DNS - Résolution de noms d'hôtes : partie théorique

Qu'est ce qu'un nom d'hôte?

Un nom d'hôte est un alias assigné à un ordinateur pour identifier un hôte TCP/IP. Ce nom peut être différent du nom NETBIOS. Il s'agit d'une chaîne pouvant comporter jusqu'à 256 caractères. Un hôte peut avoir plusieurs noms d'hôtes.

On peut utiliser avec **PING** le nom d'hôte plutôt que l'adresse TCP/IP.

Lorsqu'on veut utiliser le nom d'hôte pour désigner un hôte sur un réseau, il faut établir une correspondance (mappage) entre le nom et l'adresse TCP/IP. Cette correspondance peut être stockée dans un fichier (hosts) ou sur un serveur de noms (serveur DNS).

Nous avons vu dans le TP précédent le fichier /etc/hosts sur Linux, et la base WINS sur NT, qui sont des exemples d'association d'adresses IP à des noms d'hôtes.

Cette association rend plus facile l'utilisation de ressources réseaux en permettant de nommer les machines sur lesquelles elles sont stockées.

Ainsi, pour accéder à une ressource sur internet on écrit :

http: //montblanc.btsig.fr/cours/reseau/dns.html

Ceci est plus facile à utiliser qu'une adresse du type :

http: //200.100.40.11/cours/reseau/dns.html

L'utilitaire **HOSTNAME** (valable aussi pour Linux) permet d'afficher le nom d'hôte d'un système. Par défaut sur une machine Windows, le nom d'hôte est le nom NETBIOS donné à la machine.

Résolution de noms d'hôte

Comme pour un nom NETBIOS, on ne peut pas directement adresser une trame avec un nom d'hôte. La trame doit comporter l'adresse MAC du destinataire. Cette adresse est donnée par le protocole ARP qui cherche la correspondance entre l'adresse IP et l'adresse MAC d'un hôte. Donc il nous faut également une méthode qui convertisse un nom d'hôte en adresse IP.

En environnement purement TCP/IP (Unix/Linux ou Windows 2000 avec NetBIOS désactivé), les méthodes de résolution de noms sont les suivantes :

- cache (volatile)
- fichier hosts (à créer sur chaque poste)
- diffusion locale (ne passe pas les routeurs)
- serveur de noms DNS (la solution TCP/IP)

Microsoft TCP/IP peut utiliser les méthodes suivantes pour résoudre un nom d'hôte :

- cache (volatile)
- diffusion locale
- fichier hosts
- fichiers lmhosts
- serveur de noms WINS
- serveur de noms DNS

La résolution d'un nom d'hôte avec les fichiers hosts, lmhosts, le serveur de noms WINS, ou la diffusion locale ne diffèrent pas de ce que nous avons vu dans le TP précédent. Donc ici nous allons nous intéresser plus particulièrement au serveur de noms DNS.

Qu'est-ce que DNS?

Un serveur de nom de domaine (Domain Name System) est une base de données en ligne permettant de résoudre en adresses IP les noms de domaine complets (FQDN, Fully Qualified Domain Names, nous verrons par la suite de quoi il s'agit) et d'autres noms d'hôtes.

Le DNS est décrit par les RFC 1034 et 1035.

C'est un SGBD client serveur distribué et hiérarchisé. Il fonctionne au niveau de la couche application et utilise UDP ou TCP comme protocoles sous-jacents.

DNS effectue la correspondance entre les noms d'hôtes et leur 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 (approfondir avec requête itérative ou récursive)

Notion de domaine

Attention : un domaine DNS n'a rien à voir avec un domaine NT/2000

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 référencé généralement par un point (autrement dit label null).

En dessous se trouvent des domaines appelés domaines supérieurs dits de premier niveau (*Top Level Domains*). Ils sont divisés en trois zones principales :

Les **domaines organisationnels** (ou *Generic* TLD) sur trois caractères :

- com : Entreprises Commerciales
- edu : Education
- gov : Gouvernement
- org : Organisations Non commerciale
- net : Sites Réseaux (ex: nsf.net)
- mil : Militaire
- int : Organisations Internationales (ex : nato.int)

nouveaux noms de domaines organisationnels :

