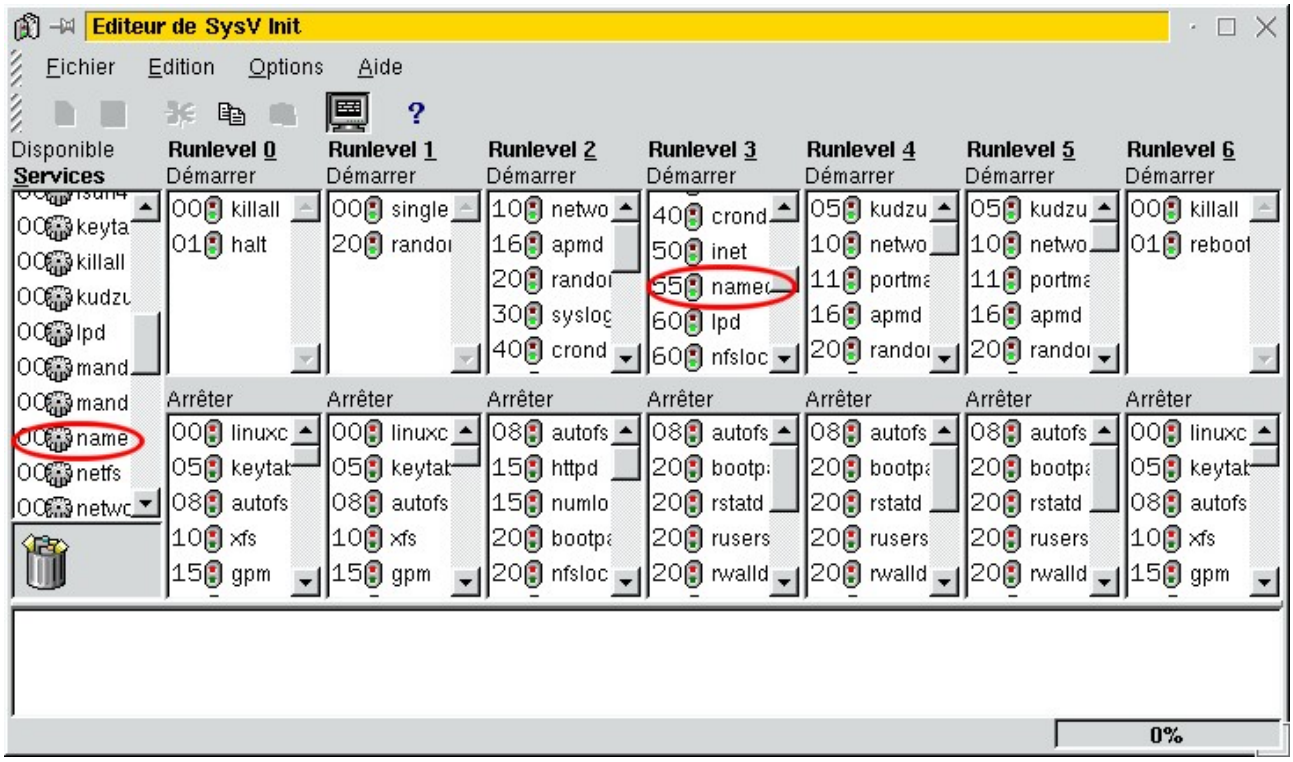
Si vous avez bien suivi, vous savez qu'il se trouve dans /var/named. Une version est installée avec BIND, vous avez tout de même intérêt à vous assurer qu'elle est à jour en allant voir à ftp://ftp.rs.internic.net/domain/named.root

vous pouvez télécharger ce fichier et le placer directement dans le répertoire /var/named après, bien entendu, avoir renommé votre copie de "root.cache", par exemple en "root.cache.old".

## Le daemon named

Vérifiez que le Daemon "named" est en service, mieux encore, vérifiez qu'il est lancé à chaque démarrage. Un outil pratique est l'éditeur de Sys V Init. La photo est un peu grande mais elle vaut le coup d'œil :

4 Webmin : http://www.webmin.com/




Les puristes me pardonneront, c'est certainement un outil qu'ils n'aiment pas... Mais je ne suis pas un puriste LINUX (ça viendra peut-être un jour). Si vous ne disposez pas de cet outil, pas de panique, il y en a d'autres comme tksysv par exemple.

Vérifiez d'une part que "named" figure dans la colonne "services", ce doit être le cas si BIND est correctement installé.

Vérifiez ensuite qu'il figure également au moins dans la colonne du "runlevel" que vous utilisez (3 par défaut) au chapitre "Démarrer". Si ce n'est pas le cas, mettez-le (drag and drop), sauvez le fichier et ça ira mieux au prochain démarrage. En attendant, vous pouvez le démarrer manuellement, soit depuis l'outil graphique, soit par la ligne de commande :

```
/etc/rc.d/init.d/named start
```

Un autre moyen pour effectuer la vérification en mode texte, est d'utiliser une commande qui doit fonctionner si BIND est installé, c'est la commande "ndc". en étant "root", tapez dans une console "ndc help", mieux encore (ou en complément), tapez "man ndc" et voyez ce que vous pouvez faire avec. Entre autres choses, vous pouvez démarrer BIND avec la commande "ndc start".

