Ensimag 1ère année

TP n°2 — DNS, configuration réseau et routage statique

Il est recommandé de prendre des notes.
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 observer les mécanismes mis en jeu lors de la résolution de nom DNS. Puis nous effectuerons la configuration d'un réseau LAN. Enfin, nous verrons ensuite comment faire du routage entre différents réseaux.

1 `DNS` — la résolution des noms de domaine

On a pu voir dans les séances précédentes que lorsque vous consultez un site web, une l'adresse IP destination est nécessaire, et ce 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...).

1.1 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.

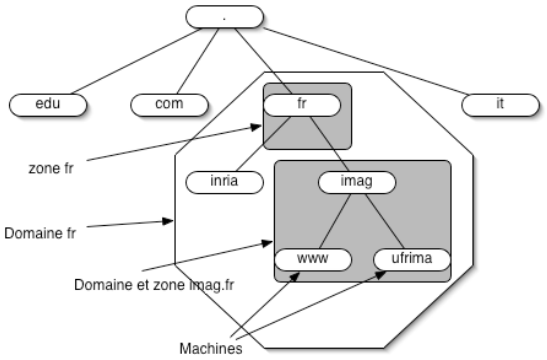

FIGURE 1 – Arborescence des noms DNS

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.

1.2 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 2 (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).

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.

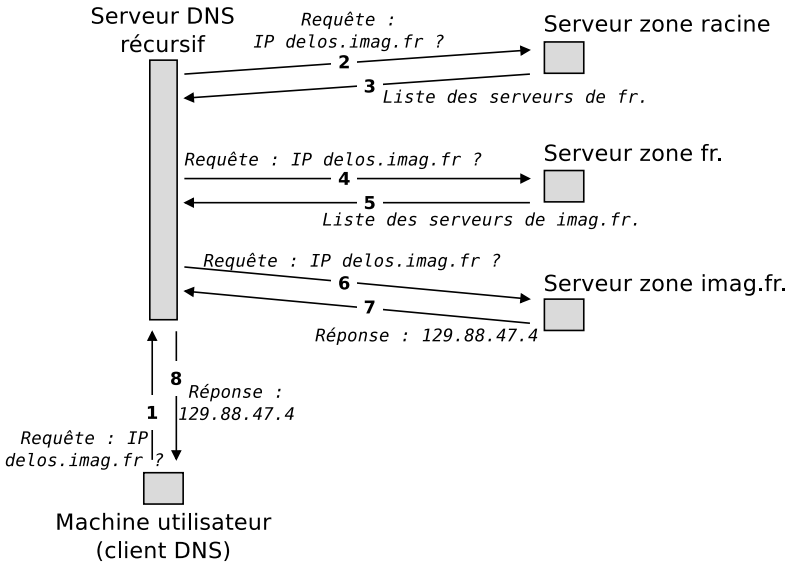

FIGURE 2 – Exemple de requête DNS

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).

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

Pour démarrer :

— Démarrez les machines sous « `Linux Ubuntu` » si ce n'est pas déjà fait.

— Identifiez-vous à l'aide de vos identifiants Ensimag.

`DNS` est un protocole *requête/réponse* : le client 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 (**ceux qu'on a vus en cours :** `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 <nom DNS> [<enregistrement>]`. Par exemple, la commande `dig grenoble-inp.fr MX` 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`. Si enregistrement n'est pas précisé, la valeur par défaut est A (adresse IPv4).

— 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. La valeur par défaut du type est A.

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

Utilisez `wireshark` pour observer les requêtes `DNS` effectuées pendant la consultation d'un ser-

veur de noms. Vous pourrez filtrer l'affichage dans wireshark pour ne montrer que les échanges DNS en utilisant le filtre « dns ». Ignorez les échanges MDNS qui sortent du cadre de ce TP.

Q 1 — *Trouvez l'adresse IPv4 de* `delos.imag.fr`.

Q 2 — *Trouvez l'adresse IPv6 de* `delos.imag.fr`.

Q 3 — *Quels sont les serveurs de courrier du domaine* `imag.fr`*? Appartiennent-ils au domaine* `imag.fr`*?*

Q 4 — *Quels sont les serveurs de courrier du domaine* `grenoble-inp.org`*? Remarquez que ces serveurs n'ont pas tous la même priorité.*

Q 5 — *Quel(s) est(sont) le(s) serveur(s) de noms du domaine* `imag.fr`*?*

Q 6 — *Quel est le nom DNS associé à l'adresse IP 128.59.21.231 (DNS inverse)? Trouvez ce nom à l'aide de chaque commande,* `nslookup` *et* `dig`.

