SOMMAIRE

Liens :

http://www.frameip.com/dns/

http://www.frameip.com/rfc/rfc1035.php

http://fr.wikipedia.org/wiki/Domain_Name_System

http://fr.wikibooks.org/wiki/R%C3%A9seaux_TCP/IP/Les_serveurs_DNS

Bibliographie:

« Le réseau Internet » de Stéphane Lohier et Aurélie Quidelleur - Collection: Sciences Sup, Ed. Dunod

INTRODUCTION

Objectifs

Découvrir le service DNS.

Contexte

Un ordinateur équipé d'une carte de communication et d'un accès Internet.

On utilisera dans les manipulations les outils suivants : ping, whois, host (ou nslookup), traceroute (ou tracert), ...

nslookup/dig/host

`nslookup` est un programme informatique de recherche d'information dans le *Domain Name System* (DNS), qui associe nom de domaine et adresses IP. `nslookup` permet donc d'interroger les serveurs DNS pour obtenir les informations définies pour un domaine déterminé.

Il n'est plus maintenu pour UNIX et il est recommandé d'utiliser `dig` ou `host` à la place. Néanmoins cette commande est toujours d'actualité sous Windows. Il existe une version de dig pour Windows à cette adresse http://members.shaw.ca/nicholas.fong/dig/

[Source : http://fr.wikipedia.org/wiki/Nslookup]

whois

Whois est un service de recherche fourni par les registres Internet, par exemple les Registres Internet régionaux (RIR) ou bien les registres de noms de domaine permettant d'obtenir des informations sur une adresse IP ou un nom de domaine. Ces informations ont des usages très variés, que ce soit la coordination entre ingénieurs réseaux pour résoudre un problème technique, ou bien la recherche du titulaire d'un nom de domaine par une société qui souhaiterait l'obtenir.

[Source : http://fr.wikipedia.org/wiki/Whois]

Installation des outils sous Linux Manadriva (si nécessaire) :

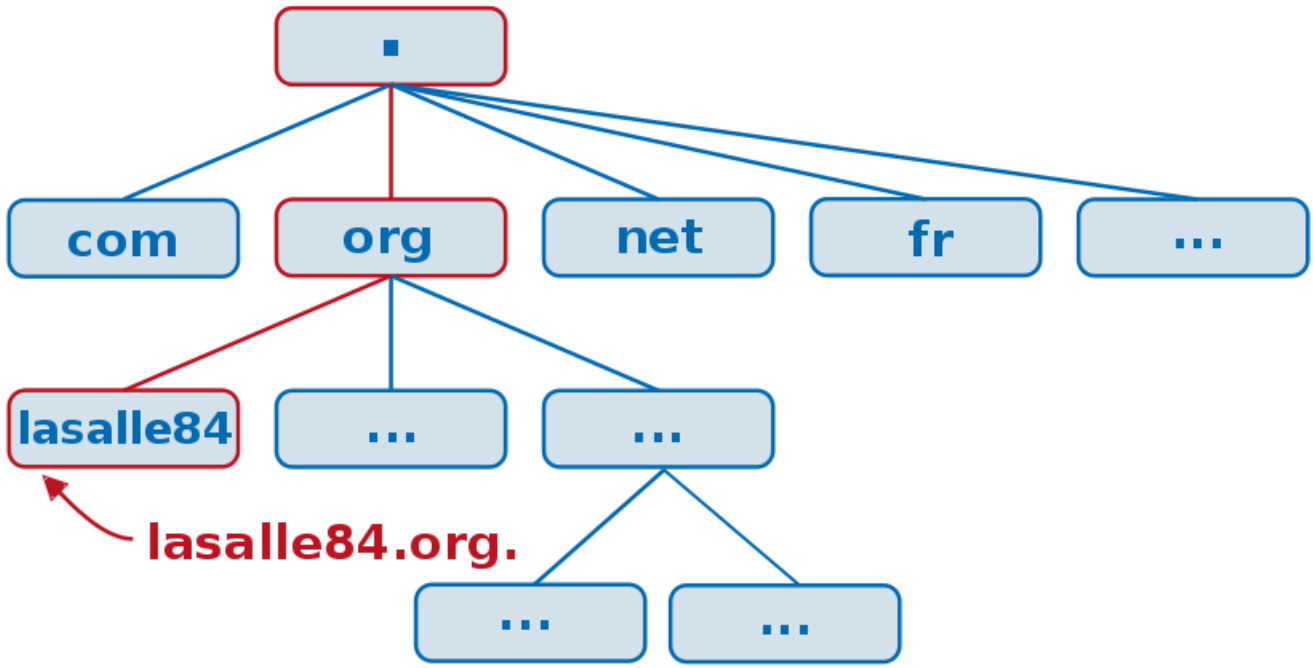
```
# urpmi whois
# urpmi traceroute
```

Introduction DNS

Hiérarchie du DNS (Domain Name System)

Le système des noms de domaines consiste en une hiérarchie dont le sommet est appelé la racine. On représente cette dernière par un point. Dans un domaine, on peut créer un ou plusieurs sous-domaines ainsi qu'une délégation pour ceux-ci, c'est-à-dire une indication que les informations relatives à ce sous-domaine sont enregistrées sur un autre serveur.