### Est-ce que ça marche ?

Il suffit, sur l'un de vos postes clients du réseau privé, de configurer le DNS par défaut avec l'adresse de votre DNS et d'effectuer une résolution de nom sur Internet. Normalement, ça doit fonctionner.

## La galère Wanadoo

Tous les abonnés Wanadoo y ont droit.

**Identification du problème.**

Wanadoo propose à ses abonnés deux DNS qui sont 193.252.19.3 et 193.252.19.4. Mais ces DNS ne sont pas publics. Si, au moyen de nslookup (ou autre), vous recherchez les DNS publics du domaine wanadoo.fr, vous trouvez par exemple :

```
ns.wanadoo.fr internet address = 193.252.19.10
ns2.wanadoo.fr internet address = 193.252.19.11
ns.wanadoo.com internet address = 194.51.238.1
ns2.wanadoo.com internet address = 194.51.238.2
```

Il se trouve que les DNS des abonnés wanadoo ne donnent pas les mêmes adresses que les DNS publics pour, au moins, les serveurs smtp.wanadoo.fr et news.wanadoo.fr. Pourquoi? Je n'en sais fichtre rien...

Bien entendu, les adresses fournies par les DNS publics ne sont pas utilisables par les abonnés. La conclusion de tout ceci est que votre magnifique cache ne vous permettra pas :

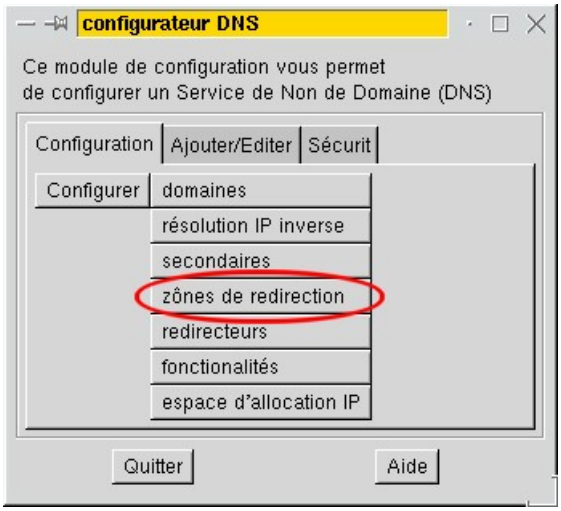
- Ni d'envoyer votre courrier par smtp.wanadoo.fr
- Ni de vous connecter aux news par news.wanadoo.fr

(Cette particularité m'a fait chercher longtemps...)

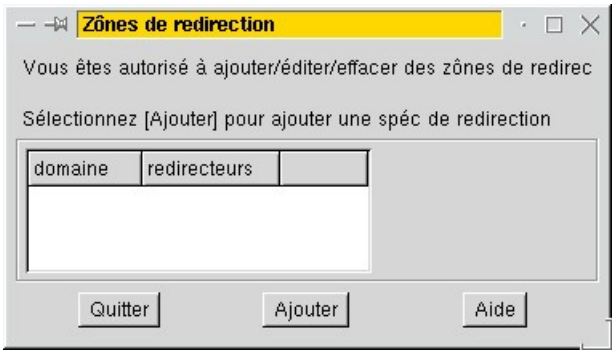
Depuis (courant avril/mai 2003), Wanadoo a remanié considérablement ses structures? Aujourd'hui (1 juin 2003), vous n'obtiendrez pas tout à fait les mêmes choses, mais la curiosité décrite plus haut est toujours visible.

**Contournement du problème : Ajouter une zone de redirection.**

Nous allons créer sur notre serveur de noms une zone "wanadoo.fr" pour laquelle nous transmettrons toutes les requêtes au DNS de Wanadoo Câble. Ce type de zone est appelé: "zone de redirection". Voici l'opération :




Après avoir démarré linuxconf, Tâches, serveur, DNS; choisissez le bouton "zones de redirection".




Cliquez sur "Ajouter"...

Et pour le domaine "wanadoo.fr", on va rediriger les requêtes sur le serveur de notre irremplaçable FAI **(L'adresse du vôtre n'est certainement pas celle qui est écrite ici).**

Que va-t-il se passer maintenant ? A chaque requête concernant le domaine "wanadoo.fr", votre DNS, au lieu d'effectuer une recherche récursive à partir des root-servers comme on l'a vu par ailleurs, transmettra directement au DNS du FAI qui, lui, répondra avec la bonne adresse pour nous. C'est la solution la moins sale que j'ai trouvée, mais nous restons tributaires de Wanadoo pour son domaine.

Pour ceux qui souhaitent utiliser le service DNS de Windows NT ou 2000, tout ce qui est dit dans ce chapitre est réalisable à l'exception de la redirection d'une zone (du moins à ma connaissance).