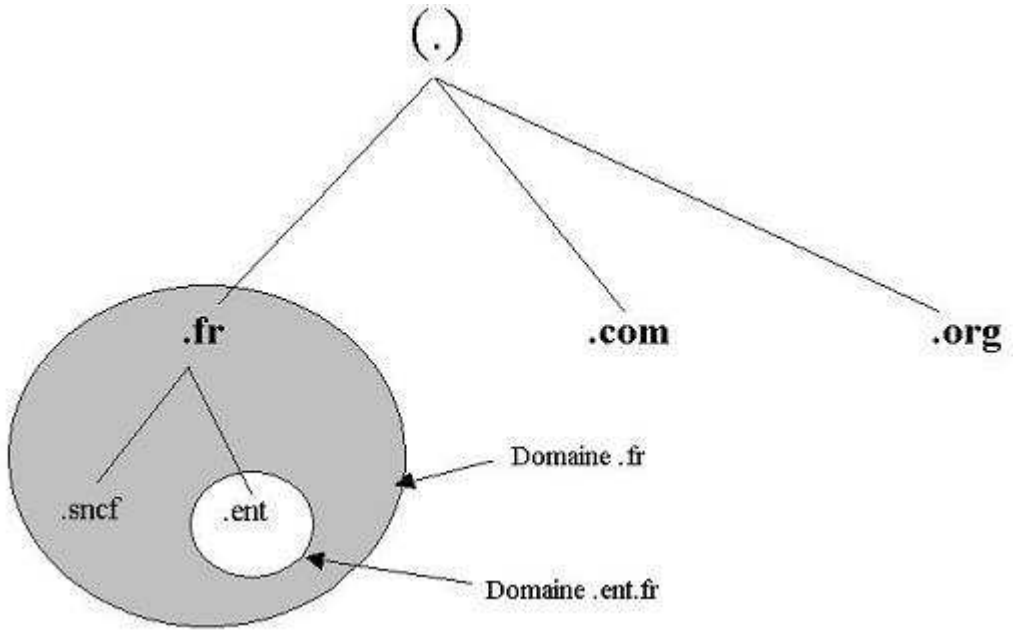
- biz : Business
- info : Information
- pro : sites professionnels
- museum : Musées
- aero : Voyages
- name : Noms
- coop : Coopération

Les **domaines géographiques** (codes de pays de deux caractères définis dans la norme ISO 3166) comme **fr** pour la France , **uk** pour le royaume uni…

Le **domaine in-addr.arpa** (domaine spécial utilisé comme annuaire inversé afin de retrouver un nom en connaissant l'adresse IP).

Sur le réseau Internet, les domaines racine et de niveau supérieur de la base de données DNS sont gérés par le **NIC** (*Network Information Center*). En fait, les premiers domaines organisationnels ont surtout été utilisés par les Etats-Unis. La répartition géographique est plutôt utilisée dans les autres 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 (Fully Qualified Domain Name).

Exemple : **Montblanc.btsig.lmd.fr** désignerait la machine **montblanc** du sous-domaine **btsig**, du sousdomaine **lmd**, du domaine **fr**

Un nom d'hôte doit être unique dans son domaine, on peut donc avoir une machine qui s'appellerait **montblanc** dans un autre domaine, par exemple : **montblanc.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).

Notion de zone d'autorité

Comment sont gérés les noms par un serveur de noms dans une telle structure ?

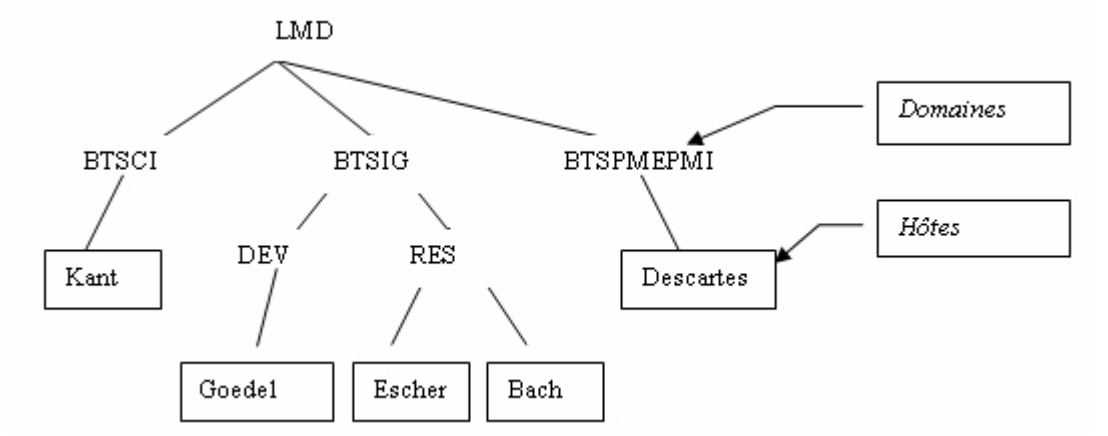
On pourrait imaginer simplement un serveur de nom par nœud de l'arborescence hiérarchique, gérant les noms des niveaux strictement inférieurs. Mais une telle organisation multiplierait les serveurs de noms et par voie de conséquence l'appel à ces serveurs.

