Ensimag 1ère année

TP n°2 – Configuration réseau, routage statique et DNS

> Il est recommandé de prendre des notes. Se référer à la version numérique de ce document pour les liens (sur Chamilo).
> Une question sur l'effet ou l'utilisation d'une commande ?
> **man nom_de_la_commande** !

Ce TP se compose de trois parties : nous commencerons par la configuration d'un réseau LAN. Nous verrons ensuite comment faire du routage entre différents réseaux. Enfin, nous observerons les mécanismes mis en jeu lors de la résolution de nom DNS.

Introduction

Pour ce TP, vous allez travailler sur des machines tournant sous `FreeBSD`. Sans rentrer dans les détails, il ne s'agit pas d'un système basé sur `Linux`, mais sur `BSD`.

`FreeBSD` a l'avantage de proposer une implémentation de référence pour l'ensemble des stacks `IPv4` et `IPv6`, et permet une configuration facile de celle-ci.

Démarrage de la séance : Au démarrage de l'ordinateur, sélectionnez *FreeBSD restore*, qui réinstallera une version réinitialisée de FreeBSD. L'ordinateur va redémarrer, sélectionnez maintenant *FreeBSD*. Suivant les salles dans lesquels vous êtes, on vous demandera un login : entrer `root`, éventuellement mot de passe : `root./`

1 LAN

Dans cette partie, nous allons voir comment mettre en place un réseau local à base d'adresses IP fixes. Nous observerons ensuite les mécanismes intervenant lors de la découverte de machines sur le réseau local.

Sur votre banc de travail, réalisez les branchements (avec les cables Ethernet) pour mettre e0n place un réseau constitué de quatre machines, reliées entre elles par un switch comme illustré sur la Figure 1.

1.1 Configuration du réseau

— *Choisissez une plage d'adresses IP ainsi que son masque de sous réseau associé. Justifiez.*

— *Choisissez une adresse IP pour chacune des machines.*

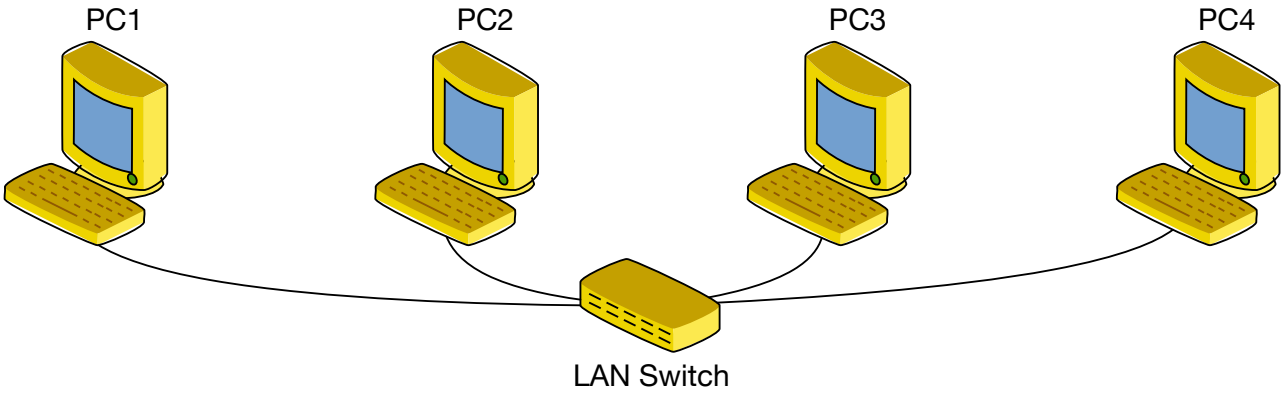

FIGURE 1 – Topologie simple d'un réseau local

1.2 Configuration des machines

Afin de configurer les machines, ouvrez un terminal. Nous allons maintenant configurer les interfaces réseau de vos machines.