Les domaines se trouvant immédiatement sous la racine sont appelés domaine de premier niveau (TLD : Top Level Domain). Les noms de domaines ne correspondant pas à une extension de pays sont appelés des domaines génériques (gTLD), par exemple .org ou .com. S'ils correspondent à des codes de pays (fr, be, ch...), on les appelle ccTLD (country code TLD).



On représente un nom de domaine en indiquant les domaines successifs séparés par un point, les noms de domaines supérieurs se trouvant à droite.

FQDN (Fully qualified domain name)

On entend par FQDN (*Fully qualified domain name*) ou Nom de domaine pleinement qualifié un nom de domaine écrit de façon absolue, y compris tous les domaines jusqu'au domaine de premier niveau (TLD), il est ponctué par un point final. Dans une réseau TCP/IP, une adresse FQDN sera l'association entre le nom de la machine et le domaine auquel elle appartient.

Remarque : la norme prévoit qu'un élément d'un nom de domaine (appelé label) ne peut dépasser 63 caractères, un FQDN ne pouvant dépasser 255 caractères.

Notions de résolution

Résolution de noms directe

Dans un réseau IP, lorsqu'une machine A veut communiquer avec une machine B, la machine A connaît le nom FQDN de B.

Pour que A puisse communiquer avec B grâce au protocole IP, A va avoir besoin de connaître l'adresse IP de B.

A doit posséder un moyen d'effectuer la résolution de noms directe, c'est-à-dire un moyen de trouver l'adresse IP de B à partir de son nom qualifié.

Le **résolveur** est le programme chargé de cette opération.

Résolution de noms inverse

La machine B reçoit un datagramme IP en provenance de A. Ce datagramme contient l'adresse IP de A. B peut avoir besoin de connaître le nom FQDN de la machine A.

B doit donc être capable de trouver le nom FQDN de A à partir de son adresse IP. C'est ce qu'on appelle la résolution de noms inverse.

Le résolveur est également chargé de cette opération.

Remarque : La déclaration inverse est importante sur les adresses IP publiques Internet puisque l'absence d'une résolution inverse est considérée comme une erreur opérationnelle (RFC 1912) qui peut entraîner le refus d'accès à un service. Par exemple, un serveur de messagerie électronique se présentant en envoi avec une adresse IP n'ayant pas de résolution inverse a de grandes chances de se voir refuser, par l'hôte distant, la transmission du courrier (message de refus de type : IP lookup failed).

Client UNIX/Linux

Résolution de noms par fichier hosts

Le fichier **/etc/hosts** comprend l'adresse FQDN de chaque machine du réseau ainsi que son adresse IP.

Résolution de nom par serveur DNS

On installe un serveur de noms sur le réseau. Chaque machine du réseau doit connaître l'adresse IP de ce serveur DNS. Dès qu'une machine veut effectuer une résolution de noms directe ou inverse, elle va interroger le serveur de noms. L'administrateur doit configurer le serveur de noms pour que ce dernier connaisse l'adresse IP et le nom de toutes les machines du réseau.

Configuration de la résolution de noms

Le fichier **/etc/host.conf** contient des informations spécifiques pour la configuration de la bibliothèque de résolution de noms.

Le mot-clé **order** indique dans quel ordre la résolution des noms d'hôtes doit avoir lieu. Il doit être suivi par une ou plusieurs méthodes séparées par des virgules. Ces méthodes sont (généralement dans cet ordre) : **hosts, bind, nis**. Ce qui correspond à faire d'abord une résolution locale par le fichier hosts (hosts), puis par un accès à un serveur DNS (bind) et enfin par un accès à un serveur "yellow pages" (nis).

Pour en savoir plus, faire : `man host.conf`

Fichier de configuration de la résolution de noms

Le fichier **/etc/resolv.conf** contient des informations utilisées par la bibliothèque *resolver* qui est un ensemble de routines de la bibliothèque C fournissant un accès au système DNS Internet.

Les options de configuration de base sont :

- **nameserver** adresse IP du serveur de noms que la bibliothèque *resolver* interrogera
- **domain** Nom du domaine local

Pour en savoir plus, faire : `man resolv.conf`

Manipulations sous Linux

1) Que contient votre fichier /etc/hosts ?

```
$ cat /etc/hosts
```

2) Dans quel ordre se fera la résolution de noms sur votre machine ?

```
$ cat /etc/host.conf
```

3) Quelle est l'adresse IP du serveur de noms DNS que le résolveur interrogera ?

```
$ cat /etc/resolv.conf
```

4) Modifier le fichier de configuration de la résolution de noms pour qu'il interroge les serveurs DNS suivants :

Pour les serveurs DNS d'Orange :

DNS Primaire : 80.10.246.2

DNS Secondaire : 80.10.246.129

```
$ vim /etc/resolv.conf
nameserver 80.10.246.2
nameserver 80.10.246.129
```

ou avec OpenDNS :

