Côté Cours : le système DNS

Description du thème

Propriétés	Description
Intitulé long	Mise en œuvre pratique d'un système DNS complet
Formation concernée	BTS Services informatiques aux organisations
Matière	SI5 - Support des services et des serveurs
Présentation	L'objectif est de reproduire le fonctionnement complet du système DNS dans la salle de TP. Le professeur gère un serveur racine et chaque groupe d'étudiants gère son propre nom de domaine sans connaître les détails des domaines des autres groupes . La résolution de noms se fait par rapport au serveur racine local.
Savoirs	Caracteristiques des applicatifs standards
Compétences	Installer, configurer et administrer un service
Transversalité	
Pré-requis	Installer, configurer et administrer un serveur Linux ou Windows 2003 Routage IP pour mettre en place les plate-formes
Outils	Serveur Linux debian Lenny (stable), bind9 (version 9.6) ou serveur Windows 2003 Clients linux, Windows ou autre système.
Mots-clés	DNS, nom de domaine, serveur primaire, serveur secondaire, serveur racine, zone primaire, zone secondaire, zone reverse, délégation, requête récursive, requête itérative, ACL
Durée	8 heures pour le TP complet, Certaines étapes comme la sécurisation, la délégation et les serveurs secondaires peuvent n'être traitées que dans la cadre d'un approfondissement. 4 heures si ces dernières étapes ne sont pas mises en œuvre.
Auteur(es)	Frédéric Varni, Apollonie Raffalli, avec l'aide précieuse de Roger Sanchez
Version	v 1.2
Date de publication	13/04/09
Dernière modification	02/07/10

Présentation

Contexte logistique et matériel

Il s'agit de simuler, dans la salle laboratoire réseau, l'organisation du système DNS tel qu'il existe sur Internet.

Chaque groupe dispose d'un minimum de 2 postes. Trois ou quatre serait l'idéal (cela peut être des machines virtuelles) :

- un poste pour le serveur maître de la zone principale ;
- un poste pour le serveur maître de la délégation ;
- un (des) hôte(s) DNS correctement configuré(s).

Outre les postes des étudiants, il faut au moins :

- un poste pour que l'enseignant puisse gérer la racine du DNS ;
- un poste "n'appartenant" à aucun groupe et configuré sur le système DNS du TP permettra aux groupes de tester leur propre configuration DNS de "l'extérieur".

Cette configuration peut servir de base à plusieurs autres TP sur la messagerie électronique ou la gestion de certificats SSL sur des sites web par exemple.

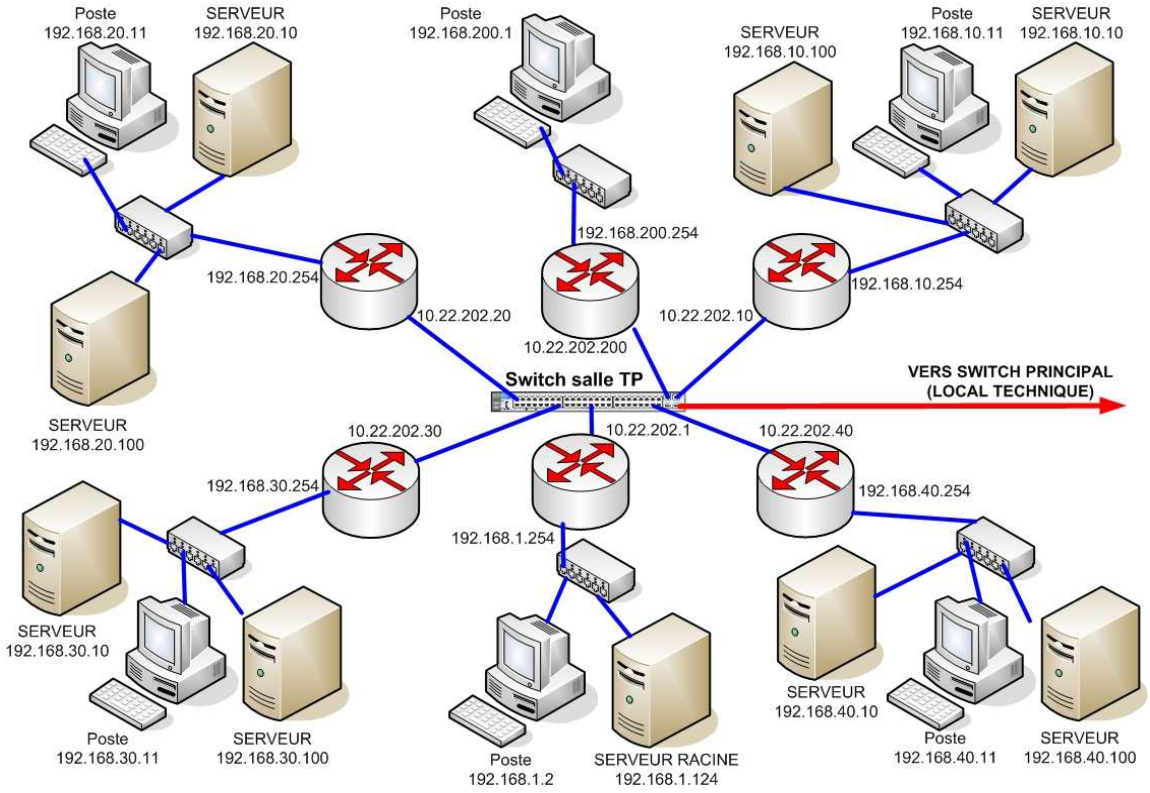
On supposera que :

- Le réseau du professeur est en 192.168.1.0/24.

L'adresse du serveur racine est 192.168.1.124.

- Les réseaux des plate-formes des élèves vont de 192.168.10.0/24 à 192.168.120.0/24 (de 10 en 10 sur le troisième octet)
- Le réseau "indépendant" permettant aux étudiants de tester leur configuration de l'extérieur est en 192.168.200.0/24

Schéma réseau "possible" réduit à 4 plate-formes



On considère dans la suite que le système DNS local à votre laboratoire comporte un **seul serveur racine géré par le professeur.** Chaque groupe va gérer son propre nom de domaine sans connaître les détails des domaines des autres groupes (à part leur nom et le nom des machines principales). La résolution de noms se fera par rapport au serveur racine local. Pour la résolution de noms chaque groupe utilise son propre serveur DNS pour répondre aux questions récursives des clients.

Des exemples de fichiers de configuration sont donnés pour le **serveur DNS BIND9** (*Berkeley Internet Name Domain)* sous Debian, ils doivent être transposables pour d'autres serveurs sous Windows et sous Unix. On trouve en **annexe 6** des pistes pour Windows 2003 Server ainsi que de nombreux liens vers les pages technet de Microsoft.

Des compléments de cours sont disponibles ici :

- Un diaporama sur le système DNS de l'AFNIC : http://www.afnic.fr/noncvs/formations/dns_court/dns.pdf
- Une auto-formation complète proposée par l'AFNIC : http://www.afnic.fr/ext/dns/index.html.
- Les **RFC** sont disponibles à l'adresse suivante : http://www.dns.net/dnsrd/rfc/.

Sous Linux, il convient d'installer le serveur DNS bind9 sur les machines qui vont faire office de serveurs DNS. Sous debian :

```
apt-get update
apt-get install bind9 bind9-doc
```

Un nouveau groupe ainsi qu'un utilisateur système "bind" sont créés. Le démon *named* est démarré automatiquement.

Les fichiers principaux nécessaires à la configuration du DNS sont créés par défaut dans **`/etc/bind/`** :

- **le fichier de configuration globale du serveur** `named.conf` qui inclut en fait 2 autres fichiers `named.conf.local` et `named.conf.options`. Leur rôle est notamment de :
  - ➔ donner les chemins vers les autres fichiers comme le fichier des serveurs racines et les fichiers de zone ;
  - ➔ déclarer l'autorité sur les zones (localhost par défaut)
  - ➔ définir diverses options (mode récursif ou itératif, etc.)

  Le contenu de ces fichiers est expliqué en annexe 1.

- **le fichier des serveurs racine** `db.root` qui contient la liste de tous les serveurs root avec leur adresse IP respective.

- **un fichier par zone** pour toutes les zones pour lesquelles le serveur a autorité, il contient les enregistrements propres à chaque zone ; le fichier créé par défaut est `db.local` et correspond à la zone "localhost".