Pour activer une interface avec une adresse IP particulière, on utilisera :
`ifconfig <Interface> <IP address> netmask <Netmask> up`, ou de manière plus concise :
`ifconfig <Interface> <IP address>/<Netmask> up` en notation *CIDR*.

Maintenant que toutes les machines ont une adresse IP, vérifiez qu'elles peuvent bien communiquer entre elles. Pour cela nous allons utiliser la commande `ping` que vous connaissez déjà. N'hésitez pas non plus à regarder l'état des interfaces réseau des machines à l'aide de la commande `ifconfig`.

— *Vérifiez que l'ensemble des machines soient capable de se « pinguer » entre elles.*

1.3 ARP — La découverte du voisinage

Le protocole de résolution d'adresse ARP (pour Address Resolution Protocol) permet de déterminer l'adresse physique (adresse MAC) à partir de l'adresse logique (adresse IP). Pour ce faire, il crée une table de correspondance entre les adresses logiques et physiques, qu'il stocke dans une mémoire cache. Lorsqu'un hôte doit communiquer avec un autre hôte connecté sur le même réseau local, il consulte sa table de correspondance (appelée aussi cache ARP). Si l'adresse IP s'y trouve, il en récupère l'adresse MAC associée. Dans le cas contraire, il envoie une requête d'interrogation sur le réseau et il met à jour sa table avec la nouvelle adresse.

Afin d'étudier et d'observer les mécanismes mis en jeu par le protocole ARP, nous allons utiliser `wireshark`, `arp` et `ping`.

1. Dans le terminal d'une des machines de votre réseau, effectuez un `ping` vers une autre de vos machines. Observez ensuite la table ARP à l'aide de la commande `arp -a -i <interface>`. *Que constatez vous ?*

Nous allons maintenant observer la mise à jour de la table de correspondance.

2. Arrêtez toute activité sollicitant le réseau sur la machine, et écoutez sur l'interface branchée (`eth0, em0, bge0`, ...) à l'aide de `wireshark`. Videz le cache ARP à l'aide de la commande `arp -ad`. Effectuez un `ping` vers une autre machine de votre réseau. *Qu'observez vous ? Y a t-il un comportement particulier ?*

3. Supprimez l'adresse correspondante à la machine que vous venez de « pinguer » du cache ARP, puis associez manuellement l'adresse MAC `42:42:42:42:42:42` à cette même machine à l'aide de la commande `arp -s <IP address> <MAC address>`. Refaites un `ping`. *Qu'observez vous ?*

1.4 Dynamic Host Configuration Protocol

La mise en place d'adresses IP "à la main" peut être un petit peu fastidieuse (voir impossible), notamment lorsque le parc de machines est important et en changement constant (eduroam par exemple), car le nombre de demandes d'adresses devient vite astronomique, et les adresses ne peuvent plus être allouées manuellement dans un temps raisonnable. De plus l'allocation statique demande une certaine rigueur dans la gestion des adresses déjà allouées afin d'éviter les doublons.

Pour pallier ces problèmes, on équipe souvent les réseaux de serveurs `DHCP`, qui se chargent de distribuer des adresses IP disponibles aux machines arrivant sur le réseau, vous évitant ainsi de le faire à la main.

2 Routage statique

Jusqu'ici, toutes les manipulations que vous avez effectuées au cours de ces TPs étaient réalisées sur un *réseau local*, c'est à dire l'interconnexion de plusieurs stations sur un même médium de communication (câble Ethernet et *hub* ou *switch*), comme le montre la FIGURE 1.

L'objectif de cette partie est d'étudier les moyens à mettre en œuvre pour interconnecter des réseaux afin que toutes leurs stations puissent dialoguer entre elles, comme dans l'Internet.

Internet est construit à partir de l'interconnexion de réseaux indépendants et fournit un service de communication universel. Ce service est indépendant de la structure et de la technologie utilisée localement dans chacun des réseaux. Le protocole IP, opérant à la couche réseau, est le support de ce service universel.