DNS Primaire : 208.67.222.222

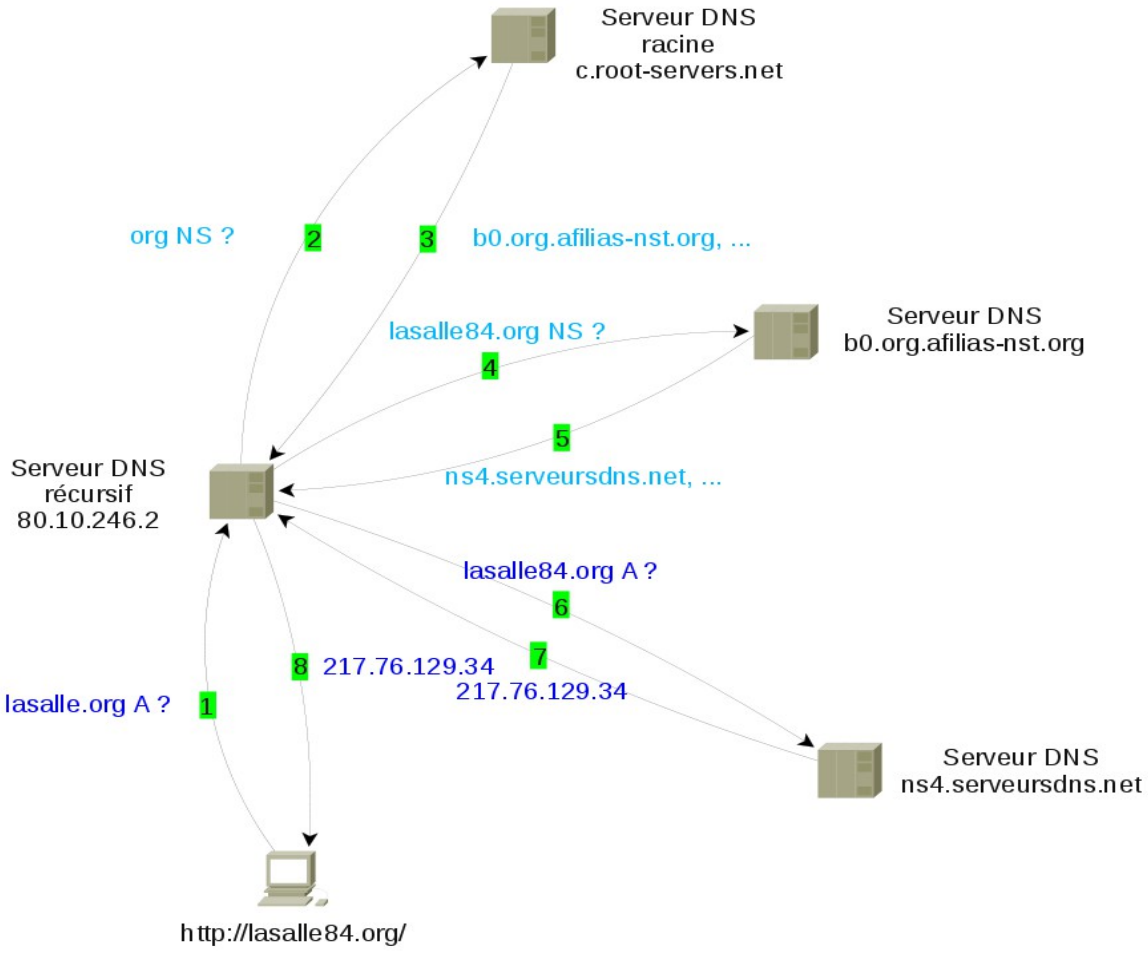
DNS Secondaire : 208.67.220.220

```
$ vim /etc/resolv.conf
nameserver 208.67.222.222
nameserver 208.67.220.220
```

Client DNS

Les hôtes n'ont qu'une connaissance limitée du système des noms de domaine. Quand ils doivent résoudre un nom, ils s'adressent à un ou plusieurs serveurs de noms dits **récursifs**, c'est-à-dire qui vont parcourir la hiérarchie DNS et faire suivre la requête à un ou plusieurs autres serveurs de noms pour fournir une réponse. Les adresses IP de ces serveurs récursifs sont souvent obtenues via DHCP ou encore configurés en dur sur la machine hôte (voir chapître précédent). Les fournisseurs d'accès à Internet mettent à disposition de leurs clients ces serveurs récursifs.

Quand un serveur DNS récursif doit trouver l'adresse IP de www.lasalle84.org, un **processus itératif** démarre pour consulter la hiérarchie DNS. Ce serveur demande aux serveurs DNS appelés **serveurs racine** quels serveurs peuvent lui répondre pour la zone org. Parmi ceux-ci, notre serveur va en choisir un pour savoir quels serveurs sont capables de lui répondre pour la zone lasalle84.org. C'est un de ces derniers qui pourra lui donner l'adresse IP de www.lasalle84.org. S'il se trouve qu'un serveur ne répond pas, un autre serveur de la liste sera consulté.