Analysez un des paquets DNS circulant sur le réseau lors d'une requête.

Q 7 — *Que pouvez-vous dire sur le protocole de transport du DNS? Quel est le numéro de port utilisé par l'application DNS?*

1.4 Unicité de l'association adresse-nom

Q 8 — *Est-ce que plusieurs noms distincts peuvent pointer vers une même adresse IP? Si oui, quel en serait l'intérêt? Donnez un exemple (parmi les machines de l'environnement de l'Ensimag).*

Q 9 — *Trouvez l'enregistrement de type CNAME de* `intranet.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.*

Q 10 — *Est-ce qu'un même nom peut correspondre à plusieurs adresses IP? Quel en serait l'intérêt?*

Q 11 — *Faites une requête de type A pour le nom* `amazon.fr`*. Que remarquez-vous?*

2 LAN

Pour cette partie LAN, 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 une autre variante d'Unix, BSD.

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

Redémarrer votre ordinateur : Au démarrage de l'ordinateur, sélectionnez le menu de réinstallation et réinstallez *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./`

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

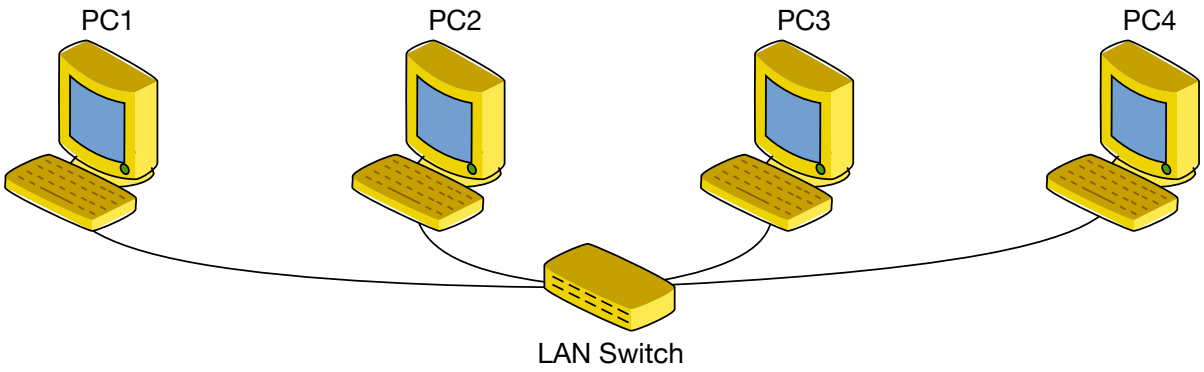

FIGURE 3 – Topologie simple d'un réseau local

2.1 Branchement des machines

Sur votre banc de travail, réalisez les branchements (avec les câbles Ethernet) pour mettre en place un réseau constitué de trois ou quatre machines, reliées entre elles par un switch comme illustré sur la Figure 3.

2.2 Configuration du réseau

Q 12 — *Choisissez une plage d'adresses IP ainsi que le masque de sous-réseau associé à cette plage. Justifiez.*

Q 13 — *Choisissez une adresse IP pour chacune des machines.*

2.3 Configuration des machines

Afin de configurer les machines, ouvrez un terminal. Lancez la commande `ifconfig` qui permet d'obtenir la liste des interfaces réseaux de la machine.

Q 14 — *De combien d'interfaces physiques dispose votre machine (en dehors de l'interface loopback `lo0`)? Quelle interface est branchée au réseau Ensimag (si vous êtes en salle de TP Ensimag)? Quelle interface venez-vous de brancher au switch?*

Pour configurer une adresse IP sur une interface réseau, on utilisera :

`ifconfig <interface> <adresse IP>/<taille> up`

Ici, `<taille>` est la taille de sous-réseau en notation CIDR (nombre de bits fixés à 1). Par exemple, si on choisit la plage `192.168.42.0`–`192.168.42.255`, la taille du sous-réseau est `24`.

Configurer l'adresse IP précédemment choisie sur chacune des machines (attention à ne pas se tromper d'interface!)

