Le DNS

DNS : Domain Name System

Permet d'établir la correspondance nom de machine -> adresse IP

Un serveur de nom de domaine (Domain Name System) est une base de données en ligne pour résoudre en adresses IP les noms de domaine complets (FQDN, Fully Qualified Domain Names, cf. plus loin) et d'autres noms d'hôtes.

Le DNS est décrit par les RFC 1034 et 1035.
C'est une BD client serveur distribuée et hiérarchisée. Il fonctionne au niveau de la couche application et utilise UDP ou TCP comme protocole sous-jacents.

Comme Wins, DNS traduit les noms d'ordinateurs en adresse IP.

*Les clients DNS sont appelés solveurs (***resolvers***) et les serveurs : serveurs de noms (name servers).*
Les clients envoient des requêtes avec UDP et n'utilisent TCP qu'en cas de problème.
Si le serveur de noms n'est pas en mesure de satisfaire la requête, il peut la renvoyer à un autre serveur susceptible d'y répondre (cf. requête **itérative** *ou* **récursive***)*

Notion de domaine
(Attention un domaine DNS est un domaine de nommage : différent d'un domaine NT)
Les serveurs sont regroupés dans différents niveaux appelés domaines.
Les domaines sont structurés de façon hiérarchique.
Au plus haut niveau on trouve un domaine racine référencés généralement par un point (autrement label null).
En dessous se trouvent des domaines appelés domaines supérieurs.

- *com*
- *edu*
- *org*
- *net*
- *gov*
- *mil*
- *num*
- *arpa*
- *xx (code de 2 lettres correspondant à un pays)*

En dessous se trouveront soit directement des machines hôtes, soit des sous-domaines et ainsi de suite. La dernière extrémité d'une branche est donc généralement un nom d'hôte.
Les noms d'hôtes au sein d'un domaine sont ajoutés au début de ce nom de domaine. La combinaison nom d'hôte et nom de domaine est appelée FQDN
Exemple : linux.beaune.infop.fr désignerait la machine linux du sous-domaine beaune, du sousdomaine infop du domaine fr
Un nom d'hôte doit être unique dans son domaine, on peut donc avoir une machine qui s'appellerait linux dans un autre domaine , par exemple : linux.lesalpes.fr

Attention *: si un nom d'hôte correspond à une machine (à une adresse IP en fait, une machine pouvant avoir quelquefois plusieurs adresses IP), une machine peut avoir plusieurs noms d'hôtes, donc apparaître sous des noms différents dans des domaines différents (la vue logique de l'espace de nommage ne se superpose pas à l'organisation physique des machines).*

Zone d'autorité

Comment sont gérés les noms par un serveur de noms dans une telle structure ?
On pourrait imaginer simplement un serveur de nom par nœud de l'arborescence hiérarchique, gérant les noms des niveaux strictement inférieurs. Mais une telle organisation multiplierait les serveurs de noms et par voie de conséquence l'appel à ces serveurs.
Une solution plus optimisée, plus souple, mais plus complexe a été développée.
Un serveur de noms ne s'occupe pas d'un nœud de l'arborescence, mais d'un ensemble de nœuds sur lequel il aura autorité. C'est à dire qu'il gérera l'attribution des noms et résoudra les noms. On dit que le

serveur gère une zone d'autorité (un barbarisme est souvent utilisé: serveur de noms autoritatif, qui est une traduction directe de Authoritative Name Server).
Une zone d'autorité recouvre au moins un domaine, appelé domaine racine de la zone et peut inclure des sous-domaines mais pas forcément tous les sous-domaines .

Prenons un exemple :
Ici nous avons une hiérarchie de domaines. Nous placerons un serveur de noms sur le domaine INFOP qui aura autorité sur BTSCI et BTSPMEPMI, c'est à dire que les résolutions d'adresses de type Kant.Btsci.info et Descartes.btspmepmi.info seront gérées par lui.
Nous placerons un autre serveur de noms sur btsig qui gérera pour tous les sous-domaines, donc les résolutions de type : Goedel.dev.btsig.info, Escher.res.btsig.info
Une zone d'autorité est donc une portion de l'arborescence (on dit une portion de l'espace de noms) mis sous la responsabilité d'un serveur de noms. Mais un serveur de noms peut avoir la responsabilité de plusieurs zones d'autorité. Les informations relatives à chaque zone seront stockées dans des fichiers distincts.