Pour optimiser les requêtes ultérieures, les serveurs DNS récursifs font aussi office de DNS cache : ils gardent en mémoire (cache) la réponse d'une résolution de nom afin de ne pas effectuer ce processus à nouveau ultérieurement. Cette information est conservée pendant une période nommée **Time to live** et associée à chaque nom de domaine.

Un nom de domaine peut utiliser plusieurs serveurs DNS. Généralement, les noms de domaines en utilisent au moins deux : un primaire et un secondaire. Il peut y avoir plusieurs serveurs secondaires.

L'ensemble des serveurs primaires et secondaires font **autorité** pour un domaine, c'est-à-dire que la réponse ne fait pas appel à un autre serveur ou à un cache. Les serveurs récursifs fournissent des réponses qui ne sont pas nécessairement à jour, à cause du cache mis en place. On parle alors de réponse ne faisant pas autorité (***non-authoritative answer***).

Cette architecture garantit au réseau Internet une certaine continuité dans la résolution des noms. Quand un serveur DNS tombe en panne, le bon fonctionnement de la résolution de nom n'est pas remis en cause dans la mesure où des serveurs secondaires sont disponibles.

La commande dig

La commande **`dig`** sous Linux est plus complète. Dig a l'avantage (ou l'inconvénient) de présenter les informations sous une forme directement utilisable dans un fichier de configuration de Zone DNS.

Suite à une commande dig, les flags renvoyés, lorsqu'ils sont présents, ont la signification suivante :

- qr (*query response*) indique qu'il s'agit d'une réponse à une requête.
- aa (*authoritative answer*) indique que la réponse vient directement d'un serveur faisant autorité.
- rd (*recursion desired*) indique qu'une requête récursive est demandée (par défaut).
- ra (*recursion available*) indique que la récursivité est disponible.

La commande **`host`** permet elle aussi de chercher des noms de machine à l'aide d'un serveur de domaine.

Manipulations

5) Déterminez l'adresse de lasalle84.org

```
$ host lasalle84.org
$ host -v lasalle84.org
```

Remarque : les enregistrements de type A (address) se trouvent dans la zone directe et permettent d'associer une adresse FQDN à une adresse IP. En général, chaque machine possède un enregistrement de type A dans sa zone directe.

6) Déterminez si la réponse du serveur DNS qui vous a répondu supporte la récursivité et si sa réponse fait autorité (« authoritative »).

```
$ dig lasalle84.org
```

Remarque : les enregistrements NS (name server) permettent de spécifier les serveurs de noms ayant autorité sur le domaine. Chaque fichier de zone comporte en général un tel enregistrement. Dans la zone org, les record NS suivants créent le sous-domaine lasalle84 et délèguent celui-ci vers les serveurs indiqués. L'ordre des serveurs est quelconque. Tous les serveurs indiqués doivent faire autorité pour le domaine.

7) Déterminez le serveur à utiliser pour obtenir une réponse « authoritative ». Ce serveur supporte-t-il la récursivité ?

```
$ host -v -t ns lasalle84.org
...
Remplacer <serveur_dns> :
$ dig @<serveur_dns> lasalle84.org
```

8) Quelle réponse vous donne un serveur DNS lorsqu'il ne supporte pas la récursivité et qu'il ne connaît pas la réponse à votre question ?

Vous pouvez par exemple utiliser un serveur de nom d'un domaine pour résoudre le nom d'un autre domaine de même niveau : dig @ns1.google.com www.yahoo.fr

```
$ dig @ns1.google.com www.yahoo.fr
```

9) Visualisez, avec l'option +trace la suite des serveurs contactés pour trouver l'adresse IP de www.lasalle84.org.

```
$ dig +trace www.lasalle84.org
```

10) Quels sont les domaines traversés et les serveurs de noms interrogés ? La requête est-elle récursive ?

11) Recherchez plusieurs fois l'adresse www.lasalle84.org. Que remarquez-vous ?

12) Qui est en charge de la zone org ?

```
$ dig ns @a.root-servers.net. org
```

Remarque : Il existe 13 serveurs racine, nommés de a à m.root-servers.net (http://www.root-servers.org/). Ces serveurs sont gérés par douze organisations différentes : deux sont européennes, une japonaise et les neuf autres sont américaines. Sept de ces serveurs sont en réalité distribués dans le monde grâce à la technique anycast et sept disposent d'une adresse IPv6. Grâce à anycast, plus de 200 serveurs répartis dans 50 pays du monde assurent ce service. Le serveur k reçoit par exemple de l'ordre de 20 000 requêtes par seconde (http://k.root-servers.org/index.html#stats).

Le DNS ne fournit pas de mécanisme pour découvrir la liste des serveurs racine, chacun des serveurs doit donc connaître cette liste au démarrage grâce à un encodage explicite. Cette liste est ensuite mise à jour en consultant l'un des serveurs indiqués. La mise à jour de cette liste est peu fréquente de façon à ce que les serveurs anciens continuent à fonctionner.

