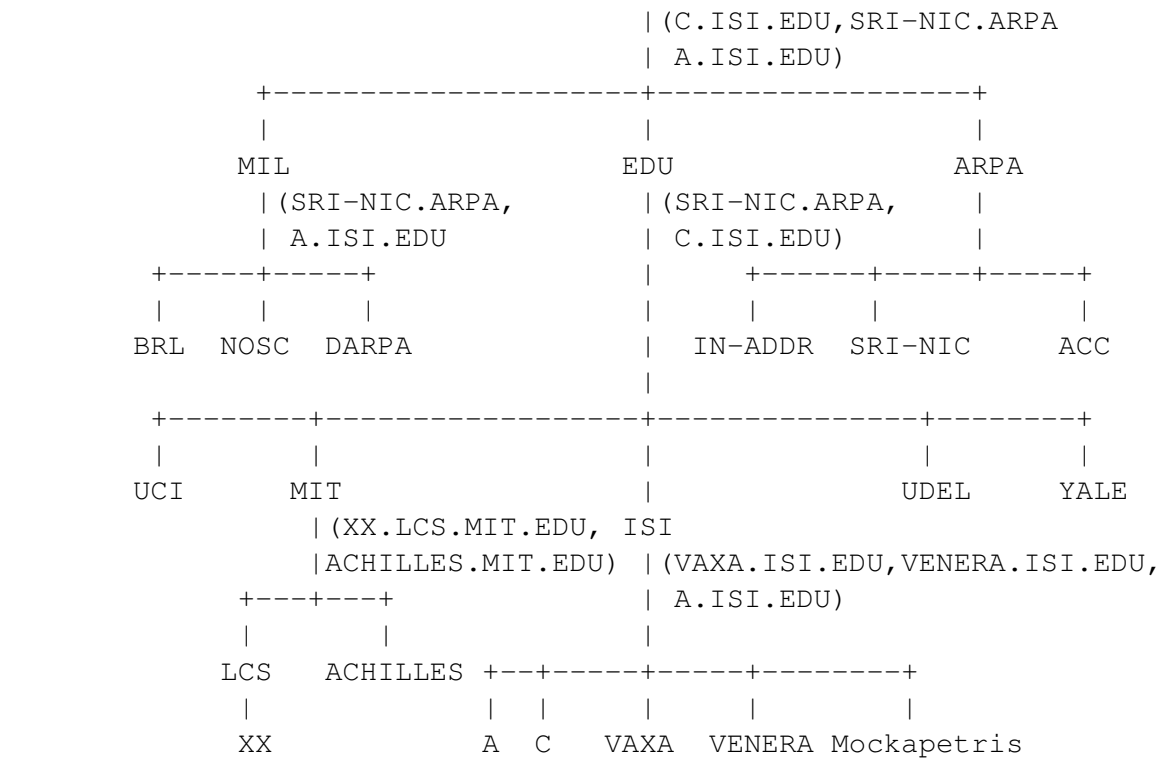
TD : Protocoles applicatifs DNS, SMTP, HTTP

1 Domain Name System (DNS)

1.1 Structure

On considère l'arbre de noms ci-dessous. Les noms entre parenthèses sont les serveurs DNS ayant autorité sur ces parties de l'arbre.



