

ASR ⇒ Administration Réseau : Adr4

TP2 : Système de Noms de Domaines (DNS)

Objectif :

Présentation du système de noms de domaines Internet (DNS) et mise en œuvre d'un serveur DNS sur une machine fonctionnant sous le système **Linux Debian**.

1. Bref historique

Dans les années 70, le réseau **ARPAnet** « ancêtre » de l'Internet d'aujourd'hui disposait de quelques centaines de calculateurs raccordés. Un fichier unique, **HOSTS.TXT** contenait alors toutes les informations à connaître sur ces calculateurs : les correspondances nom-adresse pour chaque ordinateur raccordé à **ARPAnet**. La fameuse table Unix, **/etc/hosts**, a été générée à partir de **HOSTS.TXT** (les champs non utiles à Unix ont été retirés).

Exercice 1 :

Avec les utilitaires UML (User Mode Linux):

- Créer un réseau, puis lancer 2 machines virtuelles nommées **serveurDNS** et **clientDNS**.
- Connecter l'interface **eth0** de chaque machine sur ce réseau.
- Affecter l'@IP **192.168.0.1** à l'interface **eth0** de la machine **serveurDNS** et l'@IP **192.168.0.5** à l'interface **eth0** de la machine **clientDNS**.
- Vérifier que les 2 machines communiquent.
- Sur chaque machine, éditer le fichier **/etc/hosts** et rajouter le nom et l'@ IP des 2 machines.
- Avec la commande **ping** vérifier que le fichier **/etc/hosts** sert d'annuaire local pour la résolution de nom.

2. Système de Noms de Domaines (DNS)

Il est évident qu'un fichier unique ne peut pas contenir de manière centralisée, les informations de tous les équipements raccordés à l'Internet. Un nouveau système a été imaginé avec les exigences d'une administration locale des données, d'une organisation hiérarchique et mondiale des ressources et d'une mémorisation des informations recueillies (notion de cache).

Le **DNS** (Domain Name Service) est une **base de données distribuée**. Cela permet un contrôle local de segments de l'ensemble de la base tout en rendant disponible les données de chaque segment.

La réplication des données et un système de cache peuvent améliorer la robustesse du système ainsi que les performances adéquates.

Le DNS est bâti selon le modèle client/serveur. Les serveurs de noms contiennent les informations sur des segments de la base de données et les rendent disponibles aux programmes clients appelés des « **resolveurs** ». Les **resolveurs** sont souvent de simples sous-programmes du noyau, appelés par des applications réseau nécessitant une résolution de nom, qui créent des requêtes et les envoient à travers le réseau à un serveur de noms (c'est le cas d'Unix et de Linux).

Indépendamment du « **resolverseur** » interne au noyau, il existe des « **resolverseurs** » au niveau utilisateur tels que **nslookup**, **host** ou **dig**.

Afin de pouvoir se servir du **DNS**, ces resolverseurs doivent disposer d'informations sur le serveur de noms à interroger.

Le fichier **/etc/resolv.conf**

Il contient des informations sur le domaine auquel appartient la machine et sur le(s) serveur DNS à interroger. Des explications détaillées de ces concepts sont données aux § 3 et 4 suivants.

Le fichier **/etc/nsswitch.conf**

La ligne **hosts** de ce fichier indique le choix pour les résolutions de noms entre une résolution locale par **/etc/hosts** (option **files**) ou distante par **DNS** (option **dns**). Ce fichier regroupe toutes les résolutions possibles allant des noms de login, noms de groupes, noms de machines en passant par les noms de protocoles.... Consulter le manuel de **nsswitch.conf** afin d'en savoir plus.

L'utilitaire **nslookup**

Il permet d'interroger un serveur **DNS** afin d'avoir des informations sur un domaine ou sur une machine. Par défaut **nslookup** interroge le serveur de noms configuré sur votre machine (Cf. fichier **/etc/resolv.conf**), mais vous pouvez interroger un autre serveur de noms.

Sur les systèmes Linux récents, l'utilitaire **nslookup** n'implémente pas toutes les fonctions ou est absent, pour laisser place à des outils plus performants comme **host** ou **dig**.

Le système Windows, par contre, implémente encore la version complète de **nslookup**.