Remarque : En anycast, il y a une association "de une à plusieurs" entre les adresses réseau et les points d'arrivées finaux : chaque adresse de destination identifie un ensemble de récepteurs finaux, mais un seul d'entre eux est choisi pour recevoir l'information à un moment donné pour un émetteur donné.

13) Quelles sont les informations contenues dans les entrées de type SOA du DNS ?

Remarque : les enregistrements SOA (Start Of Authority) donnent les informations générales de la zone : serveur principal, courriel de contact, différentes durées dont celle d'expiration, numéro de série de la zone. Il désigne l'autorité (start of authority) ou le responsable de la zone dans la hiérarchie DNS. Cet enregistrement permet d'indiquer le serveur de nom maître (primaire), l'adresse e-mail d'un contact technique (avec @ remplacé par un point) et des paramètres d'expiration. Ces paramètres sont dans l'ordre :

- *Serial : indique un numéro de version pour la zone. Ce nombre doit être incrémenté à chaque modification du fichier zone ; on utilise par convention une date au format « yyyymmddhhmm » (« yyyy » pour l'année sur 4 chiffres, « mm » pour le mois sur 2 chiffres, « dd » pour le jour sur 2 chiffres, « hh » pour l'heure sur 2 chiffres et « mm » pour les minutes sur 2 chiffres) ;*
- *Refresh : l'écart en secondes entre les demandes successives de mise à jour réalisées depuis le serveur secondaire ou les serveurs esclaves ;*
- *Retry : le délai en secondes que doivent attendre le serveur secondaire ou les serveurs esclaves lorsque leur précédente requête a échoué ;*
- *Expire : le délai en secondes au terme duquel la zone est considérée comme invalide si le secondaire ou les esclaves ne peuvent joindre le serveur primaire ;*
- *Minimum ou negative TTL : utilisé pour spécifier, en secondes, la durée de vie pendant laquelle sont conservées en cache les réponses qui correspondent à des demandes d'enregistrements inexistants.*

```
$ host -v -a lasalle84.org 217.76.128.161
$ dig soa @217.76.128.161 lasalle84.org
$ dig soa @217.76.128.161 lasalle84.org +multiline
```

14) Comment déterminer la durée de validité d'une adresse ('A') ?

Remarque : Chaque enregistrement est associé à un Time to live (TTL) qui détermine combien de temps il peut être conservé dans un serveur cache. Ce temps est typiquement d'un jour (86400 s) mais peut être plus élevé pour des informations qui changent rarement, comme des records NS. Il est également possible d'indiquer que des informations ne doivent pas être mises en cache en spécifiant un TTL de zéro. Certaines applications, comme des navigateurs web disposent également d'un cache DNS, mais qui ne respecte pas nécessairement le TTL du DNS.

15) Quelle est la durée de vie de l'adresse www.dyndns.org et celle de station-stchamas.dyndns.org.

Consulter la page http://fr.wikipedia.org/wiki/DynDNS

16) Effectuez plusieurs requêtes successivement. Que remarquez-vous ?

17) Déterminez le nom de la machine d'adresse 192.0.32.7 et le serveur de noms qui gère cette résolution inverse.

```
$ dig -x 192.0.32.7
```

Remarque : À l'inverse d'une entrée de type A, une entrée PTR indique à quel nom d'hôte correspond une adresse IPv4. Si elle est spécifiée, elle doit contenir l'enregistrement inverse d'une entrée DNS A.

18) www.yahoo.fr est-il le nom canonique ou un alias ?

Remarque : un enregistrement CNAME (canonical name record) permet de faire d'un domaine un alias vers un autre. Cet alias hérite de tous les sous-domaines de l'original.

19) Déterminez le ou les serveur(s) d'échange de courrier pour le domaine lasalle84.org.

```
$ dig mx lasalle84.org
```

Remarque : Une entrée DNS MX indique les serveurs SMTP à contacter pour envoyer un courriel à un utilisateur d'un domaine donné. Les adresses mail étant codés en nom DNS, on remplace le premier "." (en partant de la gauche) par "@". Et éventuellement les "." avant "@" par "\." soit : thierry.vaira@orange.fr -> thierry\.vaira.orange.fr.

Bilan

Quels sont les différents types de requêtes DNS qui sont utilisés fréquemment ?

Réponses :

- *Requête sur un serveur de noms, NS*
- *Requête sur un nom d'hôte, A*
- *Requête sur une adresse IP, PTR*
- *Requête sur un agent de transfert de courrier électronique, MX*

CAPTURE

Lancez à l'aide d'un analyseur de protocoles (wireshark) une capture lors d'une demande de résolution de noms suite à une demande de site web vers http://apscplaisance.org.