Physiquement, deux réseaux doivent être interconnectés par l'intermédiaire d'une machine qui possède un point d'attache sur l'un et l'autre des deux réseaux. Une telle connexion est nécessaire mais pas suffisante pour garantir la communication entre les machines situées sur ces deux réseaux. Il faut que la station reliant les deux réseaux coopère avec les autres machines et soit capable de passer les paquets d'un réseau à l'autre lorsque cela est nécessaire. Cette station qui joue un rôle particulier s'appelle un *routeur* (ou *passerelle*, en anglais *router* ou *gateway*). Il peut s'agir d'un simple ordinateur personnel, même si en général, des machines spécialisées sont utilisées.

2.1 Mise en place d'une topologie à deux sous-réseaux

Considérons la topologie réseau de la FIGURE 2.

1. Affectez un préfixe de sous-réseau à chaque sous-réseau. On utilisera des réseaux de type `192.168.x.0/24`, avec `x` le numéro du sous-réseau. Affectez une adresse IP aux quatre interfaces des 3 machines.
2. Mettez physiquement en place cette topologie (cables, branchements, ...)
3. Déterminez les interfaces qui ne peuvent pas communiquer entre elles à l'aide de `ping`. *Expliquez brièvement pourquoi.*

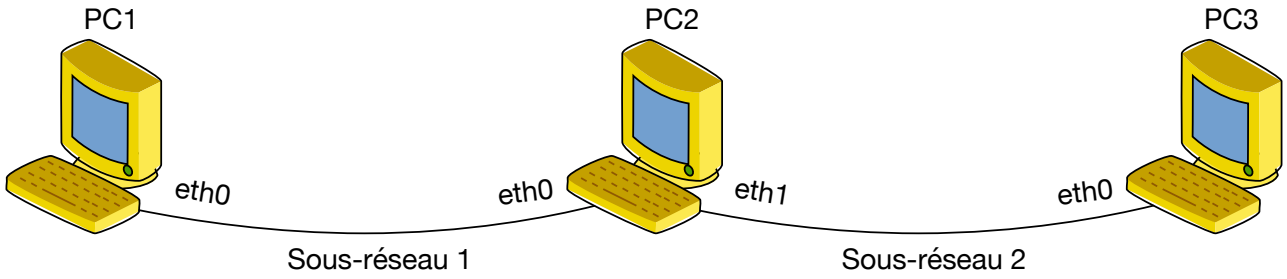

FIGURE 2 – Deux sous-réseaux interconnectés.

2.2 Mise en place du routage

Le mécanisme permettant l'acheminement des paquets à bon port à travers un ensemble de réseaux est appelé *routage*. Ce routage peut être plus ou moins perfectionné, par exemple statique, les chemins sont établis une fois pour toute, ou encore adaptatifs, les chemins peuvent varier dans le temps suivant différents critères : charge réseau, pannes, etc.

Il est à noter que, sur Internet, les tables de routages ne sont pas remplies à la main, comme nous le faisons dans ce TP (heureusement !). Des algorithmes de routages sont conçus pour découvrir les routes et remplir automatiquement la table de routage. Toutefois, le principe même du routage ne change pas.

Pour chaque paquet reçu, un routeur doit prendre une décision sur la route à suivre pour atteindre la destination indiquée dans l'en-tête IP du paquet. Le routeur ne connait pas d'information globale sur les routes de bout en bout, il dispose juste d'une information locale stockée dans une *table de routage*. Cette table est utilisée pour déterminer le prochain routeur de la route, souvent appelé *next hop* en anglais.

Voici quelques commandes de base permettant de manipuler la table de routage d'une station sous `FreeBSB`.

`netstat -rnf inet`
: afficher la table de routage pour IPv4

  L'option `-n` désactive la résolution de noms et permet d'afficher les adresses IP brutes (utile quand aucun serveur DNS n'est présent — cette option est présente pour la plupart des commandes réseau, comme `ping`). Avec l'option `-f inet`, affiche uniquement les informations IPv4 — `inet6` pour IPv6.

