



## 4-A : Les applications d'Internet

### Le modèle Client/Serveur

La connexion à distance (telnet et ssh)

Le courrier électronique (SMTP, POP, IMAP, Webmail)

La résolution des noms (DNS)

Les autres applications (FTP, NFS, LDAP...)

## Copyright

- Copyright © 2022 Olivier Glück; all rights reserved
- Ce support de cours est soumis aux droits d'auteur et n'est donc pas dans le domaine public. Sa reproduction est cependant autorisée à condition de respecter les conditions suivantes :
  - Si ce document est reproduit pour les besoins personnels du reproducteur, toute forme de reproduction (totale ou partielle) est autorisée à la condition de citer l'auteur.
  - Si ce document est reproduit dans le but d'être distribué à des tierces personnes, il devra être reproduit dans son intégralité sans aucune modification. Cette notice de copyright devra donc être présente. De plus, il ne devra pas être vendu.
  - Cependant, dans le seul cas d'un enseignement gratuit, une participation aux frais de reproduction pourra être demandée, mais elle ne pourra être supérieure au prix du papier et de l'encre composant le document.
  - Toute reproduction sortant du cadre précisé ci-dessus est interdite sans accord préalable écrit de l'auteur.

## Plan

- Le modèle Client/Serveur
  - Qu'est-ce que le modèle Client/Serveur, Les sockets et les numéros de port, l'architecture TCP/IP
- La connexion à distance (telnet, ssh et X)
  - Connexion locale et distante, L'application telnet, ssh, et X
- Le courrier électronique (SMTP, POP, IMAP, Webmail)
  - Composants et transmission du courriel, Configuration d'un client mail, Protocoles SMTP, POP et IMAP, Webmail
- La résolution des noms (DNS)
  - Les services fournis par le DNS, Un système distribué, Qu'est-ce qu'un domaine ? Les serveurs racine, Les messages DNS, host
- Les autres applications (FTP, NFS, LDAP...)



## Le modèle Client/Serveur

Qu'est-ce que le modèle client/serveur ?

Les sockets et les numéros de port

L'architecture TCP/IP

## Les applications vues dans ce cours