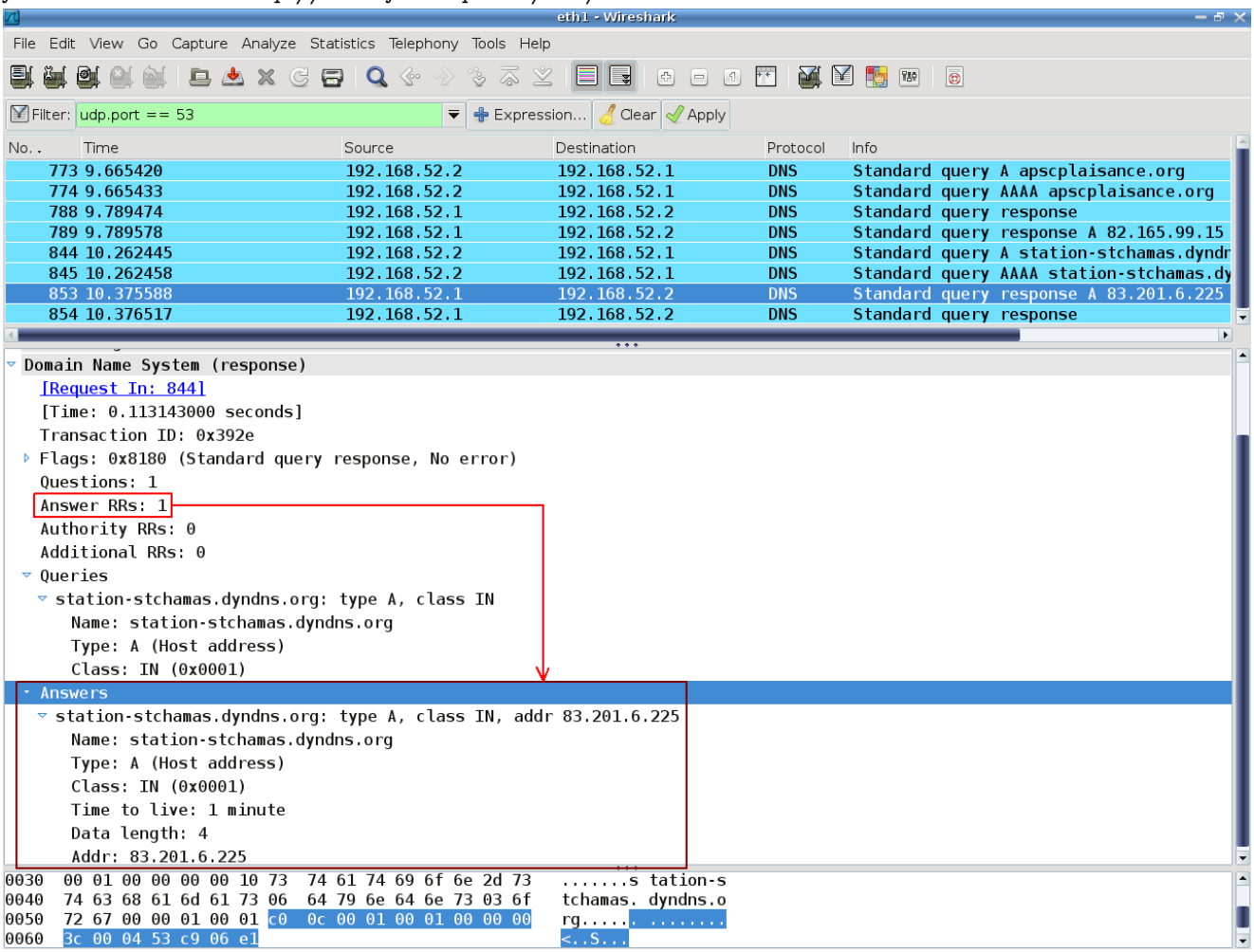
20) Quel est le protocole de niveau transport utilisé ? Justifiez.

21) Quel est le numéro de port de destination ? Justifiez.

Remarque : on pourra utiliser un filtre du type udp.port == 53

22) Quel est le type de requête DNS (le type de RR) ?

Remarque : Les fichiers de zone des serveurs de noms sont constitués "d'enregistrements de ressources" ("Ressource Records" ou RRs). Ces enregistrements sont répartis en classes. La seule classe d'enregistrement usuellement employée est la classe Internet (IN). Une description du protocole DNS est fournie sur le site : http://www.frameip.com/dns/



23) La requête est-elle de type récursive ? Quel est l'autre type de requête ?

24) Combien y a-t-il de RR dans la réponse ? À quoi correspondent-ils ?

25) Décrivez la résolution de noms obtenu suite à la consultation de ce site web.

Remarque : Technique du Round-Robin pour la distribution de la charge

Lorsqu'un service génère un trafic important, celui-ci peut faire appel à la technique du DNS Round-Robin (en français tourniquet), qui consiste à associer plusieurs adresses IP à un nom de domaine. Les différentes versions de Wikipedia, comme fr.wikipedia.org par exemple, sont associées à plusieurs adresses IP : 207.142.131.235, 207.142.131.236, 207.142.131.245, 207.142.131.246, 207.142.131.247 et 207.142.131.248. L'ordre dans lequel ces adresses sont renvoyées sera modifié d'une requête à la suivante. Une rotation circulaire entre ces différentes adresses permet ainsi de répartir la charge générée par ce trafic important entre les différentes machines ayant ces adresses IP. Il faut cependant nuancer cette répartition car elle n'a lieu qu'à la résolution du nom d'hôte et reste par la suite en cache sur les différents resolvers (client DNS).