Le fichier suivant décrit les ressources associées à la zone du nœud racine de l'arbre précédent.

```
.       IN      SOA     SRI-NIC.ARPA. HOSTMASTER.SRI-NIC.ARPA. (
                        870611          ;serial
                        1800            ;refresh every 30 min
                        300             ;retry every 5 min
                        604800          ;expire after a week
                        86400)          ;minimum of a day
                NS      A.ISI.EDU.
                NS      C.ISI.EDU.
                NS      SRI-NIC.ARPA.

MIL.    86400   NS      SRI-NIC.ARPA.
        86400   NS      A.ISI.EDU.

EDU.    86400   NS      SRI-NIC.ARPA.
        86400   NS      C.ISI.EDU.
```

```
SRI-NIC.ARPA.    A       26.0.0.73
                 A       10.0.0.51
                 MX      0 SRI-NIC.ARPA.
                 HINFO   DEC-2060 TOPS20

ACC.ARPA.        A       26.6.0.65
                 HINFO   PDP-11/70 UNIX
                 MX      10 ACC.ARPA.

USC-ISIC.ARPA.   CNAME   C.ISI.EDU.

73.0.0.26.IN-ADDR.ARPA.  PTR    SRI-NIC.ARPA.
65.0.6.26.IN-ADDR.ARPA.  PTR    ACC.ARPA.
51.0.0.10.IN-ADDR.ARPA.  PTR    SRI-NIC.ARPA.
52.0.0.10.IN-ADDR.ARPA.  PTR    C.ISI.EDU.
103.0.3.26.IN-ADDR.ARPA. PTR    A.ISI.EDU.

A.ISI.EDU. 86400 A       26.3.0.103
C.ISI.EDU. 86400 A       10.0.0.52
```

1. Quels sont les ressources associées au nœud racine ? Sur quel hôte sont hébergées les données pour la zone racine ? Quels sont les serveurs de noms racine ?
2. Pourquoi définit-on en fin de fichier les ressources d'adresses pour les noms `A.ISI.EDU` et `C.ISI.EDU` ?
3. A quoi correspond la ressource `MX` associée au nœud `SRI-NIC.ARPA` ?
4. A quoi correspondent les ressources `PTR` ? Quelles sont celles qui sont définies ici ?

1.2 La commande dig

Cet exercice s'appuie sur l'utilisation de la commande `dig(1)` (domain information groper) pour l'interrogation des serveurs DNS. N'oubliez pas de terminer les noms de domaine par un point (FQDN).

1. Trouver l'adresse IP de votre machine.
2. Obtenir l'adresse IP de la machine `aragog.emi.u-bordeaux1.fr` puis de celle de `www.google.com`.
3. Quel serveur DNS a-t-on interrogé ? Examiner le fichier local `/etc/resolv.conf` (configuration des serveurs DNS) et donnez les adresses de vos serveurs.
4. Obtenir la configuration de la zone d'autorité pour le CREMI et pour l'Université Bordeaux 1 sur le domaine `u-bordeaux1.fr`.
5. Obtenir la configuration le nom du serveur de transfert de mail pour le CREMI et pour l'Université Bordeaux 1 sur le domaine `u-bordeaux1.fr`.
6. Obtenir les serveurs de noms disponibles pour le domaine `fr.` (France).
7. Trouver les serveurs de la zone racine « `.` » (Monde).
8. La résolution inverse dans le DNS est possible grâce à l'option `-x`. Quel est le nom DNS associé à l'adresse IP `68.142.254.15` ? Celui associé à la machine d'adresse `212.134.56.1` ?
9. Ecrire un script bash qui permet de retrouver les noms des machines associés à une plage d'adresse réseau. Nous prenons le réseau `147.210.18.0/24`.
10. Interroger un autre serveur DNS que ceux configurés par défaut et récupérer l'adresse IP de `www.google.com`.
11. Que fait la commande suivante : `dig +nocmd . NS +noall +answer +additional`
12. Que fait la commande suivante : `dig. +trace www.google.com`

2 Simple Mail Transfer Protocol (SMTP)

2.1 Utilisation de Telnet

Telnet est un protocole simple de connexion à distance : il permet de transmettre des caractères entre une machine locale (écran + clavier) et une machine distante.

Par défaut, un client telnet se connecte au service *telnet* mais il est possible de préciser le service de la façon suivante : *telnet M S*. Dans ce cas le client telnet se connecte au service S de la machine M.

Quel est l'effet de la commande suivante : *telnet time-nw.nist.gov daytime* ?

Décrire le protocole *daytime* d'après le résultat de cette commande.