- **les fichiers de zone reverse** ; ils sont au nombre de 3 par défaut : db.127 (zone reverse pour localhost), db.255 (zone locale de broadcast), db.0 (zone locale de broadcast).

Des exemples de fichiers de zone sont détaillés en annexe 2. Si ces fichiers sont modifiés, il est nécessaire de les "relire" ou de redémarrer le démon *named* (`/etc/init.d/bind9 reload ou /etc/init.d/bind9 restart`)

-

Apport théorique pour comprendre le rôle des fichiers de configuration

Le **système DNS** (*Domain Name System*) a en charge d'établir la correspondance entre un nom pleinement qualifié (FQDN) et une adresse IP. Le système DNS permet à des hôtes du réseau de soumettre des requêtes à un serveur DNS afin d'obtenir l'adresse IP d'un hôte connaissant le nom de cet hôte (par exemple www.google.com → 209.85.229.99). Cette traduction des noms en adresses IP doit toujours être réalisée puisque que seule l'adresse IP permet de communiquer sur le réseau.

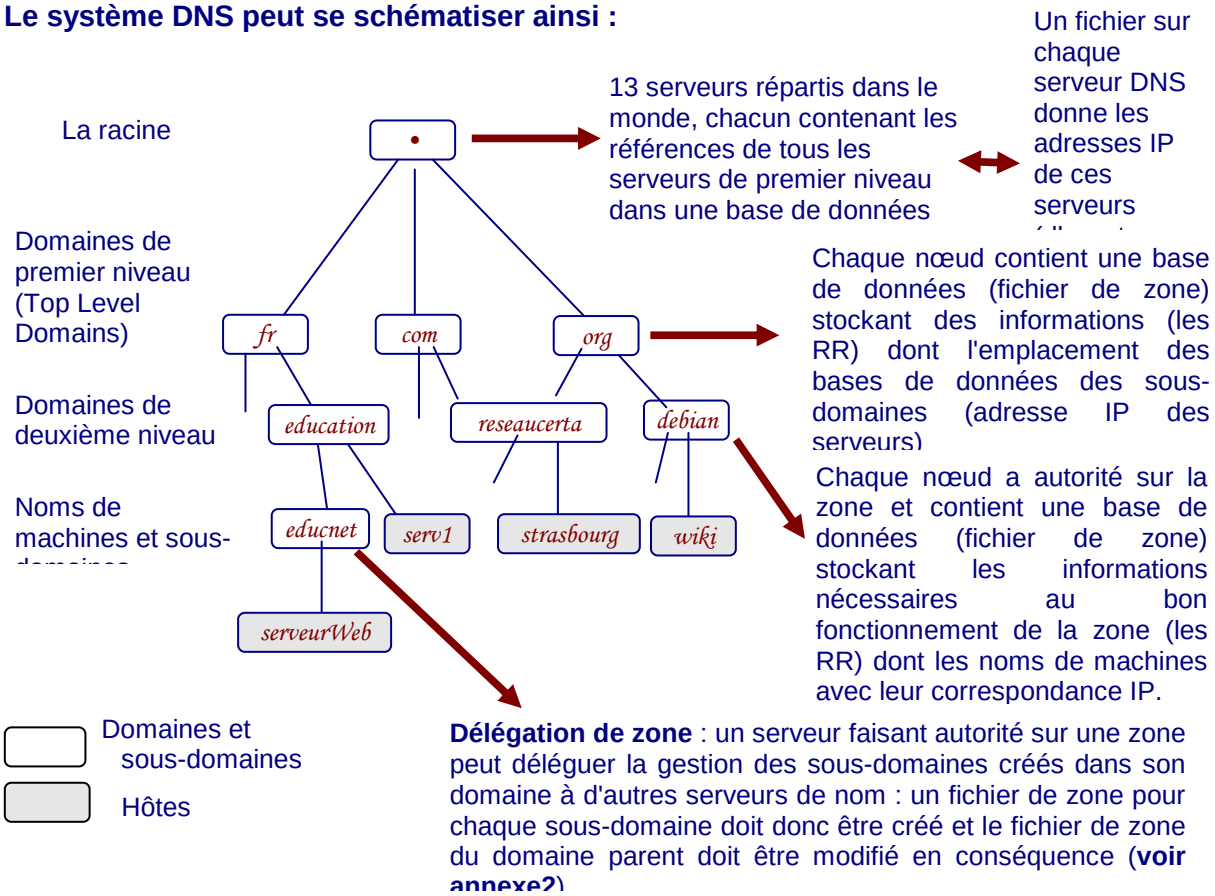
Il s'agit d'un modèle en **arborescence hiérarchique** avec une **gestion décentralisée des données** (chacun étant responsable des données de sa zone).

Le système de noms DNS se présente sous forme d'un arbre inversé avec pour sommet "la racine" et un ensemble de nœuds représentant des domaines identifiés par un label (fr, education.fr, org, com, etc.).

Un serveur de noms particulier s'occupe d'un nœud de l'arborescence ou d'un ensemble de nœuds sur lequel il aura **autorité**. On dit que le serveur gère une **zone d'autorité.** C'est à dire qu'il gérera l'attribution des noms et résoudra les noms via **une base de données** (matérialisée par ce qu'on appelle un **fichier de zone**) distincte pour chaque nœud. Chaque information élémentaire de la base de données DNS est un objet appelé "*resource record*" (RR).

Un nœud peut contenir aussi bien des domaines que des noms de machines.

Le système DNS peut se schématiser ainsi :



Délégation de zone : un serveur faisant autorité sur une zone peut déléguer la gestion des sous-domaines créés dans son domaine à d'autres serveurs de nom : un fichier de zone pour chaque sous-domaine doit donc être créé et le fichier de zone du domaine parent doit être modifié en conséquence (**voir annexe2**)

Dans notre exemple, quel serveur va résoudre le nom d'hôte pleinement qualifié serveurWeb.educnet.education.fr ?

Le serveur racine ne sait pas où se trouve cet hôte, par contre il possède un enregistrement pour le **domaine "fr".**

La **zone "fr"** est aussi restreinte au nœud correspondant. Son fichier de zone inclut donc des informations sur les délégations de gestion du reste du domaine dont le domaine "education".

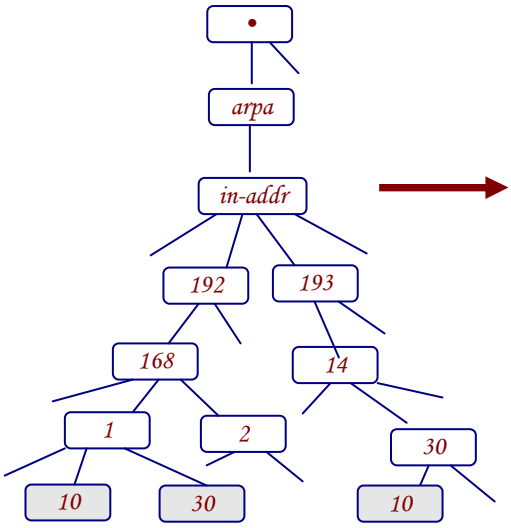
Le fichier de zone "education.fr" peut éventuellement posséder un enregistrement pour cet hôte car il est possible qu'une zone d'autorité comprenne un domaine et un sous-domaine. Mais nous supposerons ici que la gestion des noms a été déléguée. Le fichier de zone correspondant contient donc les informations nécessaires pour résoudre des noms dans cette zone (tel que serv1.education.fr) et les informations sur la délégation de la zone "educnet.education.fr".

Le fichier de zone "educnet.education.fr" possède l'information concernant l'hôte "serveurWeb" et pourra ainsi résoudre le nom **serveurWeb.educnet.education.fr.**

Il existe deux modes de résolution de noms : le mode récursif et le mode itératif.
Dans le mode récursif, le client envoie une requête au serveur DNS qui renvoie la réponse complète au client après avoir lui-même éventuellement interrogé d'autres serveurs de noms (s'il n'a pas la réponse en cache et s'il n'est pas autoritaire pour la zone).
Dans le mode itératif, le client envoie une requête au serveur DNS qui renvoie la réponse complète s'il est autoritaire pour la zone concernée mais une réponse partielle dans le cas contraire qui redirige le demandeur vers un autre serveur DNS afin qu'il poursuivre lui-même la résolution et ainsi de suite jusqu'à l'obtention de la réponse complète.

