

# Résolution de noms

(Z:\Polys\Internet de base\12.DNS.fm- 29 mars 2010 09:50)

## PLAN

- Introduction
- Noms des domaines de noms
- Principe de la résolution de noms
- La résolution de noms sous Internet/Unix
- Conclusion

## Bibliographie

- A.Fenyo, F.LeGuern, S.Tardieu, Se raccorder à Internet, Eyrolles, 1997
- L.Toutain, Internet et les réseaux locaux, Hermès, 1996.
- G.Hunt, TCP-IP network administration, O'Reilly, 1992.
- D.Comer, TCP/IP : Architectures, protocoles, et applic., InterEditions, 1998.

## 1. Introduction

On a besoin d'un service mondial d'annuaire pour Internet.

Service associant le nom d'une station à son adresse IP :

- Par exemple :

. pondichery.irisa.fr.  $\Rightarrow$  131.254.61.13

Mais aussi, il faut un service associant un serveur de messagerie au nom d'un utilisateur.

. bcousin@ifsic.univ-rennes1.fr.  $\Rightarrow$  mercure.univ-rennes-1.fr.

Il faut qu'il soit **stable, fiable et performant**.

## 2. Noms de domaine

### 2.1. Introduction

Les adresses IP sont adaptées à leurs tâches :

- identification dense (numérotation simple, sur 32 bits),
- aide à l'acheminement (netid + hostid)

Les êtres humains ont quelques difficultés à les utiliser :

- erreurs typographiques
- mémorisation difficile, etc.

La notation conventionnelle (décimale pointée) est insuffisante.

. on a besoin de noms symboliques:

. signifiants : <ma\_machine>

. facile à administrer : <ma-machine.mon-entreprise.mon-pays.>

Remarque : problème similaire entre nom de fichier et référence interne au noyau du système de gestion des fichiers ("inode" /= ~mon-nom/mon-repertoire/mon-fichier).

### 2.2. L'espace hiérarchique des noms

- . Présente une structure **arborescente** (similaire au système de fichiers, par ex.)
- . Chaque noeud de l'arbre est identifié par un "label" :
  - d'au plus 63 caractères alpha-numériques,
  - majuscules et minuscules étant identiques
- . Un noeud spécial, la racine, est identifié par une chaîne de caractères vide.
- . Le nom d'un noeud de l'arbre est identifié par la **suite de labels** rencontrés en partant de ce noeud et en allant vers la racine, chaque label étant séparé par un point.
  - un nom est unique, mais
  - plusieurs noms peuvent partager le même label
  - et un même nom peut avoir plusieurs labels identiques

Par exemple :

- ma-station.ifsic.univ-rennes1.fr.

Dénomination relative ou absolue :

- . Un nom absolu possède (est terminé par) un point [convention dépendant de l'implémentation]  
par ex : ma-station.ifsic.univ-rennes1.fr.
- . Un nom relatif doit être complété localement par un suffixe, celui du domaine local  
par ex : ma-station(.ifsic.univ-rennes1.fr.)  
nota : cette complémentation est définie par des règles purement locales

Remarque : ne pas confondre sous-domaine (de noms) et sous-réseau (IP).

### 2.2.1 Délégation / domaine de noms

Chaque organisation a la responsabilité de l'administration de son espace de noms (appelé "name domain").

Le **responsable d'un domaine** de noms peut décider de **déléguer** à un sous-responsable l'administration d'une partie du domaine des noms dont il a la charge.

Une arborescence de domaines et sous-domaines est ainsi créée.

Remarque : The NIC ("Network Information Center") à la responsabilité d'administrer le domaine de niveau le plus élevé.

Remarque : chaque domaine gère de manière autonome ses noms.

Remarque : l'arbre des domaines de noms n'a aucun besoin d'être équilibré.



## 2.4. Obtention d'un domaine en France

L'Afnic est le gestionnaire du domaine .fr.

==> <http://www.nic.fr> !