Il existe deux types de zone :

- *les zones* **directes** *permettant la résolution directe (trouver l'adresse correspondant à un nom). Le nom correspond au nom DNS (ex : mondomaine.fr)*
- *les zone inverses permettant de trouver le nom correspondant à une adresse. Ce sont des sous-zones de la zone* **in-addr.arpa.** *Par exemple un sous réseau en 192.168.1.0/24 aura un nom de zone inverse 1.168.192.in-addr.arpa. Le fichier de zone comportera le ou les octets manquants permettant de résoudre complètement le nom. Un domaine peut avoir plusieurs zones inverses (une par sous-réseau)*

Les différents types de serveurs de noms :

- **Serveur maitre***: il gère directement une ou plusieurs zones dans des fichiers locaux et récupère les informations dans ceux-ci*
- **Serveur esclave** *: il ne gère pas directement les informations sur les zones, mais les obtient à partir du serveur de noms maître de la zone via le réseau (transfert de zone). Cette redondance permet une meilleure tolérance aux pannes, une réduction de charge des serveurs principaux*
  Un serveur de noms esclave demande donc le transfert du fichier de zone à un autre serveur. Cet autre serveur peut être soit directement le serveur maître soit un serveur esclave intermédiaire. Dans tous les cas du point de vue du serveur esclave, il s'agira d'un serveur maître.
  Quand il démarre, un serveur esclave doit connaître son serveur maître pour entamer un transfert de zone avec ce serveur.
- *Une autre notion existe celle de* **serveur cache***.*
  Un serveur cache n'est ni un serveur maître, ni un serveur esclave : il n'a aucune autorité et ne fait aucun transfert de zone. Il se contente de mettre en cache les résolutions de noms qu'il effectue en transmettant des requêtes à d'autres serveurs de noms. Il ne contient donc que les informations qu'il a placé en cache après ces résolutions. Son intérêt est d'éviter le transfert de zone.

La base de données DNS

Chaque information élémentaire de la base de données DNS est un objet appelé **"ressource record" RR** *en abrégé. Chaque enregistrement est associé à un type décrivant le genre de données qu'il représente et une classe spécifiant le type de réseau auquel il s'applique.*

Les RR partagent le format commun suivant :
[domaine] [ttl] [classe] type données

- **Domaine** *: nom de domaine auquel s'applique les entrées. S'il est omis, le RR s'applique au domaine du précédent RR*
- **ttl** *: définit le "time to live", 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*
- **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*

Nous allons maintenant décrire les types les plus courants et leurs données associées :

SOA signifie "Start Of Authorithy", *l'enregistrement qui suit contient les informations ayant autorité sur le domaine.*

- **Origine** *: nom canonique du serveur de noms primaire pour ce domaine*
- **Contact** *: adresse électronique de la personne responsable du domaine (le signe @ est remplacé par un point ex:Albert.Dupont@mondomaine.fr s'écrira Albert.Dupont.mondomaien.fr)*
- **Numéro de série** *: numéro de version du fichier de zone, quand on modifie le fichier de zone, on incrémente ce numéro (Sert aux serveurs esclaves pur le transfert de zone)*
- **Rafraîchissement** *: intervalle en secondes destiné au serveur secondaire pour rafraîchir son fichier de zone (nombre décimal entier sur 8 chiffres)*
- **Tentatives** *: intervalle en secondes avant de recontacter le serveur principal si la* **demande de rafraîchissement a échoué**
- **Expiration** *: 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.*
- **Minimum** *: valeur TTL par défaut pour les RR qui n'en n'ont pas une explicitement. Cette valeur spécifie le temps maximal pendant lequel les autres serveurs de noms conserveront cette information dans leur cache*

```
@ IN SOA bach.btsig.fr albert.dupont.bach.btsig.fr. (
      200050124 ; numéro de série
      86400 ; rafraîchissement une fois par jour
      3600 ; tentatives : 1 heure
      3600000 ; expiration : 42 jours
      604800 ; minimum : 1 semaine
)
```

