DNS

Olivier Aubert

Liens

- `http://www.dns.net/dnsrd/` DNS Resource Directory
- `http://www.isc.org/products/BIND/` Internet Software Consortium - Berkeley Internet Name Domain
- `http://www.nic.fr/guides/dns-intro`
- RFC882, RFC883, RFC952, RFC974, RFC1035, RFC1131

Concept

- Comment ne plus manipuler des adresses IP numériques et désigner les objets du réseau en utilisant des noms symboliques (et représentatifs) ?
- Par exemple :
  www.univ-lyon1.fr $\Longleftrightarrow$ 134.214.100.218
- Avantages
  - Simplicité de remémoration (surtout dans l'optique IPv6...)
  - Il est possible de donner des noms à des services spécifiques, qui ne changent pas même si l'adresse IP du serveur change.

Solution ?

- Dans les premiers temps d'Internet, tous les noms de machines étaient spécifiés dans un simple fichier texte maintenu par le *Network Information Center* du SRI.
- Dans les années 1970, ARPAnet était un petite communauté informelle de quelques centaines de machines. Un simple fichier, HOSTS.TXT (RFC952), possédait toute l'information indispensable à leur propos : il contenait une corresponance nom-adresse pour chaque machine connectée à ARPAnet. Le fichier UNIX standard `/etc/hosts` était compilé à partir de HOSTS.TXT (principalement en supprimant les champs non utilisés par UNIX).
  (tiré du livre DNS & Bind (O'Reilly))

Problèmes

- Augmentation de la charge des serveurs centraux
- Collision de noms : un nom de machine pouvait entrer en conflit avec un autre déjà existant.
- Maintenance et synchronisation : le fichier pouvait être obsolète avant que toutes les machines ne l'aient reçu.

DNS

- Pour résoudre les problèmes du fichier HOST.TXT : Domain Name System
- Paul Mockapetris (University of Southern California) met au point un nouveau système, qui deviendra le DNS
- RFC 882 et 883 publiées en 1984, mises à jour par RFC1035
- Objectifs
  - système distribué (hiérachie, caches)
  - système redondant (serveurs primaires et secondaires), tolérance aux pannes
  - administration décentralisée et délégation (zones)
- 18 ans plus tard : ça marche encore.

Autres systèmes de nommage

- `/etc/hosts`
- NIS : Network Information Service, anciennement Yellow Pages (Sun)
- NIS+ : Hiérarchique et contrôles d'accès
- Annuaires X500 et LDAP
- WINS : Windows Internet Name Service (Microsoft)

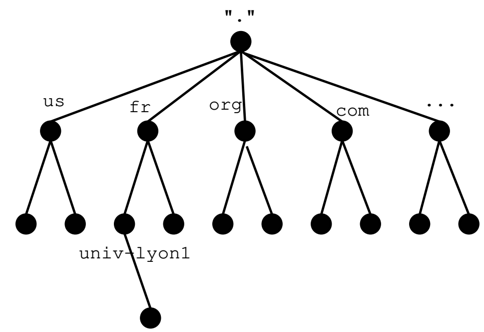
Comment ça fonctionne ?

- Une table distribuée de couples nom-valeur
- Le système peut convertir
  - Des noms en adresses (lookup direct)
    www.univ-lyon1.fr $\Rightarrow$ 134.214.100.218
  - Des adresses en noms (lookup de PTR)
    134.214.100.218 $\Rightarrow$ www.univ-lyon1.fr
- Le DNS se limite à trouver l'information et à la renvoyer au client
- Il peut également retourner d'autres types d'informations (RECORDS)

Nommage

- Noms séparés par des `.`, taille totale limitée à 255 caractères
- Taille de chaque nom <= 63 caractères (lettres, chiffres et `-`, premier caractère forcément une lettre)
- Pas de distinction majuscule/minuscule
- Concaténation d'au plus 127 noms

Comment est-il organisé ?