Q 15 — *Les machines peuvent-elles communiquer entre elles sur le réseau local? Vérifier la connectivité entre chaque paire de machines avec `ping`.*

2.4 ARP — La découverte du voisinage

Le protocole de résolution d'adresse ARP (pour *Address Resolution Protocol*) permet de déterminer l'adresse physique d'une machine, ou adresse MAC, à partir de son adresse IP.

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

Dans le terminal d'une des machines de votre réseau, effectuez un `ping` vers une autre des machines. Observez ensuite la table ARP à l'aide de la commande `arp -an -i <interface>`.

Q 16 — *Quelles adresses IP sont présentes dans cette table ? Repérez pour chaque entrée l'adresse MAC associée.*

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

Arrêtez toute activité sollicitant le réseau sur la machine, et lancez une capture `wireshark` sur l'interface réseau branchée au switch. Videz le cache ARP à l'aide de la commande `arp -ad`. Puis, effectuez un `ping` vers une autre machine de votre réseau.

Q 17 — *Qu'observez-vous dans la capture de paquets ? Expliquez l'échange de paquets ARP qui précède les messages envoyés et reçus par `ping` : qui envoie ces paquets ARP et à qui ?*

Arrêtez le `ping` et lancez-en un nouveau, toujours vers la même machine.

Q 18 — *Observez-vous de nouveaux échanges ARP ? Que pouvez-vous en conclure sur le rôle de la « table ARP » (appelée aussi « cache ARP ») ?*

Videz de nouveau la table ARP, puis associez manuellement l'adresse MAC `42:42:42:42:42:42` à l'IP de la machine que vous pinguiez, grâce à la commande `arp -s <IP address> <MAC address>`. Refaites un `ping`.

Q 19 — *Qu'observez-vous ? Analysez notamment la sortie de wireshark sur chacune des machines.*

Attention ! Pensez à vider le cache ARP avec `arp -ad` avant de passer à la suite de l'énoncé, sinon problèmes assurés...

2.5 Dynamic Host Configuration Protocol

La configuration d'adresses IP « à la main » peut être fastidieuse voire impossible, notamment lorsque le parc de machines est important et en changement constant (eduroam par exemple). 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.

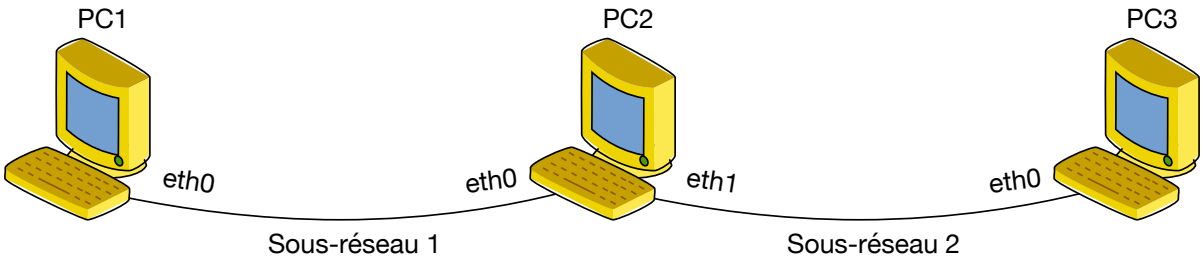

FIGURE 4 – Deux sous-réseaux interconnectés.

3 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 dans la partie précédente.

L'objectif de cette partie est d'interconnecter des réseaux afin que toutes leurs stations puissent dialoguer entre elles : vous allez construire une version miniature d'Internet!

Pour interconnecter deux réseaux, une machine doit être physique branchée à chacun de ces deux réseaux, comme le `PC2` sur la Figure 4 : il s'agit d'un *routeur*. Ce routeur peut être une machine spécialisée, ou bien simplement un ordinateur configuré pour relayer les messages d'un réseau à l'autre.

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

Modifiez les branchements pour obtenir la topologie réseau de la Figure 4. Vous aurez besoin d'une interface réseau supplémentaire pour `PC2` : soit vous disposez déjà de deux interfaces utilisables sur votre machine, soit il faut demander une interface réseau supplémentaire en USB à votre encadrant. Si vous êtes un groupe de 4, demandez de l'aide à votre encadrant.