A *Cet enregistrement associe une adresse IP à un nom de machine :*

```
Bach  IN    A           200.100.40.11
```

NS *Ce type d'enregistrement spécifie un serveur primaire et tous ses serveurs secondaires*

```
IN    NS                bach.btsig.fr           #
```

CNAME *Cet enregistrement associe un alias au nom canonique d'un hôte indiqué par un enregistrement A*

```
www   IN    CNAME escher
```

PTR *Ce type d'enregistrement sert à la recherche des noms en fonction de l'adresse IP (recherche inverse ou reverse). Nous ne détaillerons pas ici. Cette construction est automatique dans la plupart des serveurs DNS (après avoir donné un point d'entrée bien sûr, voir partie pratique). La recherche inverse est essentiellement utilisée pour des raisons de sécurité*

```
10          IN        PTR     yoda.darkstar.fr.
```

MX *ce type d'enregistrement annonce un serveur SMTP : il indique que le courrier adressé sans nom d'hôte sera envoyé à la machine en question. Indique également la piorité*

```
Mail        IN          MX 10         200.100.40.20  # MX avec priorité 10
```

TXT *Contient une information libre de type texte.*

HINFO *Donne des informations sur l'hôte : 2 champs*

Structure des fichiers de la base de données DNS

Les fichiers de configuration sont au nombre de 5 au minimum :

- */etc/named.conf : fichier de configuration principal*
- *named.ca :fichier contenant les références des root-servers (inchangé)*
- *named.local : fichier de résolution inverse pour le loopback (inchangé)*
- *le fichier de zone directe : à paramétrer*
- *le fichier ou les fichiers de zone reverse : à paramétrer*

Exemple de fichier de zone directe:

@ IN SOA bach.btsig.fr. albert.dupont.bach.btsig.fr. (
32 ; numéro de série

```
        86400 ; rafraîchissement une fois par jour
        3600 ; tentatives : 1 heure
        3600000 ; expiration : 42 jours
        604800 ; minimum : 1 semaine
)
        IN      NS      bach.btsig.fr.          #
Escher  IN      A               200.100.40.12
Godel   IN      A               200.100.40.10
Mail    IN      MX      10      200.100.40.20  # MX avec priorité 10
www     IN      CNAME           escher
ftp     IN      CNAME           Godel
```

Bind et Named
Le serveur DNS sous Unix est assuré par un exécutable appelé **named** *qui fait partie du paquetage* **bind**. *On installera :*

bind9
bind9-doc
ainsi que **host, dig et nslookup**

La version courante de bind est la 9.x, on trouve également de nombreuses versions 8 et même quelques versions 4.xx qui posent quelques problèmes de sécurité.
On peut le télécharger à www.isc.org

Pour lancer lancer/stopper named, taper **ndc start/stop** *ou service* **ndc named start** *ou bien encore :***/etc/init.d/bind9 start|stop|status|restart|reload**

Pour tester, on utilisera **nslookup** *(en direct et en reverse !)* , **host** *(disponible sous Linux uniquement) ou bien encore* **dig**

```
Ex : host st250-07
host 192.168.2.3 # fait une résolution inverse
host –t ns mondomaine.fr #donne la liste des serveurs de nom du domaine
mondonaime.fr
host –l mondomaine.fr # donne la liste de tous les enregistrement du domaine
mondomaine.fr
host machine 192.168.12.1 #effectue une requête en interrogeant le serveur
192.168.12.1
nslookup machine.domaine.fr
dig serveur nom type
```