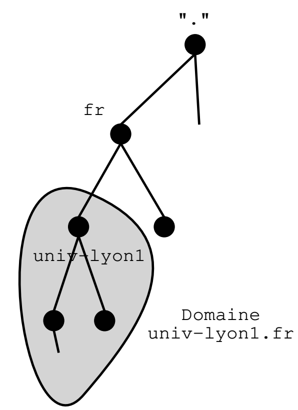


TLD

- Top Level Domains
- Domaines génériques : .aero, .biz, .com, .coop, .edu, .gov, .info, .int, .mil, .museum, .name, .net, .org
- Gestion de l'infrastructure : .arpa (reverse mapping, RFC1131)
- TLD nationaux (ccTLD : Country Code TLD) basés sur les codes ISO3166 (.fr, .de, .uk, ...)
- L'enregistrement et l'administration dans ces domaines est gérée par différents NIC (Network Information Center) :
  - .biz, .com, .info, .name, .net, .org : plusieurs entités sous la maîtrise de l'IANA (*Internet Assigned Numbers Authority*).
  - Asie et Pacifique : APNIC
  - Aux niveaux nationaux (France: FR-NIC, etc)
- La délégation est à la fois une procédure technique et administrative.

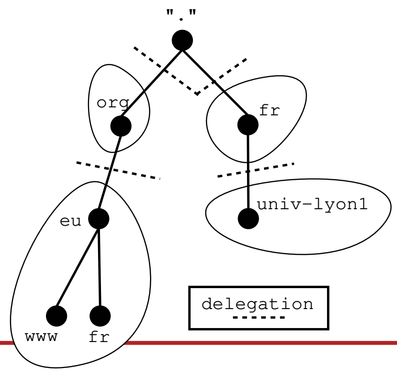
Domaines

- Un domaine est un sous-arbre de l'arbre des domaines.
- Un nom de domaine est la liste des noms rencontrés depuis un nœud vers la racine : `univ-lyon1 . fr .`
- Le nom de domaine `univ-lyon1.fr` est un FQDN (`Fully Qualified Domain Name`).



Zones

- Une zone est un sous-ensemble d'un domaine dont la gestion administrative est déléguée à une entité particulière.
- Techniquement parlant, c'est la partie de l'arbre gérée par le même serveur.
- L'administrateur d'un domaine peut choisir de déléguer certaines parties et de maintenir le reste dans une seule zone sous son contrôle.



Résolution par le client

- L'application essaye de résoudre un nom de machine (sous UNIX, via l'appel à `gethostbyname()`).
- La fonction transmet la requête au *resolver*. Quand une réponse est reçue, le résultat est renvoyé à l'application.
- Le *resolver* s'occupe d'effectuer la résolution de nom et d'adresse pour le système d'exploitation et les applications. On le configure via la fichier `/etc/resolv.conf`
- Sous UNIX, le fichier `/etc/nsswitch.conf` (Linux, Solaris) ou `/etc/host.conf` (BSD) spécifie les différentes moyens d'accès aux données de nommage (fichier `/etc/hosts`, DNS, NIS)

Le fichier resolv.conf

- Exemple

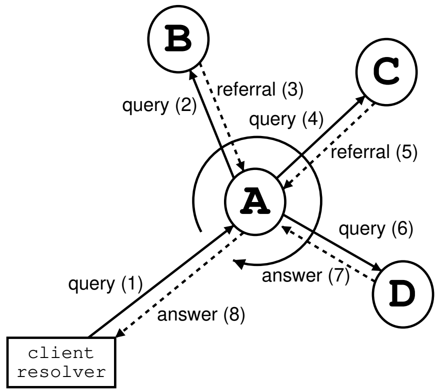
```
domain univ-lyon1.fr
search univ-lyon1.fr
nameserver 134.214.88.10
nameserver 134.214.88.23
```


Côté serveur : BIND

- Sur Internet, 99% des serveurs utilisent BIND (Berkeley Internet Name Domain).
- BIND répond aux requêtes de clients et d'autres serveurs
- Il renvoie des réponses tirées soit du fichier de zone local (s'il a autorité sur la zone auquelle le nom appartient) ou de son cache si la donnée a été précédemment demandée.
- Les serveurs sont redondants : pour chaque domaine, il y a un serveur primaire et des serveurs secondaires, qui se mettent à jour régulièrement auprès du serveur primaire.