Pour de raisons de performance et de sécurité, il est conseillé de configurer les serveurs de noms pour qu'ils n'acceptent les requêtes en mode récursif que pour les machines de la zone pour laquelle ils sont autoritaires.

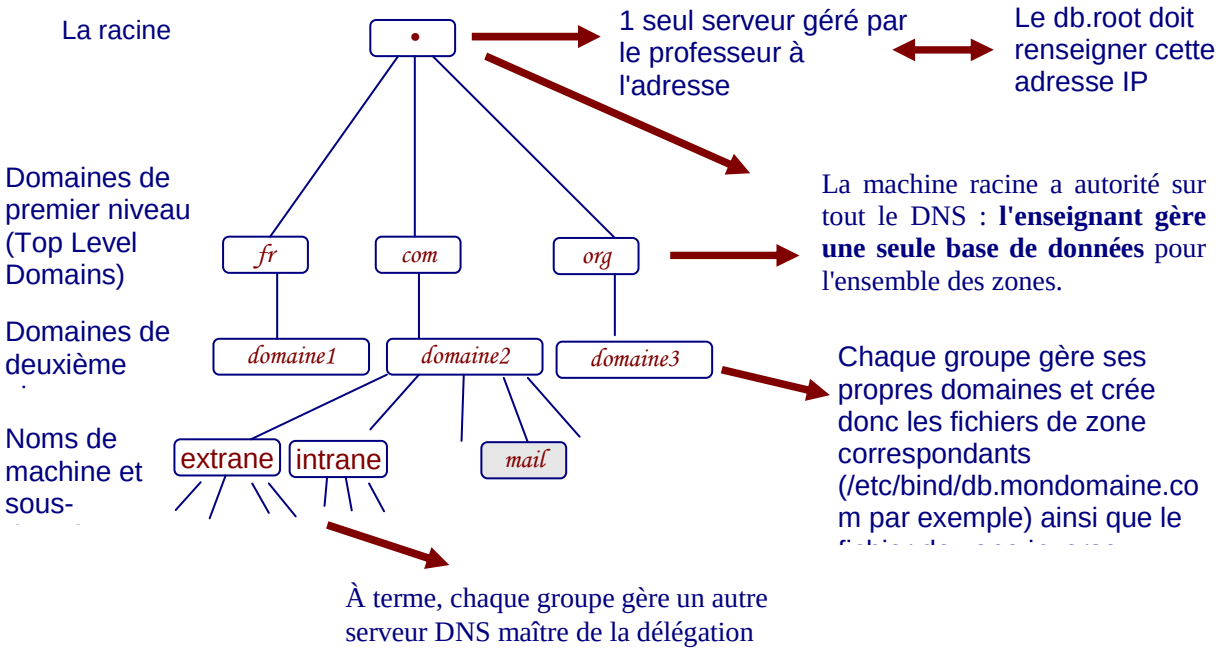
La zone in-addr.arpa (zone reverse)
Le domaine in-addr.arpa est un domaine spécial chargé de réaliser les recherches inversées, c'est-à-dire retrouver un nom en connaissant l'adresse IP.



Zone inverse pour chaque réseau dans le domaine in-addr.arpa.
Par exemple, la zone de recherche inverse pour le reseau 192.168.1.0 dans le domaine sera 1.168.192.in-addr.arpa. Cette zone devra répondre pour toutes les adresses déclarées dans la tranche 192.168.1.0 à 192.168.1.254. On inscrira dans cette zone tous les noeuds du réseau pour lesquels on désire que la résolution inverse fonctionne.

Des exemples de fichiers de zone sont donnés en annexe 2.

Le système DNS simplifié du TP



Attention : il faut garder une copie de l'original du fichier /etc/bind/db.root

Déroulement de la séquence

1. Réservation et déclaration du nom de domaine

Par groupe de 2, vous choisirez au moins un nom de domaine que vous enregistrerez auprès du professeur qui gère le serveur racine ; vous pouvez choisir chaque nom de domaine dans n'importe quelle zone (fr, com, org, edu, info, de, us, ...). Pour enregistrer chaque nom de domaine vous devez fournir les informations suivantes :

- nom du domaine pleinement qualifié
- adresse IP et nom du serveur principal du domaine
- éventuellement adresse(s) IP et nom(s) du ou des serveurs secondaires

Les noms des domaines enregistrés sont affichés au tableau au fur et à mesure des déclarations.

Votre professeur renseignera en parallèle les paramètres sur le serveur DNS racine.

2. Configuration de chaque domaine (aide en annexes 1 et 2)

Pour chaque domaine que vous enregistrez, vous devez créer au moins les machines[1] « www », « ftp » et « mail » ; la machine « mail » est désignée par l'enregistrement MX principal du domaine).

Vous n'oublierez pas de **garder une copie de l'original du fichier /etc/bind/db.root avant de le modifier**

Vous procédez à la **première batterie de tests** (voir **annexe 5**) qui consiste tout simplement à vérifier si votre serveur est correctement configuré syntaxiquement.

3. Configuration des postes clients

Vous configurez tous les postes de votre plate-forme sur votre DNS principal.
Sous Linux, cette configuration se réalise dans le fichier /etc/resolv.conf.

Vous procédez alors à la **deuxième batterie de tests** (voir **annexe 5**).

4. Configuration de la zone inverse (aide en annexes 1 et 2)

Pour le réseau de votre groupe, vous avez obtenu une délégation de la racine sur la zone inverse que vous devez configurer.

Votre professeur a présumé que les mêmes serveurs sont utilisés pour gérer votre nom de domaine et votre zone inverse.

Vous réitérez la **première et la deuxième batterie de tests** sur la zone inverse (voir **annexe 5**).

1 Le terme machine désigne un hôte DNS ; vous devez donc utiliser ici le véritable nom de la machine ou un alias.

5. Sécurisation minimale du serveur (aide en annexe 3).

Le fonctionnement par défaut de bind9 est récursif et ceci peut entraîner des problèmes de sécurité. Il est normal de prendre en charge de manière récursive les interrogations émises par les hôtes de votre réseau de manière à alimenter le cache et optimiser le fonctionnement du service.
Mais vous devez interdire la résolution récursive pour toute machine qui n'appartient pas à votre groupe. Bien entendu les requêtes itératives sur votre nom de domaine sont autorisées pour tout le monde. Vous devez pour cela utiliser les notions d'ACL et éventuellement de vues.

Vous trouverez en **annexe 3** une synthèse sur les listes de contrôle d'accès (ACL) et les vues (VIEW).

Vous procédez à la **troisième batterie de tests** (voir **annexe 5)**.

6. Création des sous domaines (délégation de zone – aide en annexe 2)

Une fois que vos domaines principaux sont servis et sécurisés correctement, vous restructurez votre espace de nom en créant, pour chaque domaine deux sous-domaines : intranet et extranet (intranet.mondomaine.org et extranet.mondomaine.org).

Sur le domaine intranet vous créerez les enregistrements pour les machines suivantes :

- www
- ftp
- support

Sur le domaine extranet vous créerez les enregistrements pour les machines suivantes :

- www
- ftp
- clients
- fournisseurs

7. Configuration des serveurs secondaires (aide en annexe 4)

Vous pouvez maintenant configurer un serveur DNS secondaire. Si vous ne disposez pas d'un autre poste, vous pouvez utiliser le serveur maître de la délégation pour votre serveur secondaire de la zone principale et votre serveur de la zone principale pour votre serveur secondaire du serveur de la délégation.

Les explications sont données en **annexe 4.**

N'oubliez pas de le déclarer à votre professeur afin qu'il l'enregistre sur le serveur racine.

Vous procédez à la **quatrième batterie de tests** (voir **annexe 5)**.

Annexes

Annexe 1 : les fichiers de configuration principale

Pour des exemples plus complets : http://www.afnic.fr/ext/dns/html/seq4893.html

Fichier named.conf

```
include "/etc/bind/named.conf.options";

// prime the server with knowledge of the root servers
zone "." {
        type hint;
        file "/etc/bind/db.root";
};

// be authoritative for the localhost forward and reverse zones, and
for broadcast zones as per RFC 1912

zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};

include "/etc/bind/named.conf.local";
```

Il est conseillé d'utiliser ce fichier pour configurer diverses options

"hint" est le type pour le fichier des serveurs root, qui ont autorité pour la zone point (".").

Chemin vers le fichier dans lequel sont listés les différents serveurs root avec leurs adresses IP.

