

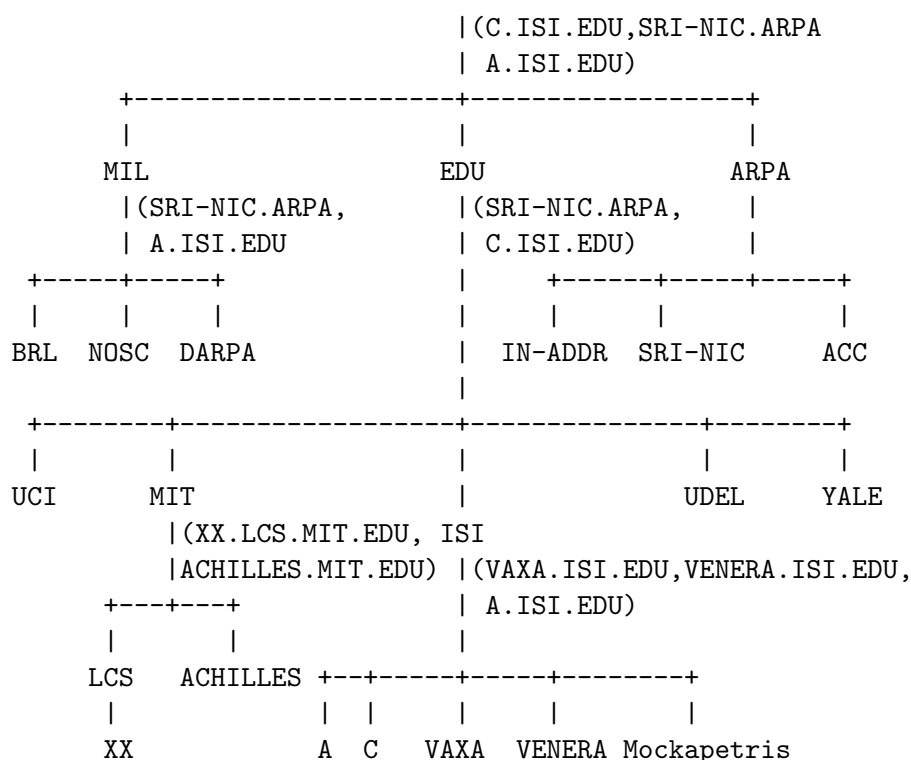
Travaux Dirigés Couches applicatives

Informatique 2ème année: 2008/2009

—Antoine Rollet - rollet@enseirb.fr —

1 Domain Name System (DNS)

► **Exercice 1.** 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 ?

► **Exercice 2.** Cet exercice s'appuie sur l'utilisation de *dig(1)* pour l'interrogation des serveurs DNS.

1. Obtenir l'adresse IP de la machine *tanit.enseirb.fr*. Quel serveur a-t-on interrogé ?
2. Obtenir la configuration de la zone d'autorité et le serveur de transfert de mail de l'ENSEIRB
3. Obtenir les serveurs de noms disponibles pour le domaine *fr*.
4. Le système DNS permet d'effectuer des conversions inverse : adresse vers nom. Obtenir le nom de la machine d'adresse *212.134.56.1*.

2 Simple Mail Transfer Protocol (SMTP)

Afin de réaliser les exercices suivants, nous utiliserons l'utilitaire *socket(1)*. 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 :

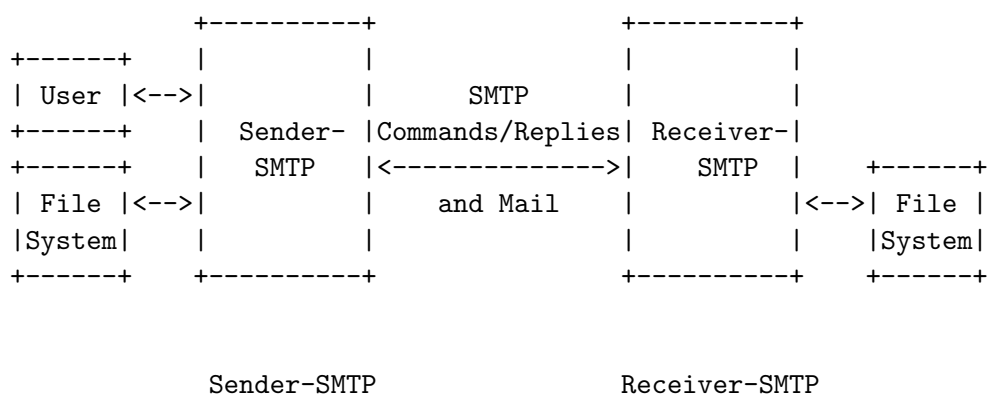


FIG. 1 – Architecture d'un échange SMTP (source RFC 821).

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 |

►Exercice 3.

- Avec l'utilitaire *socket*, utilisez le serveur *smtp.enseirb.fr* pour vous envoyer un mail.
- 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 (le port par défaut a le numéro 80) et **abs_path** est le chemin absolu de la ressource sur le serveur (si cette information manque, la racine / est désignée par défaut).

Format d'un message HTTP

Specific-line

Header

CRLF

Body

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

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 différentes méthodes⁽¹⁾ 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.

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**).

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 |

►**Exercice 4.** A l’aide de l’utilitaire *socket*, obtenir la page d’accueil du serveur web de l’enseirb. Analysez les différents champs.

⁽¹⁾La version 1.1 de HTTP en ajoute d’autres.