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 direct 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 **LMD** qui aura autorité sur **BTSCI** et **BTSPMEPMI**, c'est à dire que les résolutions d'adresses de type **Kant.Btsci.lmd** et **Descartes.btspmepmi.lmd** 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.lmd**, **Escher.res.btsig.lmd**

Une zone d'autorité est donc une portion de l'arborescence (on dit une portion de l'espace de noms) mise 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 y a plusieurs types de serveurs de noms :

Serveur de noms principal (ou primaire): il gère directement une ou plusieurs zones stockées dans des fichiers locaux et récupère les informations dans ceux-ci.

Serveur de noms secondaire : il ne gère pas directement les informations sur les zones (fichier de zone en lecture seule), mais les obtient à partir du serveur de noms principal 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 secondaire demande donc le **transfert du fichier de zone** à un autre serveur. Cet autre serveur peut être soit directement le serveur principal soit un serveur secondaire intermédiaire. Dans tous les cas du point de vue du serveur secondaire, il s'agira d'un serveur **maître**.

Quand il démarre un serveur secondaire 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 principal, ni un serveur secondaire. Autrement dit, 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 placées en cache après ses résolutions. Son intérêt est d'éviter le transfert de zone.

Résolution de noms

Alors comment marche ce qui paraît être un imbroglio de domaines, zones, serveurs...?

D'abord il faut que ma machine connaisse son serveur de noms. Sous Windows (propriétés TCP/IP, onglet DNS), sous linux il faut mettre à jour le fichier **/etc/resolv.conf**

Exemple de fichier /etc/resolv.conf :

Domain btsig.lmd.fr

Nameserver 200.100.40.11

Ma machine a pour serveur de noms, le serveur ayant autorité sur **btsig.lmd.fr**, je veux trouver l'adresse IP de **Escher**. J'envoie une requête à mon serveur de noms et celui-ci me retourne l'adresse IP.

Facile me direz vous, puisque nous sommes placés dans le même espace de nommage. Essayons de contacter **descartes** (je pense qu'il existe :)). Mon serveur de noms ne sait pas résoudre directement ce nom.

Il va demander au serveur de noms de plus haut niveau s'il connaît un serveur de noms capable de résoudre l'adresse. En l'occurrence lui sait la résoudre, donc ça s'arrête là.

De plus en plus dur, je veux contacter **fr.yahoo.com**. Mon serveur de noms qui ne sait toujours pas résoudre cette adresse, va contacter encore une fois le serveur de plus haut niveau, celui-ci reconnaît que ce nom n'appartient pas à sa zone d'autorité, mais plutôt à une zone située en dessous du domaine com. Il va indiquer à mon serveur de noms de contacter un serveur de noms de cette zone et joint à sa réponse la liste de tous ces serveurs avec leurs adresses. Mon serveur de noms contactera l'un d'entre eux, qui lui dira que yahoo.com dispose de sa propre zone d'autorité et renverra les adresses des serveurs de noms de cette zone. Le serveur de noms présentera sa requête à ceux-ci jusqu'à que l'un d'entre eux reconnaisse le nom et retourne l'adresse IP correspondante.

Beaucoup d'allées et venues direz-vous. Imaginez ce que serait la maintenance d'une base de données centrale comportant tous les noms internet. Et puis, le serveur de noms local dispose bien sûr d'un cache dans lequel il va s'empresser d'inscrire la résolution précédente, ainsi la prochaine demande portant sur ce nom ne nécessitera pas tout cela. La durée de vie dans la cache dépendra d'un TTL (time to leave) positionné par l'administrateur responsable de la zone.

Notre exemple nous montre qu'un serveur de noms ne gère pas uniquement la correspondance nom d'hôte / adresse IP mais aussi qu'il doit disposer d'informations sur les autres serveurs de noms. Regardons d'un peu plus près sa structure.

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'appliquent les entrées. S'il est omis, le RR s'applique au domaine du précédent RR
- **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
- **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 les 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 : roger.sanchez@reseaucerta.org s'écrira roger.sanchez.ac-lyon.fr)

Numéro de série : numéro de version du fichier de zone, quand on modifie le fichier de zone, on incrémente ce numéro

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 à échouer

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 ont pas explicitement une. Cette valeur spécifie le temps maximal pendant lequel les autres serveurs de noms conserveront cette information dans leur cache

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

NS (Name Server) : ce type d'enregistrement spécifie un serveur primaire et tous ses serveurs secondaires

CNAME : cet enregistrement associe un alias au nom canonique d'un hôte indiqué par un enregistrement A

PTR : ce type d'enregistrement sert à la recherche des noms en fonction de l'adresse IP (**recherche inverse**). 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.