Déclaration de la zone "localhost" pour laquelle le serveur a autorité :
"**type master**" car serveur maître sur la zone)
"**file**" donne le chemin vers le fichier de zone correspondant (donné ici en chemin absolu)

Toute déclaration de zone suit ce modèle.

Déclaration des zones reverses imposées par la RFC1912

Il est conseillé d'utiliser ce fichier pour ajouter des

Il est déconseillé de modifier ce fichier car il pourrait être modifié lors d'une mise à jour.

Fichier named.conf.local : déclarations de zone primaire et secondaire.

```
zone "exemple.fr" {
      type master;
      file "db.exemple.fr";
};
```

Chemin donné en relatif car ce fichier sera créé dans le répertoire de stockage défini par la directive "directory" du fichier

Les types de zone peuvent aussi être "*forward*" ou "*slave*", d'autres directives sont possibles.

Exemple pour une zone secondaire :

```
zone "exemple.fr" {
      type slave;
      file "slave/db.exemple.fr";
      masters {192.1.1.1;};
};
```

Voir explications en **annexe 4**

Exemple pour une zone inverse maître :

```
zone "1.192.168.in-addr.arpa" {
      type master;
      file "db.192.168.1.rev";
};
```

Fichier named.conf.options

```
options {
       directory "/var/cache/bind";
        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };
};
```

Répertoire de travail
C'est dans ce répertoire que l'on créera les fichiers de zone

Une multitude d'options existe ; en voici quelques unes :

- recursion yes | no; : Le DNS est autorisé à faire ou pas de la résolution récursive. Cette option peut être limitée en portée par l'option *allow-recursion* (voir ci-après) ou par l'utilisation de cette option dans une vue (voir annexe 3)
- allow-recursion {
  <adresseIP> | <classe-d'adresse> | <nom_acl> | <mot_cle>;
  ...;
  }; : hôtes ayant l'autorisation d'effectuer des demandes récursives sur le serveur de noms (**à partir de la version 9.4 bind ne répond, par défaut, qu'à son propre sous-réseau**)
- allow-query {
  <adresseIP> | <classe-d'adresse> | <nom_acl> | <mot_cle>;
  ...;
  }; : hôtes ayant l'autorisation d'interroger le serveur de noms (par défaut, tous les hôtes).
- Forwarders {adresse_ip1; ...,;} : adresses IP correspondant aux serveurs de noms vers lesquels les requêtes seront envoyées pour la résolution des requêtes que notre serveur ne sait pas résoudre.
- forward (first | only); :
  first : les serveurs de noms spécifiés dans la directive *forwarders* sont interrogés en premier puis en cas d'echec *named* tentera de résoudre le nom lui-même.
  only : seul le serveur de noms spécifié dans la directive forwarders sera interrogé ; en cas d'échec named ne tentera pas d'effectuer cette résolution.
- Allow-update {
  <adresseIP> | <nom_acl> | <mot_cle>;
  ...;
  }; : hôtes autorisés à mettre à jour les informations de la zone.
- Allow-transfer {
  <adresseIP> | <nom_acl> | <mot_cle>;
  ...;
  }; : hôtes autorisés à transférer les informations de la zone. Tous les hôtes par défaut.
- notify yes | no : détermine si *named* envoie une notification aux serveurs esclaves quand une zone est mise à jour (c'est *"yes"* par défaut).

Les mots clés peuvent en général être :

- *any* : toutes les adresses IP
- *localhost*
- *localnets* : les réseaux directement connectés au serveur
- *none* : aucune adresse IP

Certaines options peuvent être globales et/ou comprises dans une zone.

Annexe 2 : les fichiers de zone

Pour des exemples plus complets et plus de précisions :
http://www.afnic.fr/ext/dns/html/seq4892.html

Le fichier db.local

```
; BIND data file for local loopback interface
$TTL    604800          Durée de vie en secondes par défaut d'un
                        enregistrement de ressource (RR) : ici 7 jours
@       IN      SOA     localhost. root.localhost. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                           3600 )       ; Negative Cache TTL
;                                       4 informations utiles pour
@       IN      NS      localhost.      le serveur secondaire
@       IN      A       127.0.0.1       (voir annexe 4)
@       IN      AAAA    ::1
                                        Durée de présence d'une
                                        réponse négative dans les caches
                                        suite à une question sur le
```

Outre la variable **$TTL** (obligatoire pour bind9), on peut spécifier au début du fichier les variables suivantes :

- **$ORIGIN** : Pour définir le nom de domaine à ajouter pour reconstituer le FQDN (est positionné par défaut au nom du domaine que le fichier de zone décrit).
- **$INCLUDE** : Pour indiquer le chemin d'accès d'un fichier à utiliser .

Chaque information élémentaire de la base de données DNS est un objet appelé "*resource record*" (RR) qui partage le format commun suivant (les éléments entre crochet sont facultatifs) :

[domaine] [ttl] [classe] type données

- **domaine** : nom de la zone auquel s'appliquent les entrées. S'il est omis, le RR s'applique au domaine du précédent RR. On peut aussi comme dans cet exemple utiliser le symbole @ qui remplace le nom de la zone telle que défini dans "named.conf" ou éventuellement dans la variable $ORIGIN
- **ttl** : définit le "time to live" ou durée de vie, c'est à dire le temps pendant lequel cette information peut rester en cache. C'est un nombre décimal sur 8 chiffres, qui indique des secondes. S'il est omis, sa valeur sera égale à la valeur par défaut écrite dans la variable $TTL.
- **classe** : il s'agit d'une classe d'adresses. Toujours IN pour les adresses IP, s'il n'y a aucun champ classe, c'est la classe du précédent RR qui s'applique.
- **type** : décrit le type du RR (les plus courants sont A, SOA, PTR et NS)
- **données** : contient les données associées au RR, les données dépendront du type du RR.

Les types de l'exemple :

L'enregistrement de type SOA (**S**tart **O**f **A**uthority – en français : responsable de la zone) est obligatoirement le premier : il donne les caractéristiques techniques générales de la zone. Il est suivi de l'adresse mél de l'administrateur qui ne comporte pas de signe « @ » (remplacé par un point).

L'enregistrement de type NS définit un serveur de nom pour le nom de domaine.

L'enregistrement de type A met en correspondance un nom de machine et une adresse IP.

L'enregistrement de type AAAA est utilisé dans les réseaux IPv6.

Le fichier db.127

```
; BIND reverse data file for local loopback interface

$TTL    604800
@       IN      SOA     localhost. root.localhost.
                       (
                             1         ; Serial
                        604800         ; Refresh
                         86400         ; Retry
                       2419200         ; Expire
                        604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.
1.0.0   IN      PTR     localhost.
```

L'enregistrement PTR permet de faire de la résolution inverse c'est-à-dire associer à une adresse IP un nom FQDN.

Un fichier db.exemple.fr

```
$TTL  86400
@               IN    SOA   ns1.exemple.fr. hostmaster.exemple.fr. (
                            2008113001  ;serial
                            86400       ;refresh
                            21600       ;retry
                            3600000     ;expire
                            3600 ) ;negative caching ttl
@               IN    NS          ns1.exemple.fr.
@               IN    NS          ns2.exemple.fr.
exemple.fr.     IN    MX          servmail

servmail        IN    A           192.168.1.200
ns1             IN    A           192.168.1.50
ns2             IN    A           192.168.1.100
servftp         IN    A           192.168.1.150
mail            IN    CNAME       servmail.exemple.fr.
ftp             IN    CNAME       servftp
```

L'enregistrement MX définit un serveur de mail. Attention, un MX ne peut pas être référencé par un alias.

L'enregistrement CNAME indique que le nom est un alias vers un nom canonique.

Remarque : les directives peuvent apparaître n'importe où dans le fichier.

Un fichier db.192.168.1.rev

```
$TTL  86400
@               IN    SOA   ns1.exemple.fr. hostmaster.exemple.fr. (
                            2008113001  ;serial
                            86400       ;refresh
                            21600       ;retry
                            3600000     ;expire
                            3600 ) ;negative caching ttl
@               IN    NS          ns1.exemple.fr.
@               IN    NS          ns2.exemple.fr.

200             IN    PTR         servmail.exemple.fr.
50              IN    PTR         ns1.exemple.fr.
100             IN    PTR         ns2.exemple.fr.
150             IN    PTR         servftp.exemple.fr.
```

La délégation de zone

La délégation de zone transfère la responsabilité d'un sous-domaine à un autre serveur de noms.