`route add|delete [-net|-host] <destination> <gateway>`
: ajoute ou supprime une route, par ex. `# route add 192.168.20.0/24 192.168.30.4`

`route flush`
: supprime toutes les routes

`sysctl net.inet.ip.forwarding=1`
: active le relayage de paquets, càd. transforme l'hôte en routeur. (`sysctl` permet de modifier certains paramètres du système. Vous pouvez en avoir la liste avec la commande `sysctl -a`).

Pour une information de référence, voir les pages `man`.

1. Dans la table de routage de `PC1`, ajoutez une entrée à destination du sous-réseau 2. Spécifiez bien le bon masque de réseau ou la longueur de préfixe quand vous ajoutez une entrée de réseau. Essayez de joindre `PC3` depuis `PC1` avec la commande `ping` (ne vous inquiétez pas tout de suite si ça ne marche pas : lisez la suite). Vérifiez avec wireshark que les paquets *ICMP echo request* atteignent bien `PC2` sur son interface `eth0`.

2. Les paquets *ICMP echo request* arrivant à `PC2.eth0` ne sont pas routés vers l'interface `PC2.eth1` : le relayage n'est pas activé sur `PC2`. Activez-le. *Les paquets arrivent-ils à l'interface `PC3.eth0` ?* Vous pourrez utiliser l'utilitaire `pong` dont le rôle est de vous prévenir quand la station reçoit une requête de ping (un message de la forme "echo request from ..." apparaît à l'écran...).
3. *Pourquoi `PC3` ne répond-il pas aux paquets ICMP echo request ?* Si vous n'arrivez pas à répondre, essayez d'atteindre `PC1` depuis `PC3` avec `ping`. Modifiez sa configuration pour faire en sorte que le `ping` fonctionne.

2.3 Portée des requêtes ARP

Sur toutes les machines, lancez `arp -ad` pour nettoyer leurs tables ARP. Sur `PC2`, capturez sur chacune des deux interfaces. Lancez ensuite `ping` de `PC1` vers `PC3`. Analysez les captures de wireshark, et en particulier les adresses Ethernet et IP des paquets.

1. *Les paquets ARP arrivant sur l'interface `PC2.eth0` sont-ils routés vers le sous-réseau 2 ? Pourquoi ?*
2. *Décrivez et expliquez l'enchaînement dans le temps des paquets échangés* (pensez à mettre le temps absolu sur wireshark : `clic-droit colonne Time -> Edit Column Details -> UTC time`).
3. *`PC1` connaît-il l'adresse Ethernet de `PC3` ?*

2.4 Routeur par défaut

La plupart du temps les machines hôtes n'ont qu'une seule entrée dans leur table de routage : une route par défaut. Cette route spécifie l'adresse du routeur qui recevra tout le trafic à destination d'adresses qui ne sont pas explicitement connues de la machine.

Videz la table de routage, puis :

1. Modifiez la configuration des machines pour utiliser `PC2` comme routeur par défaut. Vérifiez que `PC1` et `PC3` peuvent communiquer entre eux.

3 `DNS` — la résolution des noms de domaine

Pour cette section, vous aurez besoin d'internet. Rebranchez vos machines au réseau Internet.

Nous avons vu que l'adresse IP destination est nécessaire pour toute communication avec une machine sur Internet. Or il est plus simple de manipuler des noms que des adresses en décimal.

Une première solution est de stocker localement sur la machine les correspondances nom-adresse. C'est le cas, avec le fichier `/etc/hosts` où un certain nombre de correspondances existent, en général celles des machines appartenant au même réseau ou à un réseau local « proche ».

Dans le cas où la correspondance n'existe pas localement, c'est l'application DNS qui est utilisée.

Cette application permet de connaître une adresse IP d'une machine quelconque se trouvant sur Internet à partir de son nom symbolique.

Ainsi au moment de la commande `ping delos.imag.fr.`, il faut que l'application ping puisse déterminer l'adresse IP de `delos.imag.fr.` : elle fait appel à l'application DNS. Cela est vrai pour toutes les applications « réseaux » (ssh , navigateurs WEB , ftp. . .).