### Contrôle

Voyons maintenant l'allure de notre fichier "named.conf" :

```
options {
        directory "/var/named";
        notify no;
        allow-transfer{
                192.168.0.0/24;
        };
        allow-query{
                192.168.0.0/24;
        };
};
zone "." {
        type hint;
        file "root.cache";
};
zone "0.0.127.IN-ADDR.ARPA"{
        type master;
        file "127.0.0";
};
```

Cette partie n'a pas changé...

```
zone "wanadoo.fr."{
        type forward;
        forward only;
        forwarders{
                62.161.120.11;
        };
};
```

Mais l'on retrouve ici la zone de redirection.

# Ajouter une zone d'autorité

## Qu'est-ce que c'est ?

Une zone d'autorité est un domaine ou un sous-domaine que le DNS sait traiter avec sa propre base de données. Nous pouvons en créer une pour résoudre les noms des hôtes de notre réseau privé. Ceci ne présente, encore une fois, pas un intérêt capital mais tant qu'on y est, pourquoi nous en priver? (surtout si vous montez par exemple un petit serveur apache pour réaliser votre intranet :-)

## Préparation du travail

Vous avez en tête, bien entendu, l' adresse de chacun de vos postes privés ainsi que leur nom "NetBIOS", si ces postes sont sous Windows. Par ailleurs, vous connaissez également l'adresse et le nom de votre passerelle Linux.

Fixons les idées par un exemple :

- Quatre hôtes Windows :

| | |
|---|---|
| chris | 192.168.0.100 |
| daniel | 192.168.0.101 |
| michele | 192.168.0.102 |
| remi | 192.168.0.103 |

- Un serveur Linux (Pas la passerelle). Il pourra par exemple héberger votre site Intranet et partager de l'espace disque pour vos hôtes Windows avec SAMBA

| | |
|---|---|
| Server1 | 192.168.0.200 |

- La passerelle Linux qui supporte le DNS. Ce poste dispose de deux interfaces réseau, nous nous intéressons ici à celle qui est connectée au réseau privé. (L'autre recevant une adresse IP dynamique par le DNS de Wanadoo Câble).

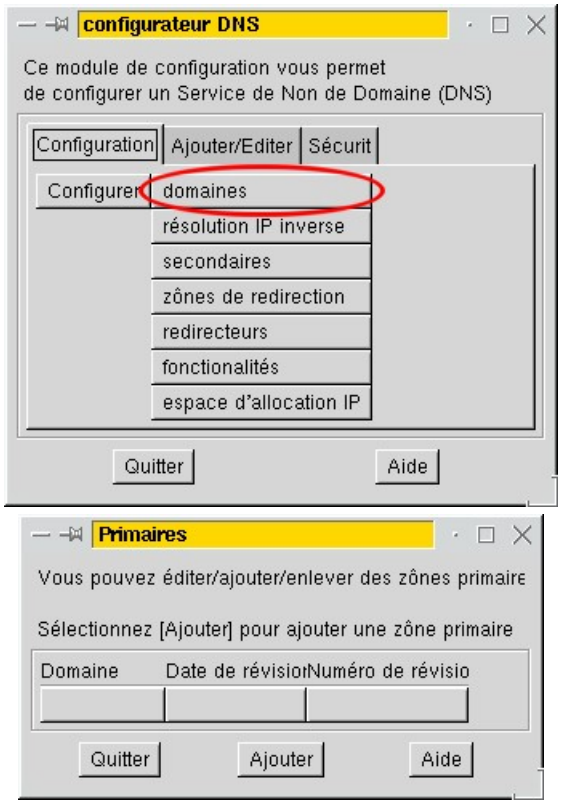
| | |
|---|---|
| gateway1 | 192.168.0.250 |

Nous allons nous choisir un joli nom de domaine :

- Soit on va choisir un nom dans un TLD existant, comme "maison.fr", mais ce nom est déposé. Le fait de le choisir pour votre domaine privé vous interdira d'aller visiter le "vrai" site www.maison.fr parce que votre DNS n'ira pas chercher plus loin que dans sa base tout hôte appartenant au domaine maison.fr. Si vous choisissez cette option, allez d'abord visiter le site "officiel" et voyez si ça peut vous poser un problème de ne plus pouvoir y avoir accès... Vous pouvez bien entendu choisir un autre nom, mais le problème risque de se reproduire ailleurs.
- Vous pouvez également pour la circonstance créer un TLD "en bois" comme mrs. par exemple. C'est pas tous les jours que de "vrais" TLD sont créés et de cette manière, vous risquez moins d'accidents.

Pour notre exemple, nous allons choisir maison.mrs.

## Création de la zone




Après avoir démarré linuxconf, option "réseau", onglet "Tâches serveurs", bouton DNS, on clique sur le bouton "domaines"...

Pour l'instant, il n'y en a pas, nous allons en ajouter un...