Prérequis
une machine dont la pile réseau est convenablement configurée

Exemple de configuration pour Bind 9

Fichier /etc/bind/named.conf

```
// generated by named-bootconf.pl
// secret must be the same as in /etc/rndc.conf
key "key" {
        algorithm       hmac-md5;
        secret
"c3Ryb25nIGVub3VnaCBmb3IgYSBtYW4gYnV0IG1hZGUgZm9yIGEgd29tYW4K";
};

controls {
    inet 127.0.0.1 allow { any; } keys { "key"; };
};

options {
       pid-file "/var/run/named/named.pid";
```

```
       directory "/var/named"; # repertoire des fichiers de zone
};

//
zone "." {
       type hint;
       file "named.ca";
};

zone "0.0.127.in-addr.arpa" {
       type master;
       file "named.local";
};

zone "darkstar.fr" { # zone directe darkstar.fr
       type master;
       file "db.darkstar.fr";
};

zone "1.168.192.in-addr.arpa" { # zone inverse
       type master;
       file "db.darkstar.fr.rev";
};
```

Fichier /etc/bind/db.darkstar.fr

```
@   IN   SOA    yoda.darkstar.fr. root.yoda.darkstar.fr. (
       2002102001 ; Serial
       28800      ; Refresh
       14400      ; Retry
       3600000    ; Expire
       86400 )    ; Minimum

             IN      NS    yoda.darkstar.fr.

localhost    IN      A     127.0.0.1
yoda         IN      A     192.168.1.10
luke         IN      A     192.168.1.11
mail         IN      CNAME  yoda
```

Fichier /etc/bind/db.darkstar.fr.rev

```
@     IN   SOA   yoda.darkstar.fr. root.yoda.darkstar.fr.  (
      2002102001 ; Serial
      28800      ; Refresh
      14400      ; Retry
      3600000    ; Expire
      86400 )    ; Minimum

       IN        NS      yoda.darkstar.fr.
10     IN        PTR     yoda.darkstar.fr.
11     IN        PTR     luke.darkstar.fr.
```

Si l'on souhaite utiliser le serveur DNS de son fournisseur d'accès, il est nécessaire de donner les instructions suivantes dans la partie options ;

```
forwarders {
                 10.0.0.1;
                 10.1.0.1;
         };
```

Cas d'un serveur esclave

Les fichiers de zone du maître doivent comporter des enregistrements NS pour le serveur de nom esclave. Le numéro de série doit être mis à jour rigoureusement pour que les serveurs esclaves puissent disposer de la nouvelle version du fichier de zone. Il doit d'autre part être croissant.

```
            IN      NS    maitre.mondomaine.fr.
            IN      NS    esclave.mondomaine.fr.
```

Fichier /etc/bind/named.conf de l'esclave

```
options {
        directory "/var/named";
};
zone "." {
        type hint;
        file "named.ca";
};
zone "0.0.127.in-addr.arpa" {
        type master;
        file "named.local";
};
zone "mondomaine.fr" {
        type slave;
        file "db.mondomaine.hosts";
        masters { 192.168.1.1 ; } ;
};
zone "1.168.192.in-addr.arpa" {
        type slave;
        file "db.192.168.1.hosts";
        masters { 192.168.1.1 ; } ;
};
```

Configuration du resolver

Le résolveur Unix

Fichier /etc/resolv.conf

```
domain          mondomaine.fr
search          autredomaine.fr # où rechercher quand la recherche a echoué
nameserver      192.168.1.1
nameserver      192.168.1.2
```

Fichier /etc/host.conf

```
order hosts,bind
```

La délégation de zones

Prenons l'exemple d'une zone zone2 d'adresse 192.168.10.0, rattachée à mondomaine.fr. Nous allons mettre en place une délégation de zone pour zone2. La résolution des noms de la zone zone2.mondomaine.fr est prise en charge par les serveurs de noms de la zone zone2, nous n'avons donc pas à nous en occuper. Par contre nous devons déclarer ces serveurs afin de maintenir la cohérence de la hiérarchie.

Configuration de la délégation : sur le serveur de noms de la zone mondomaine.fr il faut rajouter les enregistrements qui décrivent les serveurs de noms de la zone sd.mondomaine.fr dans le fichier de zone db.mondomaine.fr.

```
zone2           86400   NS      ns1.zone2.mondomaine.fr.
                86400   NS      ns2.zone2.mondomaine.fr.
```

Et les enregistrements qui déterminent les adresses de ces serveurs de noms.

```
ns1.zone2.mondomaine.fr.  IN   A  192.168.10.1
ns2.sd.mondomaine.fr.     IN   A  192.168.10.2
```

La délégation de la zone in-addr.arpa : Dans la pratique, cette délégation est différente car la zone inverse ne dépend pas de la zone supérieure, mais d'une autre entité (in-addr). Le processus est donc un peu différent.