Exemple en mode interactif

nslookup

Default Server: localhost si un serveur **DNS** est actif sur la machine locale

Address: 127.0.0.1

>help pour avoir de l'aide

>set type=ANY pour avoir toutes les informations

Remplacer ANY par le type d'enregistrement que vous souhaitez avoir.

Par exemple NS pour les serveurs DNS, MX pour les serveurs de mail,

SOA pour start of authority, A pour une adresse IP de machine, etc...

>info.iut-tlse3.fr le nom du domaine dont vous voulez avoir toutes les informations

>ls -t CNAME nom_du_domaine pour avoir tous les enregistrements de type **cname** (les alias) d'un domaine donné :

> server NAME 141.115.4.5. pour interroger un serveur DNS autre que votre serveur par défaut

L'utilitaire **host**

Il est presque équivalente à **nslookup**, mais d'un usage plus simple car il ne fonctionne pas en mode interactif.

Exemples :

host -a wanadoo.fr donne des informations sur le domaine wanadoo.fr

host -v -t MX wanadoo.fr donne des informations sur les MX (serveurs de mail) du domaine Wanadoo (vous pouvez indiquer un autre DNS).

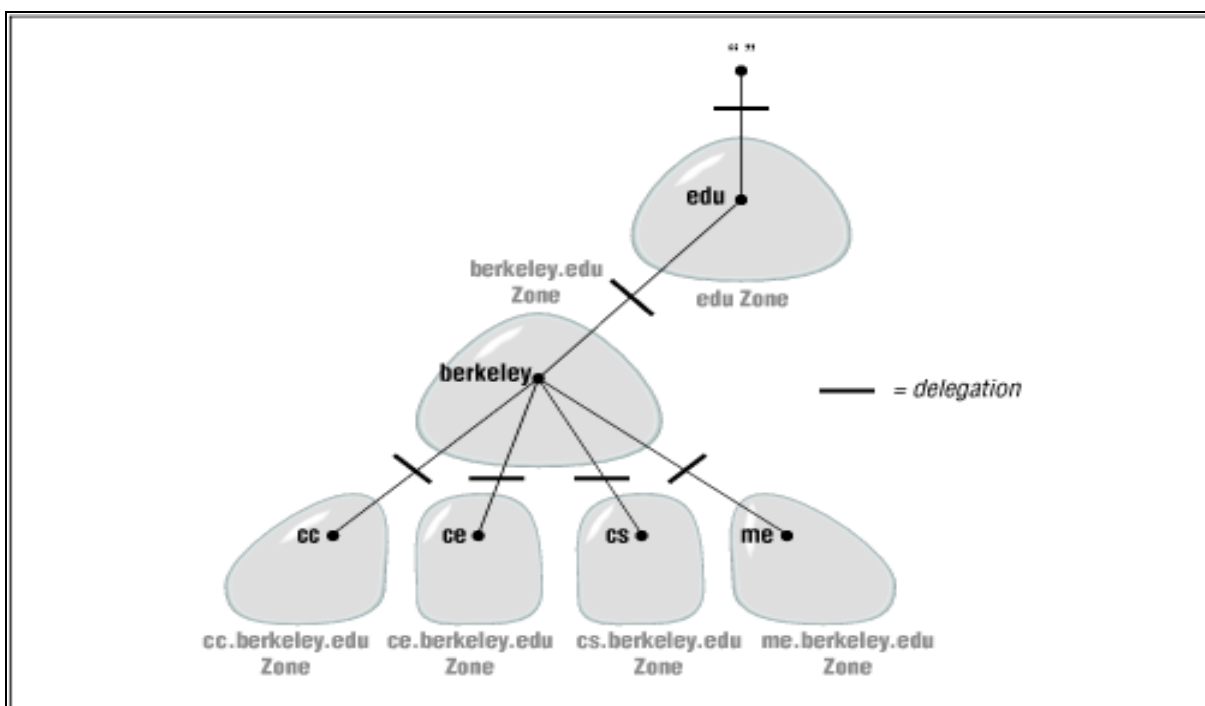
host -l -t any wanadoo.fr pour obtenir toutes les machines du domaine Wanadoo.

Ici le DNS par défaut sera utilisé, mais si ce serveur DNS n'est pas le DNS du domaine il indiquera qu'il n'est pas « autorisé ».