Comme c'est indiqué sur l'illustration. L'image a été coupée parce que la partie de droite ne nous intéresse pas.

Vérifiez sur l'onglet" serveurs de mail" qu'il n'y a aucune indication écrite, il ne nous sert à rien ici.

Une fois les champs renseignés, acceptez.

Maintenant, nous allons éditer la liste des hôtes pour le domaine...

Il n'y en a qu'un, celui que l'on vient de créer.

Ajoutons...

Le nom de la machine...

Son adresse, sur l'onglet "adrs.IP" et acceptons.

Il faut recommencer cette opération pour tous les hôtes de votre domaine privé.

Une fois l'hôte "server1" saisi, il sera même possible de créer un alias en ajoutant un enregistrement comme suit:

Et voilà le travail. Normalement, votre zone privée est définie.

## Vérifications

### Les fichiers de configuration

#### /etc/named.conf

```
options {
        directory "/var/named";
        notify no;
        allow-transfer{
                192.168.0.0/24;
        };
        allow-query{
                192.168.0.0/24;
        };
};
zone "." {
        type hint;
        file "root.cache";
};
zone "0.0.127.IN-ADDR.ARPA"{
        type master;
        file "127.0.0";
};
zone "wanadoo.fr"{
        type forward;
        forwarders{
                62.161.120.11;
        };
};
```

Ici, il n'y a rien de changé...

```
zone "maison.mrs"{
        type master;
        file "maison.mrs";
};
zone "0.168.192.IN-ADDR.ARPA"{
        type master;
        file "192.168.0";
};
```

Ces deux zones ont été ajoutées, ce qui veut dire que l'on va trouver deux nouveaux fichiers dans le répertoire /var/named

- maison.mrs pour la résolution des noms
- 192.168.0 pour la résolution inverse

#### /var/maison.mrs

```
@               IN      SOA     gateway1.maison.mrs.    hostmaster.gateway1.maison.mrs. (
                        2000042701 ; serial
                        3600 ; refresh
                        900 ; retry
                        1209600 ; expire
                        43200 ; default_ttl
                        )
```

```
gateway1        IN      A       192.168.0.250
chris           IN      A       192.168.0.100
daniel          IN      A       192.168.0.101
remi            IN      A       192.168.0.103
michele         IN      A       192.168.0.102
server1         IN      A       192.168.0.200
www             IN      CNAME   server1.maison.mrs.
@               IN      NS      gateway1.maison.mrs.
```

Le @ signifie qu'il est fait référence au serveur lui-même.

Notez le Start Of Authority (SOA) et le CNAME www de server1

### /var/192.168.0

```
@               IN      SOA     gateway1.maison.mrs.    2000042701 (
                        2000042702 ; serial
                        0 ; refresh
                        0 ; retry
                        0 ; expire
                        0 ; default_ttl
                        )
100             IN      PTR     chris.maison.mrs.
101             IN      PTR     daniel.maison.mrs.
103             IN      PTR     remi.maison.mrs.
102             IN      PTR     michele.maison.mrs.
@               IN      NS      gateway1.maison.mrs.
250             IN      PTR     gateway1.maison.mrs.
200             IN      PTR     server1.maison.mrs.
```

Pas grand chose à dire, de plus, si ce n'est que seul le dernier octet de l'adresse est signalé, c'est normal, nous sommes dans une classe C, les trois autres octets sont ceux du réseau.

#### Un coup de NSLOOKUP

Depuis un hôte quelconque du réseau:

```
D:\>nslookup chris.maison.mrs
Serveur: gateway1.maison.mrs
Address: 192.168.0.250

Nom : chris.maison.mrs
Address: 192.168.0.100
```

Si ça marche pour un, ça doit marcher pour tous. Le travail est terminé.

# Conclusion

Le DNS que nous avons construit dispose de quelques fonctionnalités intéressantes :

- Il résout les noms du réseau privé, comme le fait un DNS d'entreprise.
- Il contourne le problème FTI en transmettant les requêtes sur le domaine wanadoo.fr et seulement sur lui, au DNS de notre FAI.
- Il résout les noms de tous les autres domaines en effectuant des recherches itératives sur les serveurs de noms publics, comme le fait tout bon DNS.

Pour ceux qui voudraient en savoir un peu plus, il y a le DNS-HOWTO[5] en français, dont je me suis

5 DNS-HOWTO : http://www.freenix.org/unix/linux/HOWTO/DNS-HOWTO.html

largement inspiré pour mener cette étude. A la fin ce ce HOWTO, il y a une série de liens sur des sites plus pointus, mais ils ne sont pas en français...

Comme nous pouvons le constater, le mécanisme de la résolution des noms est très logique, mais nécessite une organisation sans failles.

Beaucoup de choses n'ont pas été dites ici, comme la notion de serveur maître et de serveur esclave, le système des notifications qui permet, lorsque l'on met à jour un DNS de répercuter cette mise à jour sur d'autres serveurs etc. Mais le but de ce chapitre était plus de décortiquer le fonctionnement que de donner un cours d'administration d'inter réseaux.