La délégation de zone est "déclarée" dans le fichier de zone du domaine parent par un enregistrement de type NS.

Et un type d'enregistrement A est ensuite nécessaire pour la correspondance entre l'adresse IP et le nom.

Exemple :

```
$TTL  86400
@                     IN    SOA   ns1.exemple.fr.
                                  hostmaster.exemple.fr. (
                                  ... )
@                                 IN    NS          ns1.exemple.fr.
...

delegation                    IN    NS    d1.delegation.exemple.fr.
d1.delegation.exemple.fr.     IN    A     192.168.1.30
```

Le fichier de zone du sous-domaine est un fichier de zone classique.

Annexe 3 : les listes de contrôle d'accès (*acl*) et les vues (*view*)

On rajoutera les listes de contrôle d'accès (ACL – *Access Control List*) dans le fichier `named.conf.options`.

Les vues permettent notamment d'affiner les options ; elles seront créées dans le fichier d'options à la suite des options globales.

Les listes de contrôle d'accès (ACL)

Le principe des ACL est le suivant : on crée des acl selon une syntaxe définie puis on "utilise" les ACL créées au niveau des options et/ou des fichiers de zone.

La syntaxe pour créer les acl est la suivante :

```
acl nom_acl {
<adresseIP> | <classe-d'adresse> | <mot_cle>;
...;
};
```

Exemples :

```
        acl reseauExterne {
                !localnets;
                !localhost;
        };

        acl servEsclave {
                192.168.1.150;
        };
```

Puis, par exemple, dans une déclaration de zone :

```
allow-transfer { servEsclave; };
```

Les vues (*views*)

Les vues permettent de modifier le comportement du serveur en fonction de l'adresse IP du client. Ainsi, il est possible par exemple de déclarer le serveur en serveur récursif pour certaines adresses IP et en serveur itératif pour d'autres (attention, ceci peut aussi être réalisé avec la directive *allow-recursif*).

La syntaxe pour créer une vue est la suivante :

```
view « nomdelavue » {
        match-clients { <classe-d'adresse> | <nom_acl> | « any » | « localnets » };
        <une série d'options>;
        }
```

Les clients DNS dont l'adresse IP correspond à celle donnée dans la directive "*match-clients*" seront affectés par les options DNS qui suivent.

Exemple:

```
        view "externe" {
                match-clients { reseauExterne ; };
                recursion no;
                include "/etc/bind/db.exemple.fr";
        };
```

Annexe 4 : Le serveur secondaire

Un serveur de noms secondaire ne gère pas directement les informations sur les zones mais les obtient à partir du serveur de noms principal de la zone (ou d'un autre serveur secondaire) via le réseau (**transfert de zone**). Un serveur secondaire ne peut modifier des données de la zone mais il a lui aussi autorité sur la zone.

Cette redondance permet une meilleure tolérance aux pannes et une réduction de la charge de travail des serveurs principaux.

Concrètement, un serveur secondaire peut être mis à jour de deux façons :

- en fonction de la valeur '*refresh*' définie dans le SOA de la zone ;
- ou lorsqu'il reçoit une notification du primaire : actif par défaut (l'option "*notify*" est par défaut à "*yes*") sur bind9.

Quand il démarre, un serveur secondaire doit connaître son **serveur maître** pour entamer un transfert de zone avec ce serveur.

Dans le fichier **named.conf.local du serveur secondaire :**

```
zone "exemple.fr" {
      type slave;
      file "slave/db.exemple.fr";
      masters { 192.1.1.1; };
};
```

Il est préférable de créer un répertoire que l'on nommera "slave" dans le répertoire d'accueil des fichiers de zone pour les zones secondaires pour deux raisons :

- cela identifie clairement les zones secondaires, d'autant plus qu'un serveur peut être à la fois maître sur certaines zones et esclave sur d'autres.
- Cela réduit les problèmes de sécurité car comme le démon named fonctionne sous l'autorité de l'utilisateur bind et le groupe bind, il faut rendre le répertoire dans lequel seront stockés les fichiers de zone secondaires accessible en écriture ; cela limite donc au minimum la portée de ces droits.

Sur le serveur principal, il est préférable de limiter le transfert de zone au(x) seul(s) serveur(s) secondaire(s).

```
zone "exemple.fr" {
      type master;
      file "db.exemple.fr";
      allow-transfer { 192.168.1.100; };
};
```

Puis, il faut **ajouter dans chaque fichier de zone à exporter (donc sur le primaire)** un **enregistrement "NS" pour chaque serveur esclave** (l'option activée par défaut "*notify yes*" agit d'ailleurs uniquement sur l'adresse IP renseignée dans ces enregistrements).

Lorsque le serveur secondaire démarre pour la première fois, **il crée automatiquement son fichier de zone à partir du serveur primaire**. Attention, il faut que le groupe système "bind" ait le droit d'écrire dans le répertoire "/var/cache/bind/slave"... ce qui n'est certainement pas le cas puisque vous n'avez pu le créer que sous l'utilisateur "root". Dans ce cas : `chmod g-w /var/cache/bind` (les fichiers de la zone secondaire étant dans "slave", on peut enlever, pour ce répertoire, ce droit au groupe bind)

```
chgrp bind /var/cache/bind/slave
chmod g+w /var/cache/bind/slave
```

db.exemple.fr

```
$TTL  86400
exemple.fr.  IN   SOA   ns1.exemple.fr.  hostmaster.exemple.fr. (

                  2008113001  ;serial

                  86400       ;refresh

                  21600       ;retry

                  3600000     ;expire


                  3600 ; negative caching ttl )

@                       IN    NS          servSec.exemple.fr.
servSec                 IN    A           192.168.1.100

...
```

Numéro de série (*serial*) : Identifie la version de la zone ; quand on modifie le fichier de zone, on incrémente ce numéro. Le format conseillé est le suivant : YYYYMMDDxx.

Rafraîchissement (*refresh*) : intervalle en secondes destiné au serveur secondaire pour rafraîchir son fichier de zone (nombre décimal entier sur 8 chiffres). Cette valeur peut être élevée si on a maintenu l'option "*notify yes*" au niveau du serveur maître.

Tentatives (*retry*) : intervalle en secondes avant de recontacter le serveur principal en cas d'échec de la demande de rafraîchissement.

Expiration (expire) : indique le temps en secondes, au bout duquel un serveur secondaire doit éliminer toutes les informations de zone s'il n'a pas pu contacter le serveur (cette valeur doit être élevée).

Toutes les "*refresh*" secondes, le serveur secondaire transfère le SOA de la zone et vérifie si le numéro de série a augmenté ; si c'est le cas le transfert de zone a lieu. En cas d'échec de cette interrogation, le serveur secondaire recommence toutes les "*retry*" secondes jusqu'à atteindre le temps d'expiration (*expire*).

Mais si l'on a maintenu, sur le serveur primaire, l'option par défaut de notification, le serveur maître notifiera immédiatement tout changement de son fichier de zone au serveur secondaire.

Attention : à chaque modification du fichier de zone, il ne faut pas oublier d'augmenter le numéro de série et de faire une relecture par le démon du fichier de configuration.

Annexe 5 – Tester un serveur de noms

Une des conditions de réussite du TP est de pouvoir faire un ping :

- sur chacune des trois machines déclarées dans **tous les domaines** affichés au tableau depuis tout poste correctement configuré de la salle de TP, par exemple : ping www.mondomaine.org ; ping ftp.mondomaine.org ; ping mail.mondomaine.org
- sur chacune des machines déclarées sur une autre plate-forme.

Les outils dont on dispose : le paquet dnsutils (normalement installé par défaut sur tout système debian) fournit quelques commandes utiles :

- Pour vérifier les fichiers de configuration globale : **`named-checkconf`**

Si la commande ne renvoie rien, c'est qu'il n'y a pas d'erreurs de syntaxe dans les fichiers.

- Pour vérifier la syntaxe des fichiers de zone : **`named-checkzone zonename filename`**

Il ne faut pas toujours se fier au "OK" final mais lire les phrases qui précèdent...

- **Les commandes `nslookup`, `host` et `dig`** permettent de vérifier les données relatives à chaque zone.

Mais pour ne pas perdre de temps, il est important de procéder méthodiquement.

Première batterie de tests : le serveur bind 9 a-t'il correctement démarré ou redémarré ?

ATTENTION, SAUF s'il y a une erreur de syntaxe dans les fichiers de configuration globaux, le serveur va apparemment démarrer ou redémarrer sans erreur. Pour voir si tout c'est réellement bien passer, il faut lire les fichiers de trace (*logs*).