3. Structure des noms de domaines

La structure de la base de données **DNS** est similaire à celle d'un système de fichiers. La totalité de la BD peut être représentée par un arbre inversé. Chaque nœud de l'arbre est étiqueté, l'étiquette servant à identifier le nœud par rapport aux nœuds parents. La racine dispose d'un label vide et est non nommée.

Chaque nœud est lui-même racine d'un nouveau sous-arbre de l'arbre total. Chaque sous-arbre représente une partition de l'ensemble de la BD appelée **domaine**. Chaque domaine peut contenir des **sous-domaines**.



Un domaine a alors un nom unique. Le nom de domaine d'un domaine identifie sa position dans la BD, comme c'est le cas pour un répertoire d'un système de fichiers. Un nom de domaine est alors une suite de labels séparés par des « . », partant du nœud vers la racine de l'arbre.

Les domaines feuilles représentent des machines et peuvent pointer sur des adresses de réseau, des informations hardware ou de routage de messagerie.

Les domaines nœuds peuvent désigner des machines et pointer sur des informations du domaine.

Exemple: **archipel.siera.ups-tlse.fr** nom de machine+serveurs de courrier, Web...

L'interrogation du serveur de noms dépendra du contexte de recherche.

Un domaine peut être subdivisé en sous-domaines et la responsabilité de ces sous-domaines peut être déléguée à différentes organisations.

La délégation de l'autorité pour un sous-domaine crée une nouvelle zone, une partie de l'espace de noms administrée de manière autonome.

4. L'espace des noms de domaines Internet

4.1. Noms de domaines

Les domaines du premier niveau de l'espace de noms de domaines sont appelés des **TLD (Top Level Domain)**. Ils sont gérés par l'**ICANN** (Internet Corporation for Assigned Names and Numbers : www.icann.org)

Exemples de TLD :

- **com, edu, gov, mil, org, net, biz, name, ...**
- **fr, uk, es...**désignations géographiques : ISO 3166 (abréviations officielles de noms de pays sur 2 lettres:)

Le nom d'un nœud est composé d'au plus **63 caractères alphanumériques** dans l'ensemble {A-Z, a-z, 0-9, -} et doit commencer par une lettre (RFC 1032). Il est conseillé de ne pas dépasser 12 caractères.

4.2. Administration des noms de domaines

L'administration de l'espace de noms de domaines est hiérarchisée et décentralisée. Le **NIC** (Network Information Center) est l'**autorité responsable** de la coordination au niveau mondial. Le **NIC** a donné **délégation** à l'**autorité RIPE-NCC** en Europe qui a donné **délégation** à l'**AFNIC** en France (Association Française pour le Nommage Internet en Coopération : www.nic.fr).

Exemple :

L'**AFNIC** enregistre tous les noms de sous-domaines du domaine **fr**, avec un gérant pour chaque sous-domaine (délégation d'autorité). Le gérant du domaine **ups-tlse.fr** est responsable de :

- la **délégation** des noms de domaines en-dessous
- la désignation d'un administrateur pour le sous-domaine
- contacter le responsable TLD ou du domaine lors de l'enregistrement d'un nouveau sous-domaine.

4.3. Serveurs de noms et Zones de noms

Les programmes qui enregistrent les informations concernant l'espace de noms de domaines sont appelés **serveurs de noms**. Ces derniers disposent de toutes les informations sur une partie de l'espace de noms (**une zone**) qu'ils récupèrent à partir d'un fichier ou auprès d'un autre serveur de noms. Le serveur de noms est alors dit **serveur autorisé** ou **ayant autorité sur la zone**.

La différence entre domaine et zone est importante mais subtile. Une zone et un domaine peuvent partager un même nom de domaine mais contiennent des machines différentes. Une zone ne doit pas contenir d'informations sur une machine appartenant à un sous-domaine en délégation. Un serveur de noms devra donc charger des informations sur sa zone et non sur le domaine (Imaginez ce que le contraire donnerait si au niveau de la racine on chargeait le domaine et non la zone racine, tout l'espace de noms serait chargé !!!)