MX : ce type d'enregistrement annonce un serveur SMTP (nous en parlerons lors d'un TP sur le sujet)

Il existe d'autres types d'enregistrements que nous ne verrons pas ici.

Structure des fichiers de la base de données DNS

Dans NT/2000 le gestionnaire DNS permet une saisie assistée qui masque la structure des enregistrements sous-jacents. Dans Linux ce sont des fichiers textes qu'il faut paramétrer. Dans tous les cas, il est bon de s'intéresser à l'envers du décor dès lors qu'on administre un réseau.

Exemple de fichier :

@ IN SOA bach.btsig.lmd.fr Daniel.Dennet.bach.btsig.lmd.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.lmd.fr

Bach IN A 200.100.40.11

Escher IN A 200.100.40.12

Godel IN A 200.100.40.10

www IN CNAME escher

ftp IN CNAME Godel

Ce fichier sous Linux est le fichier **named.hosts**

Sous NT/2000 il serait sous **%systemroot%\system\dns**\btsig.lmd.fr.dns (la convention NT est de créer un fichier de zone appelé nom_de_zone.dns). Les clés dans la base de registres se trouvent sous **Hkey_local_machine\system\Currentcontrolset\services\DNSZones\...etc...**

Peut-être vous posez vous la même question que celle que je me suis posé pendant longtemps. Où se trouve dans ce fichier le nom de domaine?

Il ne s'y trouve pas. Il se trouve dans un autre fichier qu'on appelle fichier d'amorçage.

Attention le fichier d'amorçage n'est pas requis par une RFC. Ce fichier est une implémentation BIND (Berkeley Internet Name Daemon) du DNS. Mais vu qu'il est utilisé par Linux et que le serveur Microsoft DNS NT4 peut-être configuré avec, autant s'appuyer sur celui-ci (dans ce cas on utilise des fichiers textes et pas le gestionnaire DNS).

Le fichier d'amorçage s'appelle **named.boot** sous Linux. Il comporte 4 types d'entrées :

- **Directory :** spécifie le répertoire où se situent les fichiers de configuration DNS

Cache : spécifie le fichier utilisé pour permettre au DNS de contacter les serveurs de noms du domaine racine

Primary : spécifie le domaine pour lequel ce serveur de noms détient l'autorité et le fichier de base de données contenant les RR pour ce domaine (nous y voilà!)

Secondary : spécifie un domaine et la liste des serveurs maîtres à partir duquel il sera possible de télécharger le fichier de zone pour ce domaine

Exemple de fichier **named.boot** :

Directory c:\winnts\system32\dns

Cache . named.ca

Primary btsig.lmd.fr named.hosts

Le fichier **named.ca** pourrait avoir les enregistrements suivants :

99999999 IN NS kant.btsci.lmd.fr

kant.btsci.lmd.fr 99999999 IN A 200.100.50.11

Remarque : les fichiers de résolution inverse s'appellent sous Linux : **named.local**, **named.rev**

Vérification de la configuration du serveur de noms

L'utilitaire de vérification est **nslookup**.

Il fonctionne en mode ligne de commande ou en mode interactif.

Dans Linux ou dans NT/2000 pour avoir les différentes options du mode ligne de commande, consulter l'aide en ligne (nslookup help ou man nslookup).

Nous nous intéresserons dans la partie pratique au mode interactif.

Intégration DNS-WINS dans un environnement Windows

Quel intérêt me direz-vous d'avoir les 2 ?

DNS travaille en mode statique (les enregistrements sont saisis manuellement) et WINS travaille en mode dynamique. DNS est un standard Internet incontournable. WINS est très pratique dans un environnement Microsoft.

Comment intègre-t-on les 2 ?

Dans le fichier DNS un nouvel enregistrement est défini (un seul enregistrement WINS par serveur DNS), stocké au niveau du domaine racine de la zone.

Processus de résolution de noms :

1. Un client contacte le serveur DNS
2. Le serveur DNS parcourt sa base et ne trouve pas d'adresse pour l'hôte
3. Si le fichier contient un enregistrement WINS, DNS va convertir la partie hôte en nom NETBIOS et envoyer au serveur WINS une requête portant sur ce nom
4. Si le serveur WINS connaît le nom il renvoie l'adresse IP
5. Le serveur DNS renvoie alors l'adresse IP au client.