Lorsque vous lancez ou relancez le serveur, il est judicieux d'ouvrir une autre console, d'y lancer la commande "tail -f /var/log/syslog" pour lire en direct les fichiers de trace concernant le démon *named*.

Voici par exemple 2 cas où il y a un problème :

Premier cas : le fichier de zone n'est pas trouvé :

```
Jan 26 16:01:33 ns1 named[21056]: zone domaine-grp1.com/IN: loading from master file db.domaine-grp1.com failed: file not found
```

Il n'y a pas ici une erreur de syntaxe, la commande `named-checkconf` n'aurait rien renvoyée.

Deuxième cas : le fichier de zone est trouvé mais présente une erreur :

```
Jan 26 16:13:57 ns1 named[21652]: zone domaine-grp1.com/IN: loading from master file /etc/bind/db.domaine-grp1.com failed: unexpected end of input
```

Une erreur se situe ici dans le fichier de configuration de la zone.

On a le détail des erreurs avec la commande :
`named-checkzone zonename filename /var/cache/bind/db.domaine-grp1.com`

Erreurs courantes :

- les nombreux ";" des fichiers de configuration globale ;
- les noms DNS pleinement qualifiés dans les fichiers de zone non suffixés par un point ;
- les parenthèses mal placées de l'enregistrement SOA ;

Sous linux, il est conseillé d'utiliser l'éditeur `vim` ; les "couleurs" vous guideront... (pour bénéficier des couleurs syntaxiques, écrivez (ou décommentez) "*`syntax on`*" dans le fichier de configuration `/etc/vim/vimrc`)

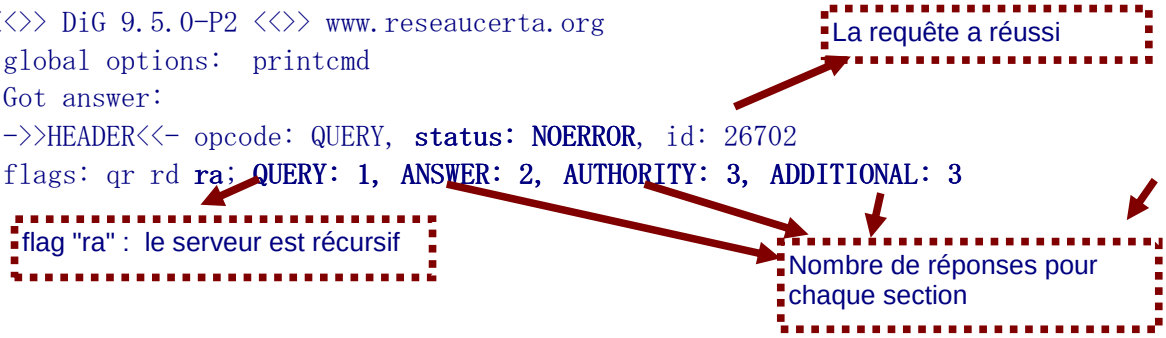
– La non relecture de bind9 lorsqu'un fichier de configuration est modifié et le non vidage du cache.

Deuxième batterie de tests : vérification des données relatives à chaque zone.

L'utilisation des commandes `nslookup`, `host` ou `dig` vous permettra de vérifier les données relatives à chaque zone.

Nous ne détaillerons ici que la commande dig (**D**omain **I**nformation **G**roper) qui donne beaucoup de détails... À l'inverse de `nslookup`, cette commande n'est pas interactive.

```
dig www.reseaucerta.org

; <<>> DiG 9.5.0-P2 <<>> www.reseaucerta.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26702
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION: (1)
;www.reseaucerta.org.            IN      A

;; ANSWER SECTION: (2)
www.reseaucerta.org.     1785    IN      CNAME   strasbourg.reseaucerta.org.
strasbourg.reseaucerta.org.  3585 IN     A       130.79.130.89

;; AUTHORITY SECTION: (3)
reseaucerta.org.         10785   IN      NS      c.dns.gandi.net.
reseaucerta.org.         10785   IN      NS      b.dns.gandi.net.
reseaucerta.org.         10785   IN      NS      a.dns.gandi.net.

;; ADDITIONAL SECTION: (4)
a.dns.gandi.net.         117683  IN      A       217.70.179.40
b.dns.gandi.net.         117683  IN      A       217.70.184.40
c.dns.gandi.net.         117683  IN      A       217.70.182.20

;; Query time: 2 msec (5)
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Wed Jan 28 12:20:18 2009
;; MSG SIZE  rcvd: 187
```



(1) La section QUESTION reprend la requête émise.

(2) La section ANSWER donne la réponse à la requête.

(3) La section AUTHORITY donne les serveurs de noms ayant autorité sur la zone.

(4) La section ADDITIONAL donne les adresses IP des serveurs de noms autoritaires.

(5) La section Query time donne le temps de réponse de la requête. Cette valeur indique donc si la réponse est en cache ou pas.

Pour filtrer les résultats, on peut préciser, dans la requête, les différents types de RR :
dig www.reseaucerta.org NS

dig www.reseaucerta.org A
dig www.reseaucerta.org MX
dig -x 192.168.1.124
Etc.

L'option +trace de la commande dig permet de faire la recherche en parcourant l'arborescence depuis la racine jusqu'à la réponse.

```
dig +trace www.domaine-grp1.com

; <<>> DiG 9.5.0-P2 <<>> +trace www.domaine-grp1.com
;; global options:  printcmd
.                       3600000 IN      NS      A.ROOT-SERVERS.NET.
;; Received 48 bytes from 192.168.1.2#53(192.168.1.2) in 0 ms

domaine-grp1.com.       3600    IN      NS      ns.domaine-grp1.com.
;; Received 72 bytes from 192.168.1.124#53(A.ROOT-SERVERS.NET) in 14 ms

www.domaine-grp1.com.  86400 IN      CNAME      ns.domaine-grp1.com.
ns1.domaine-grp1.com.  86400 IN      A         192.168.10.10
domaine-grp1.com.      86400 IN      NS         ns.domaine-grp1.com.
;; Received 72 bytes from 192.168.10.10#53(ns.domaine1-grp1.com) in 3 ms
```

Observons maintenant une requête qui fournit une réponse négative.

```
dig www.domaine-grp1.com

; <<>> DiG 9.5.0-P2 <<>> www.domaine1-grp1.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 9054
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;www.domaine-grp1.com.          IN      A

;; AUTHORITY SECTION:
domaine-grp1.com.           3600       IN          SOA         ns.domaine-grp1.com.
hostmaster.domaine-grp1.com. 20090126 86400 21600 3600000 3600

;; Query time: 3 msec
;; SERVER: 192.168.1.2#53(192.168.1.2)
;; WHEN: Wed Jan 28 15:31:44 2009
;; MSG SIZE  rcvd: 97
```

Il n'y a pas d'enregistrement ayant le nom demandé dans cette zone.

Il existe un autre type de réponse négative : NODATA qui indique qu'aucune donnée pour le triplet (nom, type, classe) demandé n'existe ; mais il existe d'autres enregistrements possédant ce nom, mais de type différent.

Si l'on trouve, après **status,** le mot clé **SERVFAIL**, il n'y a pas de réponse et il faut revenir à la première batterie de test car le serveur a mal démarré.

Attention : cela ne vient pas forcément de votre serveur mais d'un des serveurs interrogés de manière récursive.

Remarque : attention au cache qui pourrait fournir des réponses qui ne sont plus valables ; la commande `rndc flush` permet de supprimer toutes données en cache.

Troisième batterie de tests : vérification de la non récursivité d'un serveur DNS

Vous devez configurer le client DNS d'un poste n'appartenant pas à votre réseau sur votre serveur DNS qui n'accepte pas la récursivité externe.