Un serveur de noms peut être de type **primaire ou secondaire**. Un serveur de noms **primaire** pour une zone lit les informations sur sa zone depuis un fichier local. Un serveur de noms **secondaire** pour une zone récupère les données sur la zone depuis un serveur de noms ayant autorité sur la zone, appelé le **serveur maître**. On a de plus en plus tendance à appeler un serveur secondaire un **serveur esclave**.

Les fichiers locaux à partir desquels un serveur primaire charge les données sur la zone sont appelés fichiers de données de zone. Ils contiennent des enregistrements appelés **RR** (Resource Records) qui décrivent une zone : description des machines appartenant à la zone et liens vers les sous-domaines en délégation.

Les RR sont des conteneurs d'informations regroupées par classe. La classe Internet est la plus connue et la plus utilisée. Les entrées de cette classe sont elles-mêmes de plusieurs types dont l'ordre d'apparition dans les fichiers de définitions doit toujours être le suivant :

- **SOA** indique l'autorité sur la zone (Start of Authority)
- **NS** indique le serveur de nom pour la zone (Name Server)
- **A** indique la correspondance nom-adresse (Address)
- **PTR** indique la correspondance adresse-nom (Pointer)
- **CNAME** définit un alias (Canonical Name)

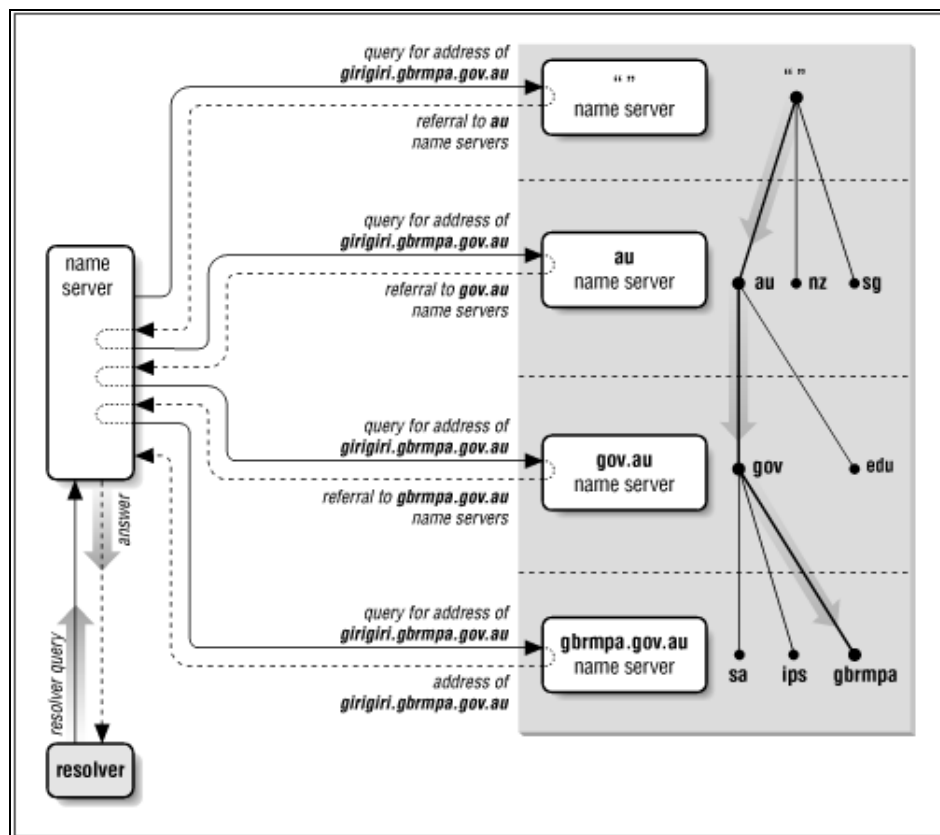
Les détails sur chacun de ces RR seront donnés dans les fichiers de configuration de zone (Cf. §5 sur **BIND**)

4.4. Résolution

Les serveurs de noms doivent retrouver les données de résolution à partir d'un espace de noms de domaines. S'il est normal qu'ils puissent retrouver des informations sur la zone pour laquelle ils sont autorisés, ils doivent également pouvoir retrouver toute information sur l'ensemble de l'espace de noms. Ce dernier étant structuré comme un arbre inversé, un serveur de noms ne nécessite alors de connaître que les serveurs de noms de la racine de l'arbre. Un serveur de noms lancera alors une requête vers un des serveurs « racines » qui pourra invoquer le serveur de noms sur le chemin du domaine.