2.2 Emulation d'un client SMTP

Afin de réaliser l'exercice suivant, nous utiliserons l'utilitaire `telnet`. Le but de ces exercices est d'appliquer manuellement les protocoles afin d'en comprendre les différentes fonctionnalités.

L'architecture du protocole SMTP est décrite figure 1, tirée de la RFC 821 :

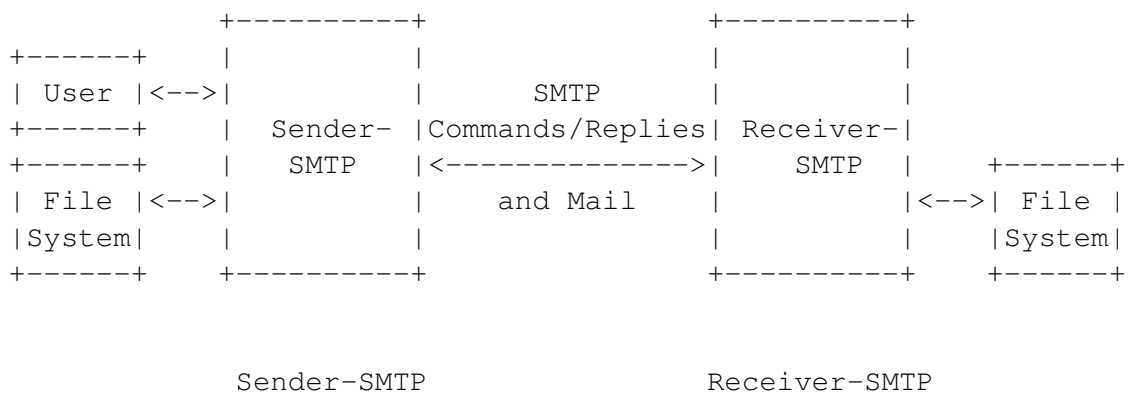


FIGURE 1 – Architecture d'un échange SMTP (source RFC 821).

Les échanges entre serveurs SMTP se basent sur un protocole constitué de procédures émises par `Sender-SMTP` et de réponses retournées par `Receiver-SMTP`. Avant cela, une connexion TCP doit être établie entre les deux entités.

Voici les principales commandes SMTP :

- `HELO` permet au `Sender-SMTP` de s'identifier auprès du `Receiver-SMTP`
- `MAIL FROM:` permet d'identifier l'émetteur du message
- `RCPT TO:` permet de donner les destinataires du mail (plusieurs commandes possibles)
- `DATA` donne le corps du mail. Celui-ci se termine par une ligne ne contenant que le caractère point « . ».
- `QUIT` permet de mettre fin à la communication.

Lors des échanges, `Receiver-SMTP` renvoie des codes de retour de 3 chiffres (le 2è précise le premier, le 3è précise le 2è). Exemple de retours :

Code	Signification
220	[domain] Service ready
221	[domain] Service closing transmission channel
250	Requested mail action ok, completed
251	User not local, will forward to [forward-path]
354	Start mail input ; end with CRLF . CRLF
451	Requested action aborted : local error in processing
500	Syntax error commande unrecognized
502	Command not implemented
550	Requested action not taken : mailbox unavailable

1. Avec l'utilitaire `telnet`, utilisez le serveur `smtp.u-bordeaux1.fr` pour vous envoyer un mail.
2. Dans votre client mail favori, lisez le mail que vous venez de vous envoyer. Que constatez-vous ?

3 HyperText Transfert Protocol (HTTP)

Le principe de HTTP est un dialogue entre un client et un serveur (requêtes et réponses). La connexion est initiée par le client (en TCP). Le client et le serveur s'échangent des *entités* (unités d'information). Celles-ci sont identifiées par des URI (Uniform Resource Identifiers) dont nous connaissons une forme simplifiées : les «Uniform Resource Locators (URL)» (RFC 1738). Une URL HTTP a la forme suivante :

```
"http:" "//" host [ ":" port ] [ abs_path ]
```

où `host` identifie un nom de machine, `port` est un numéro de port optionnel et `abs_path` est le chemin absolu de la ressource sur le serveur (si cette information manque, la racine `/` est désignée par défaut).