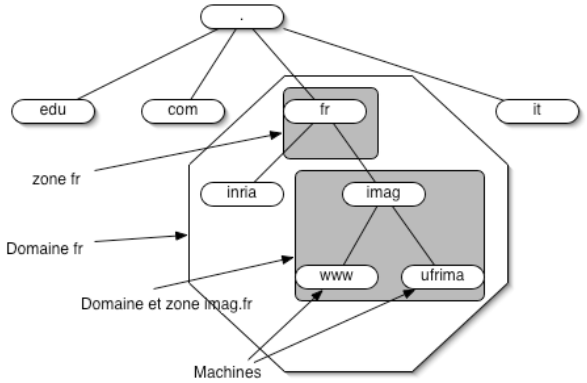

FIGURE 3 – Arborescence des noms DNS

3.1 Organisation des noms DNS

Pour faciliter la recherche de la correspondance (adresse, nom), les noms sont décomposés en plusieurs parties séparées par des points. Par exemple : `delos.imag.fr`.

Dans cet exemple, le nom appartient au domaine `fr.` (français), qui lui-même contient le domaine `imag.fr.` qui contient une machine `delos.imag.fr.` En haut de l'arborescence se trouve la racine (notée `.`). Cette hiérarchisation va permettre de faciliter la recherche de l'adresse IP associée à un nom, puisque l'information peut être distribuée selon cette hiérarchie.

On peut comparer cette notation à celle d'un fichier sous Unix avec comme séparateur le point « . » à la place de la barre oblique « / » mais noté à l'envers : la racine est à droite. On omet souvent cette racine, par exemple `delos.imag.fr.` est souvent noté `delos.imag.fr`, c'est-à-dire sans le point final.

On emploie le terme de « zone » pour désigner la base de données associée à un nœud de l'arborescence (zone `imag.fr.`, zone `fr.`, etc.).

À chaque zone (*i.e.* nœud interne de l'arbre DNS) est associé un ensemble de serveurs responsables de la base de données de la zone. On appelle ces serveurs les « serveurs de noms faisant autorité » pour la zone : ces serveurs permettent de répondre aux interrogations DNS à partir de leur base de données. Cette base de données contient principalement des adresses de machines utilisateurs (feuilles de l'arbre), ainsi que des « délégations » indiquant l'adresse de serveurs de noms faisant autorité pour des sous-zones. Par exemple, les serveurs de noms faisant autorité pour la zone `fr.` contiennent une délégation vers les serveurs de noms de la zone `imag.fr.`.

Ainsi, la hiérarchie DNS fonctionne comme une base de données distribuée : l'information est répartie sur un grand nom de serveurs de noms, qui peuvent être administrés de façon autonome.

3.2 Interrogation DNS

Pour « résoudre » un nom DNS et obtenir une adresse IP, il est donc nécessaire d'interroger la racine, puis de suivre les délégations successives jusqu'à arriver à la zone qui nous intéresse. On appelle ce processus la *résolution itérative* d'un nom, illustrée sur la Figure 4 (étapes 2 à 7).

Par exemple, pour trouver l'adresse IP correspondant à `delos.imag.fr`, il faut interroger les serveurs de noms faisant autorité pour la racine, `.` (étape 2). Ceux-ci renvoient les adresses des serveurs de noms faisant autorité pour `fr.` (étape 3). Une fois interrogés (étape 4), ceux-ci renvoient à leur tour les serveurs de noms faisant autorité pour `imag.fr.` (étape 5). Enfin, ces derniers renvoient la réponse attendue (étapes 6 et 7).