Ce qui a été développé ici vous aura aidé, du moins je l'espère :

- à comprendre comment un DNS récursif travaille
- à voir la différence entre un serveur récursif (celui que l'on a construit en exemple) et un serveur qui ne l'est pas (la plupart des root-servers et des serveurs des NICS)
- à comprendre le rôle des principaux enregistrements SOA, NS, A, CNAME, PTR
- à comprendre enfin pourquoi il il faut être prudent dans le choix de son DNS par défaut

## Liens divers

| | |
|---|---|
| Le jargon français | Un site bien utile pour avoir des définitions des divers acronymes et mots spécialisés du jargon informatique , avec explications en français<br>http://www.linux-france.org/prj/jargonf/ |
| DNS HOWTO | Un document en français qui traite de la construction d'un DNS avec BIND sous LINUX<br>http://www.freenix.org/unix/linux/HOWTO/DNS-HOWTO.html |
| L'InterNIC | INTERnet Network Information Center<br>http://www.internic.net/ |
| Le NIC France | L'organisme qui a en charge la gestion du TLD fr. Vous y trouverez (en français) pas mal d'information concernant les règles de nommage et les DNS.<br>http://www.nic.fr/ |
| www.google.fr | Un bon moteur de recherches où en trouver d'autres par vous-mêmes :-)<br>http://www.google.fr/ |

# Préceptes

L'objectif de ce chapitre n'est pas de vous inviter à jouer les "apprentis sorciers", mais de comprendre comment fonctionne le service des noms sur l'Internet.

Autrement dit, ne manipulez pas n'importe quoi n'importe comment sans être certain de la pertinence de ce que vous faites.

## Quel DNS choisir par défaut ?

Le mieux est évidemment de choisir celui que vous fournit votre fournisseur d'accès, pour de multiples raisons dont en voici quelques-unes :

- Il est normalement dimensionné convenablement en fonction du nombre potentiel de requêtes des abonnés au service.
- Il est en principe construit par un spécialiste de la question (ce qui n'est pas mon cas).
- Il est généralement optimisé pour répondre rapidement à vos requêtes, il est placé sur un réseau proche du votre, voire le même, ce qui évite le passage de plusieurs routeurs et optimise son temps de réponse.
- Il peut, comme nous l'avons vu pour Wanadoo, vous donner des informations que vous n'obtiendrez pas par ailleurs (c'est pas propre, mais c'est comme çà).

## Et si moi je n'veux pas ?

Dans ce cas, montez votre propre DNS, mais **évitez absolument** d'en choisir un au hasard, par exemple parce qu'un ami vous a dit du bien de celui de son FAI. Ça ne vous avancera à RIEN, sauf éventuellement à vous dépanner temporairement, si le DNS de votre FAI tombe...

# sniff

## Il n'y a que ça de vrai...

Rien de tel que l'analyse de ce qu'il se passe sur le réseau pour comprendre les choses. La manip. proposée est la suivante :

Rappelons qu'un "sniffer" est un outil capable de capturer toutes les trames qui passent sur le réseau où est connectée l'interface choisie. Ces trames sont bien entendu capturées en binaire, peuvent être affichées en mode hexadécimal, mais le plus intéressant, c'est qu'un bon "sniffer" est capable de les interpréter et de traduire en un langage presque compréhensible (l'anglais) leur contenu. C'est sous cette forme que la capture est présentée ici.

La machine Linux qui sert de DNS tout neuf (remis à zéro) va être interrogée par un hôte du réseau privé pour trouver successivement les adresses de :

- www.ac-aix-marseille.fr
- www.voila.fr

Le but espéré est de montrer :

- Que le serveur récursif fonctionne bien comme prévu
- Mettre en évidence le rôle du cache

Nous allons voir que la manipulation atteint son but.

### Recherche du premier hôte

#### Première requête

```
Frame 3 (83 on wire, 83 captured)
...
    Protocol: UDP (0x11)
    Header checksum: 0xe655 (correct)
    Source: ca-ol-marseille-12-195.abo.wanadoo.fr (213.56.59.195)
    Destination: h.root-servers.net (128.63.2.53)
       ***C'est bien un root-server qui est contacté
User Datagram Protocol
    Source port: 1029 (1029)
    Destination port: domain (53)
    Length: 49
    Checksum: 0xfa3a
Domain Name System (query)
    Transaction ID: 0x2673
    Flags: 0x0000 (Standard query)
        0... .... .... .... = Query
        .000 0... .... .... = Standard query
        .... ..0. .... .... = Message is not truncated
        .... ...0 .... .... = Don't do query recursively
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
        www.ac-aix-marseille.fr: type A, class inet
            Name: www.ac-aix-marseille.fr
            Type: Host address
            Class: inet
```

```
        *** Il attaque directement avec la question finale...
```