## 3. La résolution de noms

### 3.1. La "Host table"

#### 3.1.1 Présentation

Solution triviale au problème de traduction :

- chaque système contient un **fichier** des associations nom/adresse

Sous Unix : `/etc/hosts` (sous Windows : `Windows\host`)

- Exemple : le fichier `/etc/hosts` de la station "poseidon" :

```
#
# Internet host table
#
127.0.0.1 localhost
148.60.4.20 poseidon.ifsic.univ-rennes1.fr. poseidon
```

- La première entrée :
  - . 127.0.0.1 = "local loopback address"
  - . "localhost" = nom générique de la station elle-même
    - . les accès locaux ou distants peuvent être codés de manière identique
- La seconde entrée :
  - . l'adresse de la station : 148.60.4.20
  - . le nom officiel et les **alias** de la station

### 3.1.2 Conclusion

Cette technique n'est pas "scalable" :

- le réseau mondial contient des millions de stations
  - . le fichier local devrait contenir une entrée pour toutes les stations
- le réseau mondial est modifié en permanence (pannes, ajouts, suppressions, etc.)
  - . l'administration serait très difficile : mise-à-jour multiples, lenteur, risque d'incohérence, etc.

Il faut une technique offrant :

- un contrôle centralisé par domaine
- une dissémination automatique des associations : nom/adresse
  - => Système réparti de serveurs de noms : DNS ("Domain Name System")

Cependant, cette technique locale est utilisée :

- lors du démarrage du système (lors du "boot"),
- sur les petits sites isolés du reste du réseau,
- sur les vieilles stations ne disposant de DNS.

## 3.2. Résolution des noms par serveurs

### 3.2.1 Les serveurs

"Domain Name System" : un système réparti de serveurs de noms.

Trois rôles peuvent être tenus par un serveur DNS :

- serveur DNS primaire
- serveur DNS secondaire
- serveur DNS fonctionnant comme cache

A chaque domaine de noms est associé au moins un serveur **primaire**.

- c'est le serveur responsable du domaine ("authoritative server")
  - . les administrateurs ont un point central de gestion
- ce serveur contient les informations relatives au domaine
  - . vers le niveau supérieur : le serveur du domaine racine ("root server")
  - . vers le niveau inférieur : les serveurs de chacun de ses sous-domaines
  - . les règles de répartition des noms dans les sous-domaines
  - . **les noms qu'il gère** directement

Ce serveur primaire peut être flanqué de serveurs **secondaires** (aucun ou plusieurs)

- ils assurent la permanence du service : redondance
- ils échangent automatiquement une copie des infos détenues par le primaire :
  - . les informations obtenues à partir de ces serveurs sont à jour (“authoritative”).
  - . les serveurs secondaires doivent ne pas être localisés dans le domaine pour garantir la disponibilité de leur service.