Pourquoi ? parce que cette zone reverse est gérée par l'entité qui gère l'espace 192.168.0 à 192.168.255 et il est fort probable que ce n'est pas la zone domaine qui assure la résolution inverse pour tous les réseaux compris entre 192.168.0 et 192.168.255.

Ceci dit, cela n'empêche pas de réaliser cela sur une maquette. Il est possible de mettre en place cette résolution inverse. Nous allons donc considérer que la zone mondomaine.fr assure la résolution de noms inverse du réseau 192.168.254. Cela reviendrait à considérer que dans la réalité, la zone zone2 serait un sous domaine de mondomaine. La configuration ici est simple, les masques de sous-réseaux utilisés ici sont ceux par défaut des classes (255.255.255.0) pour la classe C. Le principe pour la zone inverse est identique à celui de la zone directe. Il suffit de rajouter dans le fichier db.10.168.192 les enregistrements suivants :

```
sd.mondomaine.fr.                   IN NS    ns1.zone2.mondomaine.fr.
                                    IN NS    ns2.zone2.mondomaine.fr.
1.10.168.192.in-addr.arpa   86400   IN  PTR  ns1. zone2.mondomaine.fr.
2.10.168.192.in-addr.arpa   86400   IN  PTR  ns2. zone2.mondomaine.fr.
```

Bind et la sécurité

La restriction des transferts de zone

Il est recommandé de poser des restrictions pour les transferts de zone pour éviter que des indélicats récupèrent le plan d'adressage complet du réseau. Il suffit d'utiliser la directive **allow-transfert***;*

```
zone "mazone" {
      allow-transfer { 192.168.1.4; localhost; };
};
```

La restriction des requêtes

Il peut être également intéressant de restreindre les requêtes : on utilise la directive **allow-query***.*

```
allow-query { 127/8; ! 192.168.1.10; 192.168.1/24; };
```

Bind 9

Bind 9 a apporté un certain nombre d'améliorations et de changements :

- **plus de sécurité** *: il est lancé comme un utilisateur normal (named) et nom plus comme root*
- *le* **Split DNS** *qui permet de créer des "Vues" DNS comme des vues SQL qui permettent de donner qu'une partie des informations de zone selon l'adresse des clients*
- *le* **DNSSEC** *ou DNS sécurisé qui permet des zones DNS sécurisées par cryptographie et authentifiées*
- **IPV6**
- *et enfin le* **DNS dynamique** *(DDNS) qui permet à un serveur DHCP (Version 3.0+) de mettre à jour les fichiers de zone automatiquement lors de l'attribution d'un bail DHCP.*

  Remarque importante *: l'utilisation de DDNS implique une certaine prudence lors de l'édition à chaud des fichiers de zone. En effet Bind, réécrit périodiquement ses fichiers de zone et maintient un journal pour les reprises en cas de problème.*

  Pour changer les informations dans un fichier de zone, il faut couper bind, supprimer les fichiers en **.jnl** *dans /var/named, puis éditer les fichiers de zone et enfin relancer bind.*
  Il est également possible d'utiliser la commande **nsupdate** *qui permet de modifier les enregistrements de ressource à chaud. cf* `man nsupdate`

D'autre part, la commande **ndc** *n'existe plus, elle est remplacée (presque !) par la commande* **rndc** *, qui ne permet plus de lancer bind : il faut utiliser* **service named start**

Pour plus d'information, voir le **Bind 9 Advanced Reference Manual** *dans /usr/share/doc/bind-9.2.0*
Le fichier /etc/named.conf de bind 9 a changé : il comporte des informations de sécurité indispensables à son fonctionnement : la directive **key** *servant à l'authentification des serveurs.*