## Première réponse

```
Frame 4 (403 on wire, 403 captured)
...
    Protocol: UDP (0x11)
    Header checksum: 0x51e4 (correct)
    Source: h.root-servers.net (128.63.2.53)
    Destination: ca-ol-marseille-12-195.abo.wanadoo.fr (213.56.59.195)
User Datagram Protocol
    Source port: domain (53)
    Destination port: 1029 (1029)
    Length: 369
    Checksum: 0x8662
Domain Name System (response)
    Transaction ID: 0x2673
    Flags: 0x8000 (Standard query response, No error)
        1... .... .... .... = Response
        .000 0... .... .... = Standard query
        .... .0.. .... .... = Server isn't an authority for domain
        .... ..0. .... .... = Message is not truncated
        .... ...0 .... .... = Don't do query recursively
        .... .... 0... .... = Server can't do recursive queries
        .... .... .... 0000 = No error
    Questions: 1
    Answer RRs: 0
    Authority RRs: 8
    Additional RRs: 8
    Queries
        www.ac-aix-marseille.fr: type A, class inet
            Name: www.ac-aix-marseille.fr
            Type: Host address
            Class: inet
    Authoritative nameservers
        FR: type NS, class inet, ns DNS.CS.WISC.EDU
            Name: FR
            Type: Authoritative name server
            Class: inet
            Time to live: 2 days
            Data length: 17
            Name server: DNS.CS.WISC.EDU
        FR: type NS, class inet, ns NS1.NIC.FR
            Name: FR
            Type: Authoritative name server
            Class: inet
            Time to live: 2 days
            Data length: 10
            Name server: NS1.NIC.FR
        FR: type NS, class inet, ns NS3.NIC.FR
            Name: FR
            Type: Authoritative name server
            Class: inet
            Time to live: 2 days
            Data length: 6
            Name server: NS3.NIC.FR
        FR: type NS, class inet, ns DNS.INRIA.FR
            Name: FR
            Type: Authoritative name server
            Class: inet
            Time to live: 2 days
            Data length: 12
            Name server: DNS.INRIA.FR
        FR: type NS, class inet, ns NS2.NIC.FR
            Name: FR
            Type: Authoritative name server
            Class: inet
            Time to live: 2 days
            Data length: 6
            Name server: NS2.NIC.FR
```

```
        FR: type NS, class inet, ns NS.EU.NET
            Name: FR
            Type: Authoritative name server
            Class: inet
            Time to live: 2 days
            Data length: 11
            Name server: NS.EU.NET
        FR: type NS, class inet, ns DNS.PRINCETON.EDU
            Name: FR
            Type: Authoritative name server
            Class: inet
            Time to live: 2 days
            Data length: 16
            Name server: DNS.PRINCETON.EDU
        FR: type NS, class inet, ns NS-EXT.VIX.COM
            Name: FR
            Type: Authoritative name server
            Class: inet
            Time to live: 2 days
            Data length: 16
            Name server: NS-EXT.VIX.COM
     *** Bien entendu, il ne connaissait pas la réponse, mail il a donné une liste
     *** De serveurs qui connaissent le TLD fr.
     *** En prime, il nous donne leurs adresses.
    Additional records
...
        NS1.NIC.FR: type A, class inet, addr 192.93.0.1
...
        NS3.NIC.FR: type A, class inet, addr 192.134.0.49
...
        DNS.INRIA.FR: type A, class inet, addr 193.51.208.13
...
        NS2.NIC.FR: type A, class inet, addr 192.93.0.4
...
        NS.EU.NET: type A, class inet, addr 192.16.202.11
...
        DNS.PRINCETON.EDU: type A, class inet, addr 128.112.129.15
...
        NS-EXT.VIX.COM: type A, class inet, addr 204.152.184.64
...
```

## Deuxième requête

```
Frame 5 (83 on wire, 83 captured)
...
    Protocol: UDP (0x11)
    Header checksum: 0xe3ef (correct)
    Source: ca-ol-marseille-12-195.abo.wanadoo.fr (213.56.59.195)
    Destination: ns-ext.vix.com (204.152.184.64)
    *** Notre DNS a choisi le dernier de la liste précédente
User Datagram Protocol
    Source port: 1029 (1029)
    Destination port: domain (53)
    Length: 49
    Checksum: 0xead6
Domain Name System (query)
    Transaction ID: 0x3272
    Flags: 0x0100 (Standard query)
        0... .... .... .... = Query
        .000 0... .... .... = Standard query
        .... ..0. .... .... = Message is not truncated
        .... ...1 .... .... = Do query recursively
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
        www.ac-aix-marseille.fr: type A, class inet
            Name: www.ac-aix-marseille.fr
            Type: Host address
```

```
            Class: inet
    *** Et toujours la même question...
```