Mécanisme des requêtes

- Lors d'une requête, il y a interaction entre plusieurs serveurs pour obtenir la réponse :
  - Le client envoie une requête à son serveur de noms.
  - Le serveur de noms (A) transmet la requête à un des serveurs de noms racine (B).
  - Le serveur de noms racine renvoie l'adresse du serveur de nom (C) qui a autorité sur le domaine demandé.
  - Le serveur (A) interroge le serveur (C) à propos du domaine demandé.
  - Le serveur (C) peut soit renvoyer la réponse, soit renvoyer l'adresse du serveur de noms (D) responsable du sous-domaine.
  - Ces deux dernières étapes sont répétées jusqu'à ce qu'une réponse soit renvoyée au serveur de noms (A), qui la transmet alors au client.
- À chaque niveau, un serveur peut renvoyer une donnée déjà présente dans son cache
  - si la même requête a déjà été effectuée
  - si le TTL (*Time To Live*) n'a pas expiré pour cette donnée



Configuration du serveur

- Le fichier `/etc/named.conf` contrôle la configuration du serveur de noms BIND. Il comprend
  - l'emplacement du fichier HINTS `named.root` qui contient les noms et les adresses des 13 serveurs de noms racine.
  - les noms des domaines pour lesquels le serveur est primaire ou secondaire, et l'emplacement du fichier de zone qui contient les informations.
  - dans le cas d'un serveur secondaire, on précise l'adresse IP du serveur primaire qui contient les informations.
  - des options de configuration

Exemple de fichier de conf

```
options {
        directory "/var/cache/bind";
};
zone "." {
        type hint;
        file "/etc/bind/db.root";
};
zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};
zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};
zone "info.univ-lyon1.fr" {
        type slave;
        file "info.univ-lyon1.fr.BAK";
        masters { 134.214.88.10; };
}
```


Fichier de zone

- Un fichier de zone contient des éléments appelés *Resource Records* (RR).
- Les noms de domaines complets sont terminés par un point. Sinon, ils sont considérés comme relatifs au domaine courant, qui sera ajouté automatiquement au nom dans les réponses.
- Premier RR = *Start Of Authority* (SOA) : nom du serveur primaire et adresse mail du responsable du domaine (avec `@` remplacé par un `.`).
- `NS` Record : indique les noms des serveurs de noms pour cette zone
- `A` Record : fait correspondre un nom à une adresse IP
- `CNAME` Record (Canonical Name) : crée un alias pour un nom de machine rdb
- `MX` Record (Mail eXchanger) : nom du serveur de mail (avec priorité) (RFC974)
- Autres *Records* : `PTR` (pour le mapping inverse), `HINFO` (infos sur la machine)

Valeurs du SOA

5 valeurs après le SOA :

- numéro de série : incrémenté par l'administrateur à chaque modification. Permet de détecter si un rechargement est nécessaire.
- *refresh time* : intervalle de temps (en secondes) entre deux vérifications des données auprès du serveur maître.
- Si la vérification échoue, le serveur réessaiera toutes les *retry* secondes. Si au bout de *expire* secondes le serveur maître n'a pas répondu, il est désactivé.
- *ttl* : combien de temps les autres serveurs sont autorisés à garder une copie de ces données