Deux modes de résolution sont possibles : le mode **itératif** et le mode **récuratif**.

Dans le mode **itératif**, lorsqu'un serveur reçoit une requête de résolution, il retourne à l'initiateur l'adresse du serveur de noms du sous-domaine à qui il a donné la délégation. Le serveur initiateur envoie alors sa requête à ce serveur qui à son tour retourne l'adresse du serveur de noms du sous-domaine en délégation et ainsi de suite jusqu'à la résolution complète.



Dans le mode **récuratif**, lorsqu'un serveur reçoit une requête de résolution, il la relaie au serveur du domaine en délégation et ainsi de suite jusqu'au serveur sachant résoudre la requête. La réponse remonte alors en sens inverse.

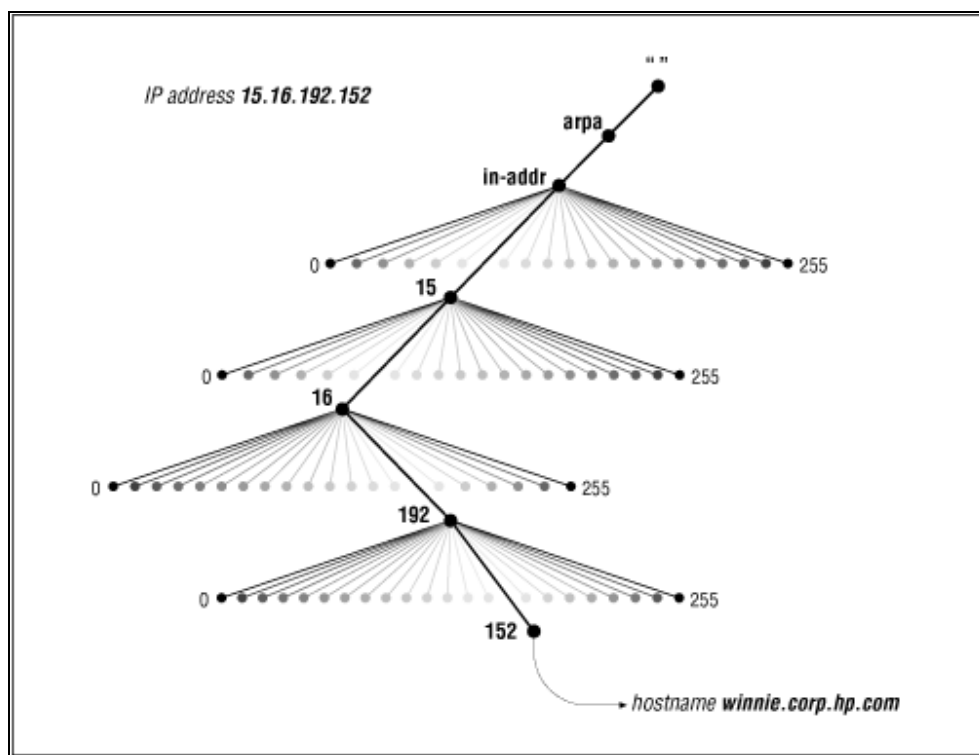
4.5. Résolution inverse

Une fonctionnalité manquante au processus de résolution jusque là est la résolution inverse qui consiste à faire correspondre un noms de domaine à une adresse IP : l'annuaire **inversé**.

Quand on utilise des tables comme `/etc/hosts` cette correspondance est triviale. Elle l'est beaucoup moins dans le cadre du **DNS** puisque l'espace de noms de domaines est un arbre indexé par des noms, et il faudrait une recherche exhaustive des données attachées à chaque domaine pour pouvoir satisfaire une telle requête de résolution inverse.

En fait une solution plus simple a été imaginée. Il a suffi de définir un nom de domaine **fictif** qui utilisera des adresses comme labels dans l'arbre, le domaine **in-addr.arpa**.

Les nœuds du domaine **in-addr.arpa** sont désignés en utilisant la notation décimale pointée des adresses IP. Ainsi le domaine **in-addr.arpa** peut avoir jusqu'à 256 sous-domaines correspondant aux 256 valeurs du 1^{er} octet de l'adresse IP. Chacun de ces sous-domaines peut avoir 256 sous-domaines à son tour...