3.1 Format d'un message HTTP

```
Specific-line
Header
CRLF
Body
```

Une en-tête HTTP est constitué d'une liste d'options.

3.1.1 Cas d'une requête

Dans ce cas, `Specific-line` est de la forme :

```
Method SP Request-URI SP HTTP-Version CRLF
```

- `Method` indique la méthode HTTP invoquée
- `Request-URI` entité sur laquelle s'applique l'appel
- `HTTP-Version` version du protocole utilisée

Les principales méthodes sont :

- `GET` demande au serveur de renvoyer l'entité désignée au client
- `HEAD` qui demande au serveur de ne renvoyer que l'en-tête de la requête `GET` équivalente
- `POST` qui demande au serveur d'accepter l'entité jointe à la requête comme nouvelle subordonnée à la ressource identifiée.

3.1.2 Cas d'une réponse

Dans ce cas, `Specific-line`) est de la forme :

```
HTTP-Version SP Status-Code SP Reason-Phrase CRLF
```

- `HTTP-Version` version du protocole utilisée
- `Status-Code` code de retour à 3 chiffres
- `Reason-Phrase` phrase explicative

Cette "ligne" est ensuite suivie de l'entête et le cas échéant, de l'entité demandée (champ `Body`).

3.1.3 Codes de retours classiques

Code	Signification
200	OK
201	Created (POST)
202	Accepted (POST : entité reçue, traitement différé)
204	No Content (R.A.S.)
301	Moved Permanently (Redirection)
302	Moved Temporarily
304	Not Modified (GET conditionnel)
400	Bad Request
401	Unauthorized (WWW-Authenticate)
403	Forbidden
404	Not Found
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable

3.2 Analyse d'en-têtes

Conseil avant de commencer : n'hésitez pas à installer l'extension Firefox "Live HTTP Headers". Pour la suite, nous travaillerons sur le protocole HTTP. Quel est le port utilisé par ce protocole ? Nous utiliserons telnet pour envoyer directement des commandes aux serveurs ‘à la main“.

3.2.1 Méthode `HEAD`

Quels sont les entêtes de la page de garde de www.emi.u-bordeaux1.fr, www.labri.fr, www.inria.fr, www.cnrs.fr ? Analysez les différents champs. Pour chaque exemple, sur quelle machine est implanté le serveur et quel est le type de ce serveur ? Quelle est la classe de réponse ?

En utilisant la méthode `HEAD`, essayez d'obtenir les différentes classes de réponse.

1. Succès.
2. Erreur client.
3. Inchangé. Après la méthode `HEAD` et le champ d'en-tête `Host`, envoyez un champ d'en-tête du type `If-Modified-Since: Sat 05 Nov 2005 23:23:59 GMT`.
4. Redirection.

3.2.2 Méthode `GET`

Reprenez la question précédente, en utilisant cette fois la méthode `GET` pour obtenir le contenu des pages de garde de `www.emi.u-bordeaux1.fr`, `www.labri.fr`, `www.inria.fr`, `www.cnrs.fr`. Analysez les différents champs. Obtenez vous la même chose que sous Firefox ?

3.3 Utilisation d'un proxy

Comme vous utilisez régulièrement Firefox comme navigateur, savez vous si vous avez configuré ce dernier en connexion directe sur internet, ou bien en passant par un proxy ? Si vous ne le savez pas, essayez de trouver cette option de configuration sous votre navigateur.

Depuis l'université, vous pouvez utiliser le proxy `cache.u-bordeaux.fr`. Avec *telnet*, récupérez la page d'accueil de `http://www.inria.fr` au travers du proxy. Expliquer les différences entre une connexion avec et sans proxy.

Essayons d'utiliser un autre proxy. Configurez un proxy de manière manuelle en donnant le nom de serveur suivant : 200.65.0.25 (port 3128). Rechargez `http://www.inria.fr`. Qu'en concluez vous ?