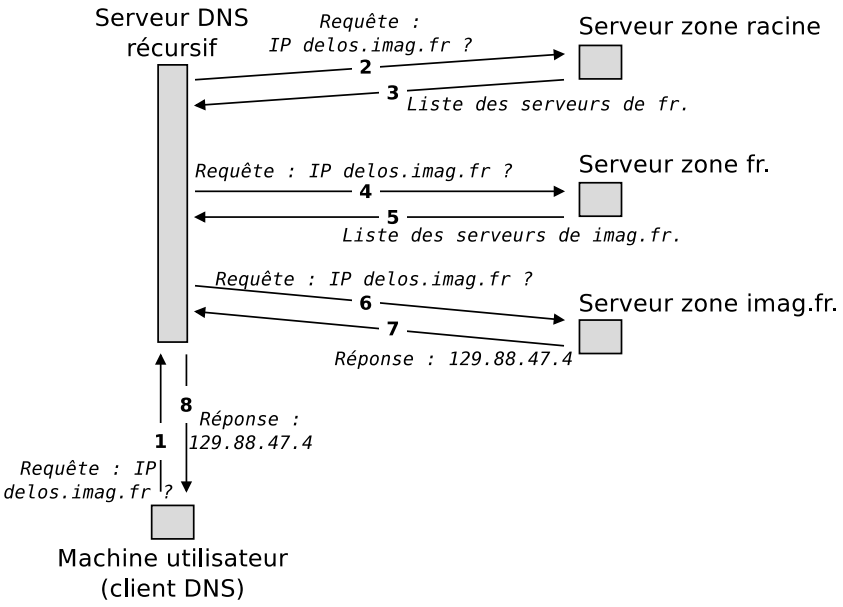

FIGURE 4 – Exemple de requête DNS

En pratique, ce travail est fait par un « serveur DNS récursif », qui garde également un cache des différentes informations obtenues. Les serveurs DNS récursifs sont typiquement fournis par les administrateurs systèmes du réseau local, ou bien par le fournisseur d'accès à Internet.

Pour résoudre un nom DNS, le client DNS pose alors la question à son résolveur DNS récursif (étape 1), et ce dernier renvoie soit la réponse obtenue par le processus de résolution itérative, soit une erreur (étape 8).

3.3 Requêtes `DNS` — les types d'enregistrement.

Un `DNS` est une base de données répartie contenant des enregistrements, appelés RR (Resource Records), concernant les noms de domaines. En effet, si le `DNS` sert principalement à retrouver une adresse IP à partir d'un nom, d'autres informations sont associées aux noms `DNS`.

Les enregistrements que nous étudierons dans ce TP sont :

- **A** : enregistrement associant une adresse IPv4 à un nom `DNS` ;
- **AAAA** : enregistrement associant une adresse IPv6 à un nom `DNS` ;
- **MX** : enregistrement associant un serveur de mail à un nom de zone `DNS` ;
- **NS** : enregistrement associant un serveur de noms à un nom de zone `DNS` ;
- **CNAME** : enregistrement associant le nom canonique de la machine à un nom `DNS`.
- **PTR** : reverse DNS.
- **SOA** (*Start Of Authority*) : enregistrement retournant l'adresse IP de la machine DNS qui a l'autorité sur le nom de domaine demandé.

`DNS` est un protocole *requête/réponse* : le client `DNS` construit une requête à destination d'un serveur `DNS` qui répond en fonction des informations contenues dans cette requête. Le contenu d'une requête `DNS` se résume à :

- un type d'enregistrement (A, AAAA, MX...) ;
- un nom `DNS` sur lequel porte cette requête (par exemple, `imag.fr`).

Pour faire des requêtes DNS en ligne de commande, il nous faut pouvoir construire la requête, l'envoyer à un serveur DNS, puis récupérer la réponse correspondante. Nous utiliserons, pour cela, les commandes suivantes :

— La commande `dig`. Sa syntaxe est : `dig <enregistrement> <nom DNS>`. Par exemple, la commande `dig MX grenoble-inp.fr` demande au serveur DNS par défaut de retourner au client un enregistrement DNS de type MX concernant le nom DNS `grenoble-inp.fr`. On obtiendra alors, si elle existe, la liste des serveurs de mails associée au domaine `grenoble-inp.fr`.