Une même requête doit être :

- **refusée si demande récursive à partir d'un poste n'appartenant pas au réseau**

On remarque l'absence du flag "ra" : le serveur n'est pas récursif et il refuse la

```
dig www.domaine-grp1.com
; <<>> DiG 9.5.0-P2 <<>> www.domaine-grp1.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 20371
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.domaine-grp1.com.                  IN      A

;; Query time: 0 msec
;; SERVER: 192.168.1.2#53(192.168.1.2)
;; WHEN: Wed Jan 28 19:06:41 2009
;; MSG SIZE  rcvd: 32
```

- **acceptée si provenant du réseau interne** :

```
dig www.domaine-grp1.com

; <<>> DiG 9.5.0-P2 <<>> www.domaine-grp1.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12560
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;www.domaine-grp1.com.                  IN      A

;; ANSWER SECTION:
www.domaine-grp1.com.      86233        IN    CNAME    ns.domaine-grp1.com.
ns1.domaine-grp1.com.      86233  IN      A        192.168.10.10

;; AUTHORITY SECTION:
domaine-grp1.com.      86233    IN      NS      ns.domaine-grp1.com.

;; Query time: 17 msec
```

```
;; SERVER: 192.168.1.2#53(192.168.1.2)
;; WHEN: Wed Jan 28 19:14:01 2009
;; MSG SIZE  rcvd: 90
```

Quatrième batterie de tests : contrôle des transferts de zone

Il suffit en fait de lire les fichiers de trace (*logs*) lorsque vous lancez pour la première fois votre serveur secondaire ou que vous modifiez le contenu d'un fichier de zone (sans oublier de modifier le numéro de série).

Dans les fichiers de trace **du serveur secondaire**, vous devriez trouver quelque chose de ce style :

```
Jan 29 16:15:26 nsd named[20736]: zone domaine-grp1.com/IN: Transfer started.
Jan 29 16:15:26 nsd named[20736]: transfer of 'domaine-grp1.com/IN' from 192.168.1.2#53: connected using 192.168.10.100#56170
Jan 29 16:15:26 nsd named[20736]: zone domaine-grp1.com/IN: transferred serial 20090126
Jan 29 16:15:26 nsd named[20736]: transfer of 'domaine-grp1.com/IN' from 192.168.10.10#53: Transfer completed: 1 messages, 7 records, 225 bytes, 0.015 secs (15000 bytes/sec)
Jan 29 16:15:26 nsd named[20736]: zone domaine-grp1.com/IN: sending notifies (serial 20090126)
```

Sur le serveur maître :

```
Jan 29 16:21:29 ns1 named[8190]: client 192.168.10.100#56170: transfer of 'domaine-grp1.com/IN': AXFR started
Jan 29 16:21:29 ns1 named[8190]: client 192.168.10.100#56170: transfer of 'domaine-grp1.com/IN': AXFR ended
```

Vous pouvez maintenant vérifier que les fichiers de zones sur le serveur secondaire ont bien été créés.

L'erreur fréquente consiste à ne pas accorder le droit d'écrire au groupe système bind sur le répertoire /var/cache/bind/slave. C'est certainement le cas si vous avez les traces suivantes sur le serveur secondaire :

```
Jan 29 16:20:23 ns1 named[20970]: dumping master file: /var/cache/bind/slave/tmp-FqJO3iWCxz: open: permission denied
Jan 29 16:20:23 ns1 named[20970]: transfer of 'domaine-grp1.com/IN' from 192.168.10.10#53: failed while receiving responses: permission denied
```

Pour tester le serveur secondaire, vous devez configurer les DNS du client de manière à ce qu'il pointe vers le serveur secondaire.

Annexe 6 : pistes pour Windows 2003 Server

On renvoie ici aux pages Technet utiles à la réalisation de ce TP avec des serveurs DNS installés sous Windows 2003 Server.

Installer un serveur DNS
http://technet.microsoft.com/fr-fr/library/cc782017.aspx

Sous Windows créer un domaine sur lequel on a autorité revient à créer une « zone de recherche directe » et pour gérer une zone « in-addr.arpa » associée il faut créer une « zone de recherche inversée »

Pour créer une « zone de recherche directe »
http://technet.microsoft.com/fr-fr/library/cc782017.aspx

Pour créer une « zone de recherche inversée » (ou indirecte)
http://technet.microsoft.com/fr-fr/library/cc783250.aspx

Configurer les propriétés de zone
http://technet.microsoft.com/fr-fr/library/cc757732.aspx
(Remarque : attention il peut être important de modifier certaines propriétés comme par exemple le nom du serveur DNS et le SOA qui sont initialisés généralement par windows avec le nom Netbios du poste).

Ajouter un enregistrement de ressource d'hôte (A)
http://technet.microsoft.com/fr-fr/library/cc779029.aspx

Ajouter un enregistrement de ressource de messagerie (MX)
http://technet.microsoft.com/fr-fr/library/cc779227.aspx

Ajouter un alias (CNAME)
http://technet.microsoft.com/fr-fr/library/cc776292.aspx

Ajouter un enregistrement de ressource pointeur (PTR) à une zone indirecte
http://technet.microsoft.com/fr-fr/library/cc775703.aspx
(Remarque : un enregistrement pointeur peut être créé automatiquement à partir de la création d'un enregistrement ressource dans une zone directe)

Serveurs racines ou notion de « redirecteur » avec windows
http://technet.microsoft.com/fr-fr/library/cc782142.aspx

Configurer un DNS pour qu'il utilise des redirecteurs
http://technet.microsoft.com/fr-fr/library/cc773370.aspx

Créer une délégation de zone
http://technet.microsoft.com/fr-fr/library/cc785881.aspx

Ajouter un serveur secondaire pour une zone
http://technet.microsoft.com/fr-fr/library/cc757524.aspx

Utilisation de nslookup avec Windows (dig n'existe pas sous windows nativement il faut l'installer à partir par exemple de http://pigtail.net/LRP/dig/)
http://technet.microsoft.com/fr-fr/library/cc756097.aspx

Configurer DNS pour des clients windows
http://technet.microsoft.com/fr-fr/library/cc738308.aspx

Désactiver le recursivité sous windows
http://technet.microsoft.com/fr-fr/library/cc787602.aspx
(Remarque : c'est donné à titre d'information, car en réalité cela supprime la redirection. On ne peut pas mettre en œuvre des ACLs avec le DNS windows comme avec Bind9. On ne peut mettre en oeuvre des problématiques de sécurité qu'en liaison avec Active Directory (voir le DACLs). Voir notamment la page suivante :
http://technet.microsoft.com/fr-fr/library/cc783606.aspx.

Fichiers générés par le DNS windows
L'interface graphique ou les lignes de commandes windows génèrent des fichiers textes consultables généralement par le chemin suivant *racineSysteme\system32\drivers.* On y trouvera aussi des fichiers logs et on trouvera aussi des informations dans le gestionnaire d'évènements

Conseil : il est préférable de faire ce TP sur un serveur autonome et non un contrôleur disposant d'une « *active-directory* » pour éviter les interactions avec cette dernière. Si cela n'est pas possible, on peut cependant dissocier les zones créées et « *l'active directory* » en agissant sur les propriétés de la zone.

Éléments de correction

Pour ces éléments de correction, on suppose que l'élève dispose de 3 postes (à adapter donc selon vos plate-formes) :
Serveur maître : 192.168.10.10
Serveur maître de la délégation : 192.168.10.100
Autre poste du réseau : 192.168.10.11

Rappel :

- le fichier `named.conf` n'est modifié sur aucun serveur de noms ;
- les fichiers de zone primaire seront créés dans `/var/cache/bind`;
- les fichiers de zone secondaire seront créés dans `/var/cache/bind/slave`.

Le poste du professeur (adresse IP : 192.168.1.124) :

Fichier de configuration principale `named.conf.local` (le professeur gère la racine)

```
zone "." {
        type master;
        file "/var/cache/bind/db.root-servers.net";
};
```

Fichier de zone pour la racine (".")

Base de données de la racine pour le serveur de l'enseignant enrichie au fur et à mesure des déclarations de domaines :

```
$TTL 1H
.                   in soa root-servers.net.  root.root-servers.net.  (
                          2008093003
                          3H
                          1H
                          1W
                          1H )

.                         in ns        a.root-servers.net.
a.root-servers.net. in a          192.168.1.124

; domaine déposé par le groupe1 (2 dns officiels - 1 dns primaire et 1 dns
secondaire)
domaine-grp1.com.   in ns        ns.domaine-grp1.com.
domaine-grp1.com.   in ns        serveurSecondaire.domaine-grp1.com.

10.168.192.in-addr.arpa.   in ns  ns.domaine-grp1.com.
10.168.192.in-addr.arpa.   in ns  serveurSecondaire.domaine-grp1.com.

ns.domaine-grp1.com.                          in a         192.168.10.10
serveurSecondaire.domaine-grp1.com.     in a         192.168.10.100

; domaine déposé par le groupe2 (1 seul dns officiel)
gupi.net.                         in ns        ns1.gupi.net.
```

```
20.168.192.in-addr.arpa.   in ns        ns1.gupi.net.
ns1.gupi.net.                    in a         192.168.20.10
```

Sur le serveur maître (adresse IP :192.168.10.10)

Rappel : ce serveur fait aussi office de serveur secondaire du serveur maître de la délégation.