Exemple de fichier de zone

```
@          IN SOA  info.univ-lyon1.fr.  hostmaster.info.univ-lyon1
                          1       ;  serial number
                          21600   ;  refresh time : 6 hours
                          3600    ;  retry        : 1 hour
                          2419200 ;  expire       : 4 weeks
                          172800 ) ;  time to live : 48 hours
           IN NS     server.info.univ-lyon1.fr
           IN MX  10 server.info.univ-lyon1.fr

server     IN A      192.168.10.1
mail       IN CNAME  server
www        IN CNAME  server
; Gnralement gnr automatiquement...
machine    IN A      192.168.10.2
```


Transferts de zone

- Pour améliorer la redondance, chaque serveur primaire est secondé par des serveurs secondaires.
- Ceux-ci récupèrent depuis le serveur primaire une copie des données (transfert de zone).
- Pour déterminer si un transfert de zone est nécessaire, le serveur secondaire :
  - contacte le primaire après la durée *refresh* et demande le numéro de série de la zone
  - compare le numéro de série avec celui de sa copie locale
  - si le numéro de série a augmenté, effectue le transfert
- En cas de délégation d'un sous-domaine, tous les serveurs de nom (primaires et secondaires) doivent être spécifiés dans le fichier de zone.

Délégation d'un sous-domaine

- Ajouter dans le fichier de zone du domaine parent des enregistrements pour le nouveau sous-domaine avec les noms des serveurs :

```
...
info        IN  NS        192.168.10.1  ; primaire
            IN  NS        192.168.10.10 ; secondair
...
```

- Sur le serveur primaire, créer le fichier de zone (qui spécifie également les serveurs de noms pour le sous-domaine)
- Sur le serveur primaire, ajouter la nouvelle zone au fichier `/etc/named.conf` :

```
zone "info.univ-lyon1.fr" { type master;
     file "info.univ-lyon1.fr";  };
```

- Sur les serveurs secondaires, ajouter la déclaration (en type slave) dans le fichier `/etc/named.conf`.
- Redémarrer les serveurs

Interrogation du DNS

- `dig` est un outil d'interrogation du DNS (`nslookup` est en voie d'abandon)
- Syntaxe : `dig [@server] domain query-type`
- Types :
  - `a` Adresse IP
  - `any` Toutes les informations sur le domaine
  - `mx` Mail-eXchanger du domaine
  - `ns` Serveurs de noms
  - `soa` Start Of Authority
  - `hinfo` Host Information
  - `axfr` Transfert de zone
  - `txt` Zone de texte arbitraire

Exemple d'interrogation

```
>dig univ-lyon1.fr mx
[...]
;; QUESTION SECTION:
;univ-lyon1.fr.                 IN      MX
;; ANSWER SECTION:
univ-lyon1.fr.          432000  IN      MX      10 cismrelais.univ-lyon1.fr.
univ-lyon1.fr.          432000  IN      MX      5 pop.univ-lyon1.fr.
;; AUTHORITY SECTION:
univ-lyon1.fr.          432000  IN      NS      dns.univ-lyon1.fr.
[...]
;; ADDITIONAL SECTION:
cismrelais.univ-lyon1.fr. 432000 IN     A       134.214.101.250
pop.univ-lyon1.fr.      432000  IN      A       134.214.100.7
dns.univ-lyon1.fr.      432000  IN      A       134.214.100.6
;; Query time: 6 msec
;; SERVER: 134.214.88.10#53(134.214.88.10)
;; WHEN: Thu Jan 24 17:11:57 2002
;; MSG SIZE  rcvd: 260
```


Traduction inverse

- Problème : convertir une adresse IP `x.y.z.t` en nom
- Principe : inverser les nombres de l'adresse, et créer des sous-domaines de `in-addr.arpa`
- Exemple : `134.214.88.10`
- Demande de résolution de `10.88.214.134.in-addr.arpa`
- Résolution par le DNS de Lyon du domaine `88.214.134.in-addr.arpa`