UN CAS CONCRET

1 . On « ping » un site hébergé par free.fr, donc en Europe, pour obtenir son adresse IP :

```
$ ping -c 1 tvaira.free.fr
PING perso132-g5.free.fr (212.27.63.132) 56(84) bytes of data.
64 bytes from perso132-g5.free.fr (212.27.63.132): icmp_seq=1 ttl=54
time=52.8 ms
```

2 . On interroge RIPE (pour l'Europe) pour en savoir plus sur cette adresse :

```
$ whois -h whois.ripe.net 212.27.63.132 | grep -E -e "inetnum|route"
inetnum:      212.27.60.0 - 212.27.63.255
route:        212.27.32.0/19
```

On obtient l'adresse CIDR donné à l'ISP Free SA par RIPE soit 212.27.32.0/19. On peut vérifier :

```
$ whois -h whois.ripe.net 212.27.32.0/19 | grep -E -e "inetnum|route|
descr"
inetnum:        212.27.32.0 - 212.27.63.255
descr:          Free SAS
route:        212.27.32.0/19
descr:        ProXad network / Free SA
descr:        Paris, France
```

3 . On "trace" la route pour atteindre l'adresse IP du site :

```
# traceroute -nI 212.27.63.132
traceroute to 212.27.63.132 (212.27.63.132), 30 hops max, 60 byte
packets
 1  192.168.52.1  0.906 ms  1.172 ms  1.412 ms
 2  90.28.192.1  40.591 ms  41.236 ms  42.156 ms
 3  10.125.49.78  42.986 ms  43.883 ms  44.876 ms
 4  193.253.86.238  45.790 ms  46.640 ms  52.786 ms
 5  193.252.101.118  61.209 ms  61.255 ms  61.328 ms
 6  193.251.126.114  65.186 ms  64.453 ms  64.751 ms
 7  81.253.181.230  131.360 ms  92.237 ms  91.848 ms
 8  193.252.103.110  52.082 ms  52.121 ms
 9  193.252.160.154  52.334 ms  52.162 ms *
10  212.27.50.157  53.751 ms  52.575 ms  53.272 ms
11  212.27.58.125  54.027 ms  52.047 ms  53.139 ms
12  212.27.63.132  53.233 ms  52.203 ms  51.952 ms
```

Les routeurs 10 et 11 sont bien des routeurs de chez Free dans des sous réseaux différents (les 2 adresses sont dans la plage 212.27.32.0 - 212.27.63.255 de chez Free).

4 . On vérifie sur le site www.iana.org à qui appartient l'adresse 212.x.x.x. :

```
$ wget -q -U "" -O - http://www.iana.org/assignments/ipv4-address-
space/ipv4-address-space.xml | grep -A 4 "212"
    <prefix>212/8</prefix>
    <designation>RIPE NCC</designation>
    <date>1997-10</date>
    <whois>whois.ripe.net</whois>
    <status>ALLOCATED</status>
```

L'IANA a attribué cette adresse 212/8 à RIPE NCC (*Réseaux IP Européens Network Coordination Center*) Europe depuis octobre 1997. RIPE l'a ensuite découpée en plusieurs adresses (RIPE a la charge du netmask) pour l'attribuer à différents ISP ou sites.

À l'étape 1, on a directement « pingé » le nom du site pour obtenir son adresse IP. C'est une méthode que l'on peut qualifier de "bruyante" (on a en quelque sorte alerté la cible). Il est possible d'utiliser une méthode plus "silencieuse".

5 . On recherche les serveurs DNS de free.fr en interrogeant NIC :

```
$ whois -h whois.nic.fr free.fr | grep -E -e "domain|address|nserver"
domain:      free.fr
nserver:     freens1-g20.free.fr [212.27.60.19]
nserver:     freens2-g20.free.fr [212.27.60.20]
address:     8, rue ville l'Eveque
address:     75008 Paris
address:     Free SAS / ProXad
```

On obtient deux adresses de serveurs DNS.

6 . On interroge maintenant le premier serveur DNS pour savoir s'il connaît le site xxx (si oui on obtiendra son adresse IP) :

```
$ host -v -a tvaira.free.fr 212.27.60.19
Trying "tvaira.free.fr"
Using domain server:
Name: 212.27.60.19
Address: 212.27.60.19#53
Aliases:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41415
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;tvaira.free.fr.                        IN      ANY
;; ANSWER SECTION:
tvaira.free.fr.         3600    IN      CNAME   perso132-g5.free.fr.
perso132-g5.free.fr.    86400   IN      A       212.27.63.132
```

On obtient l'adresse IP 212.27.63.132 associée au nom tvaira.free.fr sans faire de « ping » directement.