Remarque : après cet exercice, n'oubliez pas de revenir à la configuration initiale.

3.4 Vos traces

En mode accès direct puis en accès via le proxy de l'université, chargez sous Firefox la rubrique : `http://www.cnil.fr/vos-libertes/vos-traces/`. Comparez le contenu de ces deux pages. Vous porterez une attention particulière aux champs relatifs à la géographie et à l'utilisation d'un proxy. Avec accès direct, votre machine a-t-elle été identifiée par le site de la CNIL ? Même question que précédemment, mais en utilisant l'accès via votre proxy.

3.5 Cookie et formulaire

Pour aller à l'essentiel, un cookie est un enregistrement d'informations par le serveur sur l'ordinateur client (le vôtre), informations que ce même serveur peut aller relire et modifier ultérieurement. Plus précisément, un cookie se compose d'un ensemble de variables (ou de champs) que le client et le serveur s'échangent lors de transactions HTTP, lesquelles variables sont tout simplement stockées sur la machine cliente. Un cookie est obligatoirement rattaché à un nom de domaine et un ensemble d'URL de telle sorte que seule une requête provenant du même serveur pourra y accéder. Par exemple, grâce à un programme CGI, le serveur a la possibilité de mettre à jour ou d'effacer un cookie. Mais pour cela, il doit spécifier tous les attributs du cookie, par conséquent seul le serveur qui a créé un cookie peut le modifier ou le supprimer.

Sachez que leur fonctionnement est assez simple sous Firefox : les cookies sont stockés dans un fichier binaire qui est une base de données SQLite. Vous pouvez modifier ce fichier en utilisant `sqlitebrowser` depuis la ligne de commande ou en installant l'extension firefox « SQLite Manager ».

Examinez votre fichier de cookies (`$HOME/.mozilla/firefox/xxx.default/cookies.sqlite`) Les tuples ont la structure suivante :

- **id** : clé primaire
- **name** : le nom laissé par le serveur qui a déposé le cookie
- **value** : ça parait évident !
- **host** : le nom du serveur qui a laissé ce cookie
- **path** : restriction sur le chemin d'accès du serveur
- **expiry** : date d'expiration du cookie en secondes depuis le 1/1/70
- **lastAccessed** : dernière utilisation en microsecondes depuis le 1/1/70
- **isSecure** : si égal à 1, le cookie ne peut transiter que *via* une connexion SSL
- **isHttpOnly** : si égal à 1, le cookie n'est pas lisible par un script côté client

Le formulaire `http://www.labri.fr/perso/franco/cookies/index.html` traite deux cookies « nom » et « fruit » et permet de les mettre à jour. Après avoir soumis le formulaire, constatez que son effet a bien été pris en compte dans la base de données des cookies de Firefox. Après combien de temps les cookies expirent-ils ?

Quelle est l'URL qui traite effectivement votre requête lorsqu'elle est soumise, dans la manipulation précédente ? Via `telnet`, invoquez directement ce script en utilisant la méthode POST. Dans un premier temps passez-lui l'état de deux cookies « nom » et « fruit ». Puis complétez la requête en passant via le POST les champs « nom » et « fruit » comme s'ils venaient du formulaire (n'oubliez pas `Content-type:application/x-www-form-urlencoded` dans l'en-tête). Vérifiez que la réponse fournie par le serveur concorde avec ce qui est attendu.

3.6 Manipulation du serveur Apache

Installer un serveur Apache et le faire tourner sur le port 8000. Essayez de mettre en place le mécanisme de redirection d'URL d'un fichier ou d'un répertoire quelconque par au minimum trois moyens différents : une directive directement dans le fichier httpd.conf (bien lire les nombreux commentaires du fichier), un fichier .htaccess et enfin par du code HTML. En utilisant `telnet` et votre navigateur, expliquez les avantages et les inconvénients de chacune des méthodes.