Q 20 — *Choisissez une plage d'adresse IP pour chaque sous-réseau. On utilisera des réseaux de type `192.168.x.0/24`, avec `x` le numéro du sous-réseau. Attribuez ensuite une adresse IP à chaque interface.*

Q 21 — *Faites un schéma clair de votre réseau, qui comprendra : chaque machine et son nom, les liens réseaux, le nom de chaque interface, et l'adresse IP attribuée à chaque interface.* ***Les noms de vos interfaces réseau seront différents de ce qui indiqué sur la Figure 4, c'est normal!***

Configurez les adresses IP sur toutes les interfaces.

Q 22 — *Testez la connectivité à l'intérieur de chaque sous-réseau avec `ping`. À ce stade, pensez-vous pouvoir déjà communiquer d'un sous-réseau à l'autre?*

3.2 Mise en place du routage

Le routage consiste à construire des chemins à travers un ensemble de réseaux. Sur ces chemins, des routeurs relayent les paquets jusqu'à leur destination. Pour cela, les routeurs consultent leur *table de routage* qui indique, pour chaque IP destination, le prochain routeur sur le chemin, souvent appelé *next hop*.

Il existe plusieurs manières de construire ces tables de routage. Nous allons nous concentrer sur la méthode *statique*, c'est à dire que vous allez remplir ces tables à la main!

Affichez la table de routage de PC2, grâce à la commande `netstat -rnf inet`.

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 — on utilise `inet6` pour IPv6.

Q 23 — *PC2 a-t-il une route vers le sous-réseau 1? Vers le sous-réseau 2? Quel est le* next hop *dans chaque cas?*

Q 24 — *PC1 a-t-il une route permettant de joindre le sous-réseau 2?*

Ajouter une route sur PC1 à destination du sous-réseau 2. On utilisera la commande :

```
route add <destination>/<taille> <next-hop>
```

Par exemple : `route add 192.168.20.0/24 192.168.30.4`. Le *<next-hop>* est l'adresse IP du routeur qui doit relayer les paquets vers la destination.

Lancez wireshark sur PC1, PC2 et PC3. Sur PC2, lancez deux copies de wireshark : une par interface. Organisez vos fenêtres de façon à voir les deux copies de wireshark simultanément.

Q 25 — *Depuis PC1, pinguez PC3. Ça ne fonctionne pas, mais voyez-vous passer des paquets ICMP et si oui sur quelle interface? Sur quelle machine s'arrêtent ces paquets?*

Activer le *forwarding* sur PC2 grâce à la commande :

```
sysctl net.inet.ip.forwarding=1
```

Cela transforme PC2 en routeur et l'autorise à relayer des paquets.

Q 26 — *Depuis PC1, pinguez de nouveau PC3. Cette fois, jusqu'où vont les paquets? Pourquoi PC3 ne répond pas? Si vous n'arrivez pas à répondre, essayez de pinguer PC1 depuis PC3.*

Rajouter la route manquante sur PC3. Testez ensuite que le ping fonctionne désormais entre PC1 et PC3.

3.3 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.

Q 27 — *Les paquets ARP arrivant sur l'interface `PC2.bge0` sont-ils routés vers le sous-réseau 2 ? Pourquoi ?*

Q 28 — *Pour un même paquet ICMP Echo Request, quelle est l'adresse MAC de destination lorsque le paquet est émis par `PC1` ? Et à l'arrivée sur `PC3` ? Pourquoi ?*

Q 29 — *Décrivez et expliquez l'enchaînement dans le temps des paquets ARP et ICMP échangés (pensez à mettre le temps absolu sur wireshark : `clic-droit colonne Time -> Edit Column Details -> UTC time`).*

Q 30 — *`PC1` connaît-il l'adresse Ethernet de `PC3` ?*

3.4 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 sur chaque machine avec `route flush`.

Q 31 — *Modifiez la configuration des machines pour utiliser `PC2` comme routeur par défaut. Vérifiez que `PC1` et `PC3` peuvent communiquer entre eux.*

3.5 Démontage

Pensez à bien ranger tout le matériel que vous avez utilisé! (câbles Ethernet, câbles électriques, switches, éventuelles interfaces réseau supplémentaires...)

Si vous êtes dans les salles de TP IM²AG, lancez le script `/var/backups/BackToNormal` pour remettre la configuration réseau par défaut.

Nettoyez les tables et lavez-vous les mains.