7 . Pour terminer, on localise géographique l'adresse IP trouvée en utilisant les services de http://ipinfodb.com/

```
$ wget -q -U "" -O - http://ipinfodb.com/ip_query2.php?ip=212.27.63.132
<?xml version="1.0" encoding="UTF-8"?>
<Locations>
  <Location id="0">
    <Ip>212.27.63.132</Ip>
    <Status>OK</Status>
    <CountryCode>FR</CountryCode>
    <CountryName>France</CountryName>
    <RegionCode>A8</RegionCode>
    <RegionName>Ile-de-France</RegionName>
    <City>Paris</City>
    <ZipPostalCode></ZipPostalCode>
    <Latitude>48.8667</Latitude>
    <Longitude>2.3333</Longitude>
    <Timezone>0</Timezone>
    <Gmtoffset>0</Gmtoffset>
    <Dstoffset>0</Dstoffset>
  </Location>
</Locations>
```

Ou en récupérant le script pour Linux sur le serveur ou sur le site http://ipinfodb.com/linux_script.php :

```
$ ./getiploc 212.27.63.132 json | grep -v -E -e "{|}|Locations|]"
      "Id" : "0",
      "Ip" : "212.27.63.132",
      "Status" : "OK",
      "CountryCode" : "FR",
      "CountryName" : "France",
      "RegionCode" : "A8",
      "RegionName" : "Ile-de-France",
      "City" : "Paris",
      "ZipPostalCode" : "",
      "Latitude" : "48.8667",
      "Longitude" : "2.3333",
      "Timezone" : "0",
      "Gmtoffset" : "0",
      "Dstoffset" : "0"
```

Bonus

Écrire un script **mapiploc** qui permet de localiser dans « Google maps » une adresse IP reçue en argument.

Par exemple, pour l'adresse IP précédente, l'accès à « Google maps » sera :

```
http://maps.google.fr/maps?q=48.8667,2.3333&z=8
```

Soit avec des variables obtenues grâce à **getiploc** :

```
http://maps.google.fr/maps?q=$Latitude,$Longitude&z=8
```

Il reste à ouvrir un navigateur à partir d'un script :

```
mozilla-firefox -remote "openURL($url, new-tab)"
```

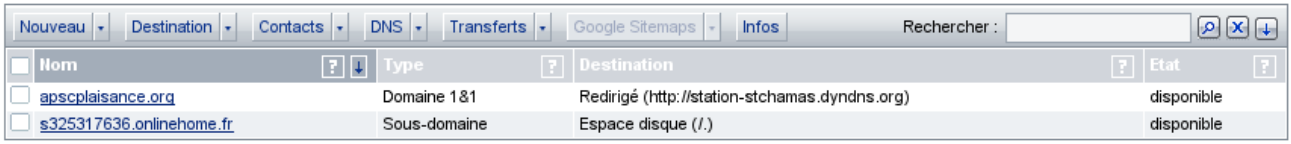
L'utilisation su script sera :

```
$ ./mapiploc 212.27.63.132
```

ANNEXES

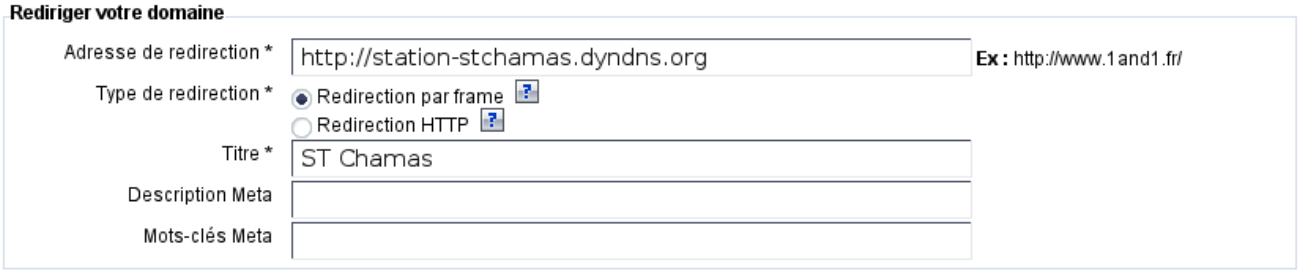
Le domaine apscplaisance.org

apscplaisance.org est un nom de domaine déposé chez 1&1 et redirigé vers :



La redirection par *frame* permet de conserver l'url (http://apscplaisance.org) dans la barre d'adresse du navigateur :

En cliquant sur « Modifier », il est possible de changer l'adresse de redirection et/ou choisir une redirection HTTP. Dans ce cas l'adresse de redirection (http://station-stchamas.dyndns.org) apparaîtra dans la barre d'adresse du navigateur :



Un compte a été crée chez dyndns.org. Le service dyndns est utilisé lorsqu'on ne dispose pas d'adresse IP publique fixe. L'adresse IP publique est donc une adresse dynamique qui est actualisée à intervalles réguliers (ici par une livebox reliée au serveur web qui héberge le site) :