On suppose que le serveur maître de la délégation (qui fera aussi office de serveur secondaire) a pour adresse IP 192.168.10.100.

Fichier "db.root" qui répertorie les serveurs de la racine (en l'espèce, un seul serveur celui du professeur) :

```
.                            3600000   IN   NS     A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.          3600000        A      192.168.1.124
```

Fichier de configuration principale `named.conf.local` et `named.conf.options`
(en gras, les directives pour les serveurs secondaires)

On part du principe que nous ne disposons que de deux postes : le serveur primaire est donc aussi serveur secondaire pour les zones `intranet.domaine-grp1.com` et `extranet.domaine-grp1.com`.

```
// Déclaration des zones autoritaires
zone "domaine-grp1.com" {
      type master;
      file "db.domaine-grp1.com";
      allow-transfer { 192.168.10.100; };
};

zone "10.168.192.in-addr.arpa" {
        type master;
        file "db.192.168.10.rev";
};

// Déclaration des zones secondaires
zone "intranet.domaine-grp1.com" {
      type slave;
      file "slave/db.intranet.domaine-grp1.com";
      masters { 192.168.10.100; };
};

zone "extranet.domaine-grp1.com" {
      type slave;
      file "slave/db.extranet.domaine-grp1.com";
      masters { 192.168.10.100; };
};
```

Fichier de conf pour les options et ACL `named.conf.options`

Pour les versions de bind inférieures à la version 9.4, une **solution simple pour empêcher les requêtes récursives** provenant des réseaux "externes" est de permettre la récursion pour le réseau interne (ceci empêche de facto les requêtes récursives pour les autres réseaux). Ceci est fait, par défaut, à partir de la version 9.4.

```
options {
        directory "/var/cache/bind";
        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };
        allow-recursion { reseauInterne; };
};

//Les ACLS
acl reseauInterne {
        localnets;
        localhost;
};
// ACL facultative
acl reseauExterne {
        !localnets;
        !localhost;
        };
```

Fichier de zone pour domaine-grp1.com (avant délégation et serveurs secondaires)

```
$TTL    86400
@      IN      SOA     ns.domaine-grp1.com.  hostmaster.domaine-grp1.com.

( 20090126    ;serial
                                        86400       ;refresh
                                        21600       ;retry
                                        3600000     ;expire
                                        3600 ); negative caching ttl

@                   IN          NS          ns.domaine-grp1.com.
@                   IN          MX          serveurdebian.domaine-grp1.com.
ns                  IN          A           192.168.10.10
serveurDebian       IN          A           192.168.10.11
www                 IN          CNAME       serveurDebian.domaine-grp1.com.
ftp                 IN          CNAME       serveurDebian.domaine-grp1.com.
mail                IN          CNAME       serveurDebian.domaine-grp1.com.
```

Fichier de zone pour **192.168.10.rev**

```
$TTL    86400
@      IN      SOA     ns.domaine-grp1.com.  hostmaster.domaine-grp1.com.

                                        (20090126    ;serial
                                        86400       ;refresh
                                        21600       ;retry
```

```
                                        3600000     ;expire
                                        3600 ); negative caching ttl

@           IN     NS              ns.domaine-grp1.com.
@           IN     MX                  serveurdebian.domaine-grp1.com.
10          PTR                        ns.domaine-grp1.com.
11          PTR                        serveurDebian.domaine-grp1.com.
100         PTR                        serveurSecondaire.domaine-grp1.com.
```

Fichier de zone pour domaine-grp1.com (après délégation et serveurs secondaires)
(en gras les RR nécessaires à la délégation et en italique les RR nécessaires au serveur secondaire)

```
$TTL    86400
@      IN      SOA     ns.domaine-grp1.com.  hostmaster.domaine-grp1.com.

                        ( 20090126    ;serial
                                        86400       ;refresh
                                        21600       ;retry
                                        3600000     ;expire
                                        3600 ); negative caching ttl

@                   IN          NS          ns.domaine-grp1.com.
@                   IN          MX          serveurDebian.domaine-grp1.com.

intranet            IN          NS          nsd.intranet.domaine-grp1.com.
extranet            IN          NS          nsd.extranet.domaine-grp1.com.

nsd.intranet.domaine-grp1.com.  IN     A       192.168.10.100
nsd.extranet.domaine-grp1.com.  IN     A       192.168.10.100

serveurSecondaire           IN     A       192.168.10.100
@                   IN     NS          serveurSecondaire.domaine-grp1.com.

ns                  IN     A           192.168.10.10
serveurDebian       IN     A           192.168.10.11
mail                IN     CNAME       serveurDebian.domaine-grp1.com.
```

Le serveur secondaire doit être ajouté dans le fichier de zone de la racine géré par le professeur.

Sur le serveur maître de la délégation (adresse IP :192.168.10.100)

Rappel : ce serveur fait aussi office de serveur secondaire de la zone principale.

Fichier "db.root" qui répertorie les serveurs de la racine (en l'espèce, un seul serveur, celui du professeur) :

```
.                            3600000   IN   NS     A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.        3600000        A      192.168.1.124
```

Fichier de configuration principale `named.conf.local` et `named.conf.options`
(en gras, les directives nécessaires pour les serveurs secondaires)

```
// Déclaration des zones autoritaires
zone "intranet.domaine-grp1.com" {
      type master;
      file "intranet.db.domaine-grp1.com";
      allow-transfer { 192.168.10.10; };
};

// Déclaration des zones autoritaires
zone "extranet.domaine-grp1.com" {
      type master;
      file "extranet.db.domaine-grp1.com";
      allow-transfer { 192.168.10.10; };
};

// Déclaration des zones esclaves
zone "domaine-grp1.com" {
      type slave;
      file "slave/db.domaine-grp1.com";
      masters { 192.168.10.10; };
};

zone "10.168.192.in-addr.arpa" {
        type slave;
        file "slave/db.192.168.10.rev";
};
```

Fichier de conf pour les ACL `named.conf.options`

```
options {
        directory "/etc/bind";
        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };
        allow-recursion { reseauInterne; };
};

//Les ACLS
acl reseauInterne {
        localnets;
        localhost;
};
```

```
// ACL facultative
acl reseauExterne {
        !localnets;
        !localhost;
        };
```

Fichier de zone pour intranet.domaine-grp1.com
(en gras les RR nécessaires au serveur secondaire)

```
$TTL    86400
$ORIGIN intranet.domaine-grp1.com.
@       IN      SOA     nsd.intranet.domaine-grp1.com.  hostmaster.intranet.  domaine-
grp1.com.

        ( 20090126      ;serial
                                        86400       ;refresh
                                        21600       ;retry
                                        3600000     ;expire
                                        3600 ); negative caching ttl

@                   IN          NS          nsd.intranet.domaine-grp1.com.
nsd                 IN          A           192.168.10.100

@                           IN      NS      servSec.intranet.domaine-grp1.com.
servSec                     IN      A       192.168.10.10

serveurDebian               IN          A           192.168.10.11
www                         IN          CNAME       serveurDebian
ftp                         IN          CNAME       serveurDebian
support                     IN          CNAME       serveurDebian
```

Fichier de zone pour extranet.domaine-grp1.com
(en gras les RR nécessaires au serveur secondaire)

```
$TTL    86400
$ORIGIN extranet.domaine-grp1.com.
@       IN      SOA     nsd.extranet.domaine-grp1.com.  hostmaster.extranet.  domaine-
grp1.com.

        ( 20090126      ;serial
                                        86400       ;refresh
                                        21600       ;retry
                                        3600000     ;expire
                                        3600 ); negative caching ttl

@                   IN          NS          nsd.extranet.domaine-grp1.com.
nsd                 IN          A           192.168.10.100

@                   IN      NS          servSec.extranet.domaine-grp1.com.
servSec             IN      A           192.168.10.10
```

```
serveurDebian           IN          A           192.168.10.11
www                     IN          CNAME       serveurDebian
ftp                     IN          CNAME       serveurDebian
clients                 IN          CNAME       serveurDebian
fournisseurs        IN          CNAME       serveurDebian
```