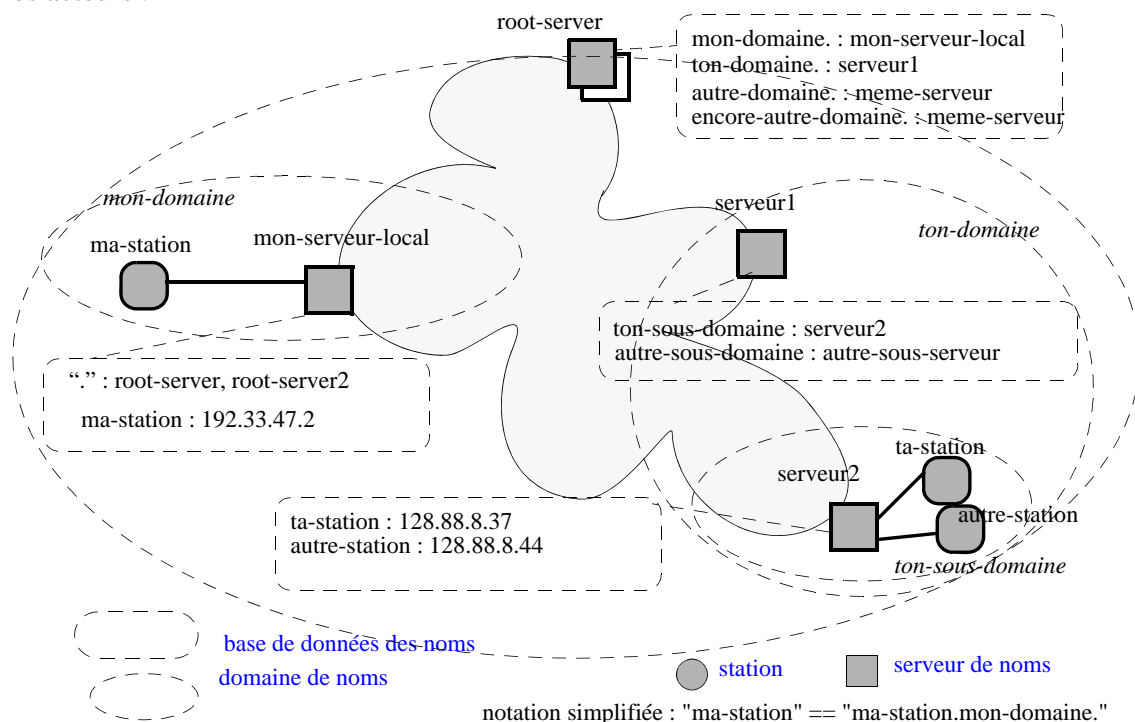
Des **caches** peuvent être mis en oeuvre afin d’accélérer la résolution des noms :

- la dissémination des infos est effectuée en fonction de l’utilisation réelle.
- les serveurs gérant ces caches stockent des infos qui peuvent ne pas concerner le domaine local.
- les informations obtenues peuvent être incorrectes ou incomplètes. Le serveur primaire associe à chaque information une durée de vie.

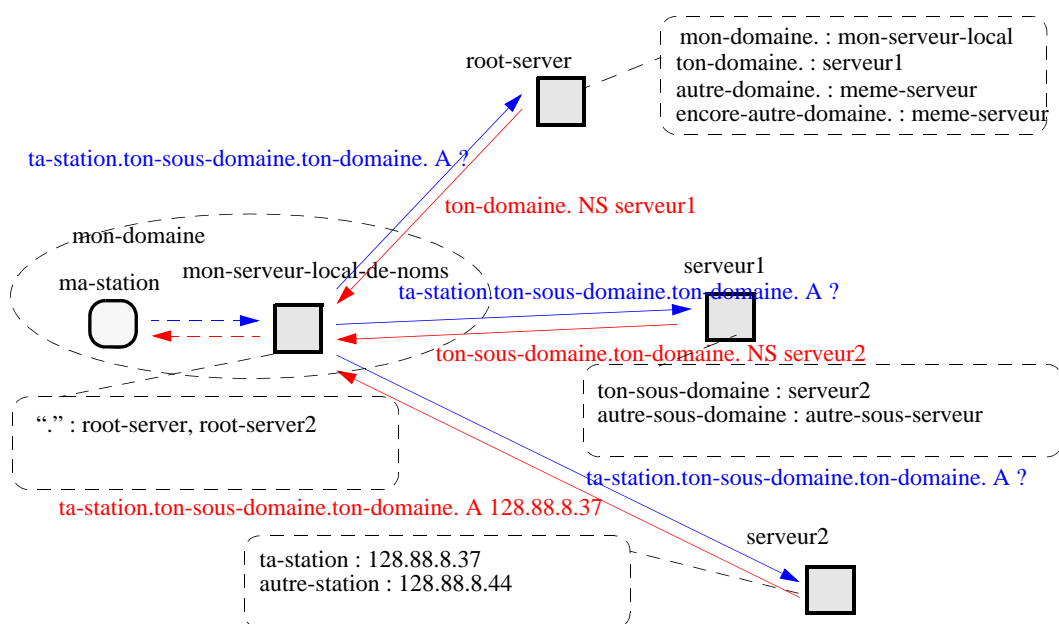
Remarque : un même machine peut supporter plusieurs serveurs de domaines de noms, et avec plusieurs rôles. Les serveurs primaires peuvent ne pas être localisés dans la zone dont ils ont la charge.