## Deuxième réponse

```
Frame 6 (168 on wire, 168 captured)
...
    Protocol: UDP (0x11)
    Header checksum: 0x5a17 (correct)
    Source: ns-ext.vix.com (204.152.184.64)
    Destination: ca-ol-marseille-12-195.abo.wanadoo.fr (213.56.59.195)
User Datagram Protocol
    Source port: domain (53)
    Destination port: 1029 (1029)
    Length: 134
    Checksum: 0x303c
Domain Name System (response)
    Transaction ID: 0x3272
    Flags: 0x8100 (Standard query response, No error)
        1... .... .... .... = Response
        .000 0... .... .... = Standard query
        .... .0.. .... .... = Server isn't an authority for domain
        .... ..0. .... .... = Message is not truncated
        .... ...1 .... .... = Do query recursively
        .... .... 0... .... = Server can't do recursive queries
        .... .... .... 0000 = No error
    Questions: 1
    Answer RRs: 0
    Authority RRs: 2
    Additional RRs: 2
    Queries
        www.ac-aix-marseille.fr: type A, class inet
            Name: www.ac-aix-marseille.fr
            Type: Host address
            Class: inet
    Authoritative nameservers
        ac-aix-marseille.fr: type NS, class inet, ns dnse.ac-aix-marseille.fr
            Name: ac-aix-marseille.fr
            Type: Authoritative name server
            Class: inet
            Time to live: 4 days
            Data length: 7
            Name server: dnse.ac-aix-marseille.fr
        ac-aix-marseille.fr: type NS, class inet, ns cianame.ac-clermont.fr
            Name: ac-aix-marseille.fr
            Type: Authoritative name server
            Class: inet
            Time to live: 4 days
            Data length: 22
            Name server: cianame.ac-clermont.fr
    ***Il n'y a pas de miracle...
    *** On reçoit la liste des DNS qui servent le domaine ac-aix-marseille.fr
    *** Comme on l'a vu dans notre recherche "à la main".
    Additional records
        dnse.ac-aix-marseille.fr: type A, class inet, addr 195.83.252.200
...
        cianame.ac-clermont.fr: type A, class inet, addr 194.254.204.31
...
```

## Troisième requête

```
Frame 7 (83 on wire, 83 captured)
...
    Protocol: UDP (0x11)
    Header checksum: 0xa8ab (correct)
    Source: ca-ol-marseille-12-195.abo.wanadoo.fr (213.56.59.195)
    Destination: dnse.ac-aix-marseille.fr (195.83.252.200)
User Datagram Protocol
```

```
    Source port: 1029 (1029)
    Destination port: domain (53)
    Length: 49
    Checksum: 0x0118
Domain Name System (query)
    Transaction ID: 0xe1ed
    Flags: 0x0000 (Standard query)
        0... .... .... .... = Query
        .000 0... .... .... = Standard query
        .... ..0. .... .... = Message is not truncated
        .... ...0 .... .... = Don't do query recursively
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
        www.ac-aix-marseille.fr: type A, class inet
            Name: www.ac-aix-marseille.fr
            Type: Host address
            Class: inet
    *** Et toujours la même question.
    *** C'est normalement la dernière pour cet hôte.
```

## Troisième réponse

```
Frame 8 (127 on wire, 127 captured)
...
    Protocol: UDP (0x11)
    Header checksum: 0x83c7 (correct)
    Source: dnse.ac-aix-marseille.fr (195.83.252.200)
    Destination: ca-ol-marseille-12-195.abo.wanadoo.fr (213.56.59.195)
User Datagram Protocol
    Source port: domain (53)
    Destination port: 1029 (1029)
    Length: 93
    Checksum: 0x440a
Domain Name System (response)
    Transaction ID: 0xe1ed
    Flags: 0x8480 (Standard query response, No error)
        1... .... .... .... = Response
        .000 0... .... .... = Standard query
        .... .1.. .... .... = Server is an authority for domain
        .... ..0. .... .... = Message is not truncated
        .... ...0 .... .... = Don't do query recursively
        .... .... 1... .... = Server can do recursive queries
        .... .... .... 0000 = No error
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 0
    Queries
        www.ac-aix-marseille.fr: type A, class inet
            Name: www.ac-aix-marseille.fr
            Type: Host address
            Class: inet
    Answers
        www.ac-aix-marseille.fr: type CNAME, class inet, cname copernic.crdp.ac-aix-marseille.fr
            Name: www.ac-aix-marseille.fr
            Type: Canonical name for an alias
            Class: inet
            Time to live: 115 days, 17 hours, 46 minutes, 39 seconds
            Data length: 16
            Primary name: copernic.crdp.ac-aix-marseille.fr
        copernic.crdp.ac-aix-marseille.fr: type A, class inet, addr 194.254.139.4
    *** Et voici la réponse finale...
    *** Avec l'indication qu'il s'agit d'un alias et avec le vrai nom.
            Name: copernic.crdp.ac-aix-marseille.fr
            Type: Host address
            Class: inet
            Time to live: 115 days, 17 hours, 46 minutes, 39 seconds
```

```
            Data length: 4
            Addr: 194.254.139.4
```

## Recherche du second hôte

Nous avons ici l'espoir de démontrer que notre DNS ne va pas partir d'un root-server, mais d'un des serveurs capable de nous documenter sur le TLD "fr." En effet, si le cache fonctionne correctement, ces informations doivent toujours être en la possession de notre DNS.