— La commande `nslookup`. Sa syntaxe est : `nslookup [-type=<type>] <nom DNS>`. Par exemple, la commande `nslookup -type=MX grenoble-inp.fr` est équivalente à la commande `dig` proposée ci-dessus.

Pour plus de détails sur ces commandes, *cf.* `man`.

1. *Trouvez l'adresse IPv4 de* ***`delos.imag.fr`***
2. *Trouvez l'adresse IPv6 de* ***`delos.imag.fr`***
3. *Quels sont les serveurs de courrier du domaine* ***`imag.fr`*** *? Appartiennent-ils au domaine* ***`imag.fr`*** *? Quels sont les serveurs de courrier du domaine* ***`grenoble-inp.org`*** *?* Remarquez que ces serveurs n'ont pas la même priorité.
4. *Quel(s) est(sont) le(s) serveur(s) de noms du domaine* ***`imag.fr`*** *?*
5. *Quel est le nom* ***`DNS`*** *associé à l'adresse 128.59.21.231 ? Trouvez ce nom à l'aide de chaque commande,* ***`dig`*** *et* ***`nslookup`****.*
6. Analysez un des paquets DNS circulant sur le réseau lors d'une requête. *Que pouvez-vous dire sur le protocole de transport du DNS. Quel est le numéro de port utilisé pour l'application DNS ?*

3.4 Unicité de l'association adresse – nom

1. *Est-ce qu'une adresse IP pourrait être associée à plusieurs noms ? Quel en serait l'intérêt ? Donnez un exemple (parmi les machines de l'environnement de l'Ensimag).*
2. Trouvez l'enregistrement de type CNAME de `pcserveur.ensimag.fr`. *À quoi sert cet enregistrement ? Pourquoi n'obtient-on pas l'adresse IP de la machine ? Cherchez l'adresse IP de cette machine.*
3. *Est-ce qu'un même nom peut correspondre à plusieurs adresses IP ? Quel en serait l'intérêt ?*
4. Faites une requête de type A pour le nom `amazon.fr`. *Que remarquez-vous ?*

3.5 Serveurs DNS

1. Lorsque vous utilisez `dig` ou `nslookup`, celui-ci va consulter un serveur de nom. *D'après la sortie de ces commandes, quelle est l'adresse IP du serveur consulté ? Retrouvez le nom de cette machine à partir de son adresse IP.*

   Chaque serveur DNS gère un cache permettant de stocker temporairement des relations adresse-nom ou des listes de serveurs pour une zone donnée. Cela permet de limiter le trafic dû à des interrogations successives.
2. Faites une interrogation `nslookup` de type A sur `www.google.com`, puis sur `ensipcserveur`. Que veut dire *non-authoritative answer* ? Pourquoi observe-t-on un réponse *authoritative* lors de la requête vers `ensipcserveur` ?

   On peut avoir une réponse *authoritative* en interrogeant directement le serveur qui a autorité sur le domaine en question, plutôt que d'interroger le DNS local (voir la deuxième figure DNS). Pour

changer de serveur à interroger, utilisez la commande
`nslookup <domaine> <serveur_DNS_à_interroger>`,
ou
`dig @<server_DNS_à_interroger> <domaine>`.

3. Essayez d'obtenir une réponse *authoritative* pour `www.google.com.` en envoyant directement la requête à un des serveurs d'autorité de la zone `google.com`.

   Indice : les serveurs d'autorité pour une zone peuvent être récupérés via le type `NS`. Mais attention : les DNS sortants sont bloqués par l'Ensimag, il est possible que vous ne puissiez pas faire ceci sur une machine de l'école ; essayez avec une machine personnelle par exemple.

4. Vérifiez l'adresse destination en capturant le paquet qui correspond à la requête DNS.

3.6 ICANN

1. Qu'est-ce que l'ICANN ?
2. Citez une zone TLD gérée par l'ICANN.
3. Citez une zone TLD gérée par un registre national.
4. Citez une zone TLD gérée par un registre privé.
5. Citez une zone TLD gérée par un registre non-officiel.