### 3.2.2 Exemple de résolution de noms

Les acteurs :



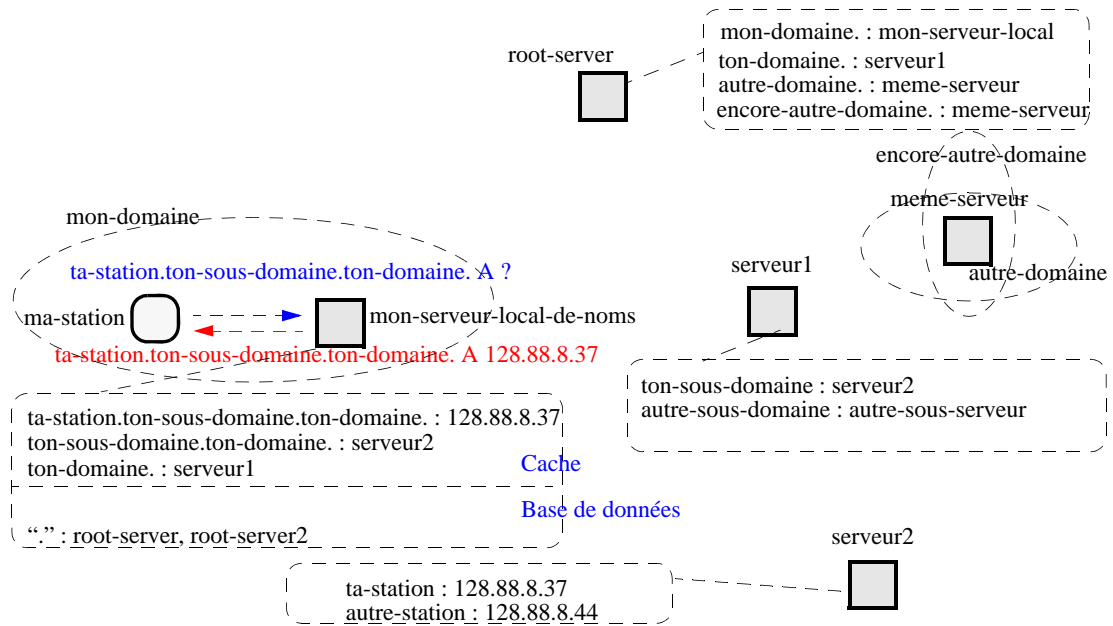
La résolution :



Cet enchainement est non-récuratif. Dans le cas récuratif, le serveur interrogé interroge d'autres serveurs distants, puis transmet la réponse au serveur local. Cela surcharge le serveur initial.



Le “cache”:



La réponse obtenue ne fait pas autorité : “non-authoritative”.

Remarque : “meme-serveur” supporte 2 domaines !

### 3.3. Caractéristiques

Différences avec d'autres systèmes de résolution de noms :

- les informations ne sont pas centralisées dans un seul fichier ou une station.
  - . répartition de la charge
- les informations obtenues peuvent être détenues par d'autres serveurs.
  - . souplesse, fiabilité
- les informations sont obtenues seulement quand on en a besoin.
  - . minimisation
- le système gère un cache des informations obtenues précédemment.
  - . optimisation

Le DNS peut s'adapter :

- à différents réseaux de communication (“Class”)
- à différents objets et informations (“Type”) :
  - . station, serveur de courrier, résolution inverse, informations diverses, etc.