Quand on lit un tel nom de domaine, l'adresse IP apparaît dans le sens **inverse** puisqu'on lit de la feuille vers la racine. Ainsi la machine **winnie.corp.hp.com** ayant l'adresse **15.16.192.152** sera enregistrée comme **152.192.16.15.in-addr.arpa**.

En faisant apparaître les premiers octets d'une adresse IP en haut de l'arbre, on donne la possibilité aux administrateurs de déléguer l'autorité à des zones **in-addr.arpa** en fonction de leur organisation. Ainsi la zone **15.in-addr.arpa** peut être déléguée à l'administrateur du réseau **15.0.0.0**. Cela n'aurait pas pu se faire si les octets apparaissaient dans l'autre sens.

5. Configuration du serveur BIND

Exercice 2 :

La version actuelle du DNS sous Linux, est la version 9 de **BIND** (Berkeley Internet Name Domain) et le programme qui l'implémente s'appelle **/usr/sbin/named**.

Sur la machine **serveurDNS**, exécuter le shell-script :

```
/etc/init.d/bind9 start
```

Dans BIND9, le programme **named** lance 4 threads par défaut sur un monoprocesseur et 1 thread de plus par processeur supplémentaire.

Vérifier avec la commande **ps -efl** que plusieurs threads **named** ont été lancés.

Quel est l'intérêt de lancer plusieurs threads **named** sur une machine?

5.1. Fichiers de configuration

Les fichiers de configuration de base d'un serveur **DNS** sont créés automatiquement dans le répertoire **/etc/bind** lors de l'installation du package BIND9 sur la machine.

- **db.0**
- **db.127**
- **db.255**
- **db.empty**
- **db.local**
- **db.root**
- **named.conf**
- **rndc.key**
- **zones.rfc1918**

Ces fichiers définissent un serveur DNS basique pour le domaine localhost et l'adresse IP de loopback.

Il faut en fonction de ce qu'on veut faire, créer les fichiers suivants :

- **/etc/bind /db.nomdedomaine**
fichier de zone qui fait correspondre les noms de machines et leur adresse IP pour la zone sur laquelle le serveur a autorité.
- **/etc/bind /db.adr_reseau**
fichier de zone inverse qui fait correspondre les adresses IP avec les noms de machines .

Remarques :

Dans la dernière version de **Linux Débian**, les options et les paramètres qui définissent le mode de fonctionnement du serveur et qui étaient placés en début du fichier **named.conf**, sont maintenant placés dans le fichier **named.conf.options**.

Les déclarations des zones sur lesquelles le serveur a autorité qui étaient placées en fin du fichier **named.conf**, sont maintenant placées dans le fichier **named.conf.local**.

Les fichiers **named.conf.options** et **named.conf.local** sont respectivement inclus en début et en fin du fichier **named.conf** chaque fois que le processus serveur **named** lit ses paramètres de configuration dans ce fichier (noter les directives **include** figurant en début et en fin du fichier **named.conf**).

Cette décomposition du fichier **named.conf** a été faite pour simplifier son paramétrage. Dans la plupart des cas, seul le fichier **named.conf.local** a besoin d'être paramétré.

5.2. Exemple de configuration

On cherche ici à configurer un domaine **mondomaine** avec comme adresse de réseau **192.168.0.0**.

named.conf

```
;Fichier d'amorçage du serveur
;primaire pour mondomaine
options {
    directory "/var/cache/bind";
    forward first;
    forwarders {
        10.2.0.5
    };
query-source address * port 53;
allow-query {
    127/8;
    ! 192.168.1.10;
    192.168.0/24;
};
allow-transfer { ! *; };
allow-update { ! *; };
listen-on port 53 { *; };
};
logging {
    category statistics { null;
    };
    category security {
    default_syslog;
    default_debug;
    };
    category default {
    null;
    };
};
zone "." {
    type hint;
    file "/etc/bind/db.root";
};
zone "0.0.127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.local";
};
zone "mondomaine" {
    notify no;
    type master;
    file "/etc/bind/db.mondomaine";
};
zone "0.168.192.in-addr.arpa" {
    notify no;
    type master;
    file "/etc/bind/db.192.168.0";
};
```

options définit les options du serveur dans son ensemble. On peut configurer plus finement en plaçant les options dans les zones (si vous gérez plusieurs domaines ou des sous domaines par exemple).