### Première requête

```
Frame 21 (72 on wire, 72 captured)
...
    Protocol: UDP (0x11)
    Header checksum: 0x6750 (correct)
    Source: ca-ol-marseille-12-195.abo.wanadoo.fr (213.56.59.195)
    Destination: dns.Princeton.EDU (128.112.129.15)
    *** C'est gagné!
    *** Il attaque sur dns.princeton.edu, serveur fourni par la recherche précédente.
User Datagram Protocol
    Source port: 1029 (1029)
    Destination port: domain (53)
    Length: 38
    Checksum: 0x89d0
Domain Name System (query)
    Transaction ID: 0x8e83
    Flags: 0x0000 (Standard query)
        0... .... .... .... = Query
        .000 0... .... .... = Standard query
        .... ..0. .... .... = Message is not truncated
        .... ...0 .... .... = Don't do query recursively
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
        www.voila.fr: type A, class inet
            Name: www.voila.fr
            Type: Host address
            Class: inet
    *** La question qui nous intéresse maintenant.
```

### Première réponse

```
Frame 22 (189 on wire, 189 captured)
...
    Protocol: UDP (0x11)
    Header checksum: 0x8b04 (correct)
    Source: dns.Princeton.EDU (128.112.129.15)
    Destination: ca-ol-marseille-12-195.abo.wanadoo.fr (213.56.59.195)
User Datagram Protocol
    Source port: domain (53)
    Destination port: 1029 (1029)
    Length: 155
    Checksum: 0x36a7
Domain Name System (response)
    Transaction ID: 0x8e83
    Flags: 0x8080 (Standard query response, No error)
        1... .... .... .... = Response
        .000 0... .... .... = Standard query
        .... .0.. .... .... = Server isn't an authority for domain
        .... ..0. .... .... = Message is not truncated
        .... ...0 .... .... = Don't do query recursively
        .... .... 1... .... = Server can do recursive queries
        .... .... .... 0000 = No error
```

```
    Questions: 1
    Answer RRs: 2
    Authority RRs: 2
    Additional RRs: 2
    Queries
        www.voila.fr: type A, class inet
            Name: www.voila.fr
            Type: Host address
            Class: inet
    Answers
        www.voila.fr: type A, class inet, addr 195.101.94.81
    *** Super!
    *** On a déjà la réponse finale.
    *** Peut-être parce que dns.princeton.edu est lui-même un serveur récursif.
    *** Normalement, on n'aurait dû recevoir que les serveur autorisés pour le domaine voila.fr
            Name: www.voila.fr
            Type: Host address
            Class: inet
            Time to live: 1 day, 10 hours, 54 minutes, 28 seconds
            Data length: 4
            Addr: 195.101.94.81
        www.voila.fr: type A, class inet, addr 195.101.94.80
            Name: www.voila.fr
    *** Tiens, il a même deux adresses (ça se fait).
            Type: Host address
            Class: inet
            Time to live: 1 day, 10 hours, 54 minutes, 28 seconds
            Data length: 4
            Addr: 195.101.94.80
    Authoritative nameservers
    *** On reçoit tout de même pour info.
    *** Les serveurs de noms pour voila.fr...
        voila.fr: type NS, class inet, ns ns.x-echo.com
            Name: voila.fr
            Type: Authoritative name server
            Class: inet
            Time to live: 4 days
            Data length: 15
            Name server: ns.x-echo.com
        voila.fr: type NS, class inet, ns ns1.x-echo.com
            Name: voila.fr
            Type: Authoritative name server
            Class: inet
            Time to live: 4 days
            Data length: 6
            Name server: ns1.x-echo.com
    Additional records
        ns.x-echo.com: type A, class inet, addr 195.101.94.1
            Name: ns.x-echo.com
            Type: Host address
            Class: inet
            Time to live: 12 hours, 7 minutes, 1 second
            Data length: 4
            Addr: 195.101.94.1
        ns1.x-echo.com: type A, class inet, addr 195.101.94.10
            Name: ns1.x-echo.com
            Type: Host address
            Class: inet
            Time to live: 12 hours, 10 minutes, 10 seconds
            Data length: 4
            Addr: 195.101.94.10
```

# Conclusions

Cet exemple vous aura j'espère aidé à comprendre comment travaille un serveur de noms récursif :

- Recherches itérative à partir :

- D'un root-server si le cache ne contient aucune information pertinente
- Des informations déjà reçues lors d'autres recherches, ce qui fait gagner du temps.

Pour ceux qui sont très observateurs, vous aurez constaté que toutes les réponses des serveurs de noms contiennent aussi des TTL (Time To Live). C'est la durée de validité de l'information. Cette information est importante, parce qu'elle permet de savoir si une information contenue dans le cache a des chances ou non d'être encore d'actualité. Vous aurez constaté aussi que, suivant les domaines ou les serveurs, ce TTL peut avoir des valeurs différentes.