## 4. Le résolution de noms sous Unix

### 4.1. Présentation

L'implémentation la plus courante sous Unix :

- BIND (“Berkeley Internet Name Domain”)
  - . un “resolver” de noms
  - . un serveur de résolution de noms
- Protocole normalisé :
  - . RFC 1034 : concepts (1987), RFC 1035: spécifications et implémentations

Le **serveur de noms** :

- répond aux demandes,
- procédé réparti de résolution des noms,
- s'exécute dans des stations distantes,
- un processus par serveur :
  - . nommé : named = “name daemon”, ou in.named,
  - . associé au port n° 53 par TCP ou UDP,
  - . présent sur tous les serveurs de noms (offrant le service DNS).

Le “**resolver**” de noms :

- code qui effectue la demande,
- sous la forme d'une bibliothèque de fonctions : gethostbyname(), gethostbyaddr()
- s'exécute sur la station locale,
- présent dans toutes les stations (accédant au service DNS).
- utilise
  - . soit des informations locales : /etc/hosts,
  - . soit des serveurs de noms distants : /etc/resolv.conf

```
domain ifsic.univ-rennes1.fr
search ifsic.univ-rennes1.fr irisa.fr univ-rennes1.fr
nameserver 148.60.4.1
nameserver 148.60.4.5
nameserver 131.254.254.2
```

```
nom du domaine par défaut
noms de domaines externes (utilisé pour compléter les noms)
adresse de serveur de noms
adresse de serveur de noms alternatifs
etc.
```

. La commande : nslookup

```
%nslookup jaihpur
Server: dns-2.irisa.fr
Address: 131.254.5.2
```

. La commande : dig

```
Name: jaihpur.irisa.fr
Address: 131.254.13.18
```

## 4.2. Exemples de fichiers de la base de données

Les fichiers contiennent des “DNS resource records”.

Le fichier du domaine de noms “mon-domaine” :

```
mon-domaine.      SOA mon-serveur.mon-domaine. admin-email.mon-domaine. (      ;; start of authority
    2002011200    ; numero de version (aaaammjjvv)
    10800         ; refresh (3h), fréquence à laquelle les serveurs secondaires tente de m-a-j
    3600          ; retry (1h), délais après lequel les serveurs secondaires retente une connexion
    604800        ; expire (1 sem.), délais après lequel les s. s. non m-a-j de répondront plus aux requêtes
    86400 )       ; TTL (1 jour), valeur par défaut du TTL
;
mon.domaine.      IN NS mon-serveur.mon-domaine.
                  IN NS mon-serveur-bis.mon-domaine.                ; les serveur de noms du domaine
;
mon-serveur.mon-domaine. IN A 192.33.47.2
ma-station.mon-domaine. IN A 192.33.47.102                          ; adresse des stations
;
autre-nom-de-ma-station.mon-domaine. IN CNAME ma-station.mon-domaine.
```

Le fichier inverse :

```
47.33.192.in-addr.arpa. SOA mon-serveur.mon-domaine. admin-email.mon-domaine. ( ;; start of authority
    1              ; serial
    10800          ; refresh (3h)
    3600           ; retry (1h)
    604800         ; expire (1 sem.)
    86400 )        ; min. TTL (1 jour)
;
47.33.192.in-addr.arpa. IN NS mon-serveur.mon-domaine.
                       IN NS mon-serveur-bis.mon-domaine.)        ; serveur de noms
;
2.47.33.192.in-addr.arpa.      IN PTR mon-serveur.mon-domaine.
102.47.33.192.in-addr.arpa.    IN PTR ma-station.mon-domaine.
```

## 5. Conclusion

Service de résolution de noms :

- . mais pas uniquement de noms: inverse, serveur de messagerie, infos, etc.
- . pour Unix et Internet mais pas uniquement.

Fiable, stable, facile à administrer et performant

On distingue les fonctions :

- . administration des noms, délégation
- . résolution des noms

Une arborescence de domaines, les noms sont une suite de labels.

Le procédé de résolution :

- . locale
- . par serveurs (primaire, secondaire, cache)

DNS : “Domain Name System”, BIND, “resolver”/”name server”