directory indique le répertoire où se trouvent les fichiers. On peut aussi indiquer le chemin complet.

forward peut avoir plusieurs options (*first, only*) **first** redirige les requêtes vers les serveurs se trouvant dans la liste forwarders, si les hôtes ne répondent pas, le serveur tentera de répondre. **only** redirige directement les requêtes vers les serveurs se trouvant dans la liste forwarders. Dans ce cas, on définit un serveur « esclave ».

forwarders indique les serveurs vers lesquels les requêtes sont envoyées. **10.2.0.5** est le serveur DNS.

query-source indique que le port 53 est le port d'échange (source et destination) entre les serveurs DNS (utile en cas de firewall).

allow-query contient une liste des adresses dont le serveur acceptera ou refusera les requêtes. L'ordre compte, le premier l'emporte. 127/8 autorise localhost, on interdit la machine 192.168.1.10 et on autorise les autres (un exemple seulement).

allow-transfer interdit les transferts de requête de zone. Par défaut cela est autorisé de partout. **allow-update** refuse les instructions de mises à jour de la base de données de zone. Par défaut les mises à jour sont refusées.

listen-on port 53 indique le port en écoute pour les clients et les interfaces. Indiquer * pour écouter sur toutes les interfaces, ou l'adresse IP de la carte.

logging permet de configurer les logs de **named**. On peut les envoyer vers deux canaux syslog ou un fichier ou null.

Il existe plusieurs types de "**category**" (statistics, security, default,...). Vous pouvez paramétrer très finement cela. **category statistics** génère un rapport périodique d'activité.

category security requêtes acceptées/refusées.
category default default est équivalent à toutes catégories.

zone définit les options s'appliquant à des zones particulières. La zone **0.0.127.in-addr.arpa** crée une zone pour le réseau loopback.

La zone **.** indique l'emplacement du root du serveur du domaine internet. Un forward only ne nécessite pas de zone. La zone **mondomaine** est la zone à créer qui a comme fichier **db.mondomaine**.

type déclare le type d'entrée, il en existe de plusieurs types (master, hint)

master déclare ce serveur comme serveur primaire. Si vous créez un serveur secondaire indiquez slave.

hint déclare que cette entrée n'est qu'un endroit ou débiter les recherches, la racine du DNS le plus souvent.

notify no pour ne pas informer les autres serveurs s'il y a des changements dans la zone.

/etc/bind/db.root

Vous n'avez pas à modifier ce fichier. Il contient les adresses des serveurs root.

/etc/bind/db.local

```
@      IN      SOA  monserveur.mondomaine. root.monserv  
                2007012600 ; numéro de série  
                28800 ; rafraîchissement toutes les 8H  
                14400 ; nouvel essai toutes les 4H  
                604800 ; expiration dans 7 jours  
                86400 ) ; temps de vie minimal dans le cache 24H  
NS      monserveur.mondomaine.  
1       PTR   localhost.
```

Remarques :

@ permet de spécifier le nom de domaine quand c'est le même que celui d'origine.

Normalement vous n'avez pas à changer les valeurs qui sont dans ce fichier.

La première partie est identique dans les trois fichiers, si vous devez faire une modification sur un fichier vous devez modifier le numéro de série afin de faire connaître cette modification aux autres serveurs dns. Changez la date (**20070126** correspond au **26 Janvier 2007**)lorsque vous faites une modification. Si vous devez faire plusieurs modifications dans la même journée incrémenter le **00**.

/etc/bind/db.mondomaine

Vous indiquez dans ce fichier avec des enregistrements de type **A** , les machines que vous souhaitez pouvoir appeler par leur nom (équivalent au fichier /etc/hosts).

Indiquez aussi avec un enregistrement de type **MX**, le serveur **SMTP** de votre domaine (s'il existe).

Les enregistrement de type **CNAME** (alias) permettent de définir des alias de noms de machines.

L'avantage étant de pouvoir changer de machine sans être obligé de faire de grosses modifications.
N'hésitez donc pas à utiliser les alias.
Ce fichier est celui que vous allez modifier le plus souvent, pensez donc à changer le numéro de série.

```
@      IN      SOA  monserveur.mondomaine.  root.monserveur.mondomaine. (
2007012600 ; numéro de série
28800 ; rafraîchissement toutes les 8 heures
14400 ; nouvel essai toutes les 4 heures
604800 ; expiration dans 7 jours
86400 ) ; temps de vie minimal dans le cache 24 heures
```

; serveur de nom

```
      IN      NS    monserveur.mondomaine.
```

;adresses IP des machines

```
localhost      IN      A      127.0.0.1
monserveur     IN      A      192.168.0.1
monserveurweb  IN      A      192.168.0.2
```

;Alias

```
www      IN      CNAME    monserveurweb
ftp      IN      CNAME    monserveurftp
```

; Serveur smtp

```
monserveursmtp  IN      A      192.168.0.3
                IN      MX      10 monserveursmtp.mondomaine.
```

/etc/bind/db.192.168.0

Fichier des correspondances inverses. Chaque entrée de type **A** dans le fichier des correspondances directes doit avoir une correspondance dans ce fichier.

Les 1, 2, 3 correspondent à respectivement aux adresse 192.168.0.1, 192.168.0.2, etc

```
@      IN      SOA  monserveur.mondomaine.  root.monserveur.mondomaine.(
2007012600 ; numéro de série
28800 ; rafraîchissement toutes les 8 heures
14400 ; nouvel essai toutes les 4 heures
604800 ; expiration dans 7 jours
86400 ) ; temps de vie minimal dans le cache 24 heures
```

; serveur de nom

```
      IN      NS    monserveur.mondomaine.
```

; adresses IP inverses

```
1      IN      PTR    monserveur.mondomaine.
2      IN      PTR    monserveurweb.mondomaine.
3      IN      PTR    monserveursmtp.mondomaine.
```

5.3. Remarques

- Pensez à toujours mettre un **point** à la fin des noms de machine + domaine.
- Vous n'aurez certainement pas le besoin d'un serveur SMTP, il est là pour l'exemple. Si vous deviez en ajouter un deuxième indiquer un poids supérieur (IN MX 15 monautreserveursmtp). Si vous souhaitez en faire votre SMTP principal indiquez un poids inférieur.

- Les numéros de série peuvent être différents d'un fichier à un autre. Vous n'avez qu'à modifier celui du fichier que vous modifiez.
- **Postmaster** doit être un compte existant sur votre machine. Ce qui est normalement le cas. Il recevra tout le courrier concernant ce domaine.
- Toutes les options disponibles n'ont pas été données, mais toutefois, il peut être utile d'enregistrer certains **logs** de votre **DNS**. Voir les paramétrages du fichier **named.conf**. Attention à ne pas enregistrer toutes les requêtes et entre autres les requêtes qui aboutissent, car vous auriez alors des fichiers **log** énormes.
- Sur la machine serveur DNS et sur les machines clients, pensez à configurer le fichier **/etc/resolv.conf** qui doit contenir les lignes suivantes :

domain mondomaine
nameserver 192.168.0.1 (autant utiliser le DNS que vous venez de configurer, vous pouvez éventuellement en indiquer un autre).
- Le shell script **/etc/init.d/bind9** permet de démarrer le serveur de nom (processus **named**) avec le paramètre **start**, de l'arrêter avec le paramètre **stop** ou de lui faire lire son fichier de configuration avec le paramètre **reload**.
- Par défaut, **named** enregistre ses messages d'erreurs en fin du fichier **/var/log/syslog**. En cas d'erreurs, lister la fin de ce fichier avec la commande **tail**.

Exercice 3 :

- En supposant que les 2 machines **serveurDNS** et **clientDNS** appartiennent au domaine **tpDNS** d'adresse réseau **192.168.0.0**, configurer sur la machine **serveurDNS**, un serveur DNS **maître** pour la zone incluant ces 2 machines.
- Configurer un client DNS sur les 2 machines.
- Avec la commande **host**, tester le bon fonctionnement du serveur DNS.
- Dans le fichier **/etc/nsswitch.conf**, sur la ligne **host**, permuter successivement les options **files** et **dns**. Avec la commande **ping**, vérifier dans chaque cas le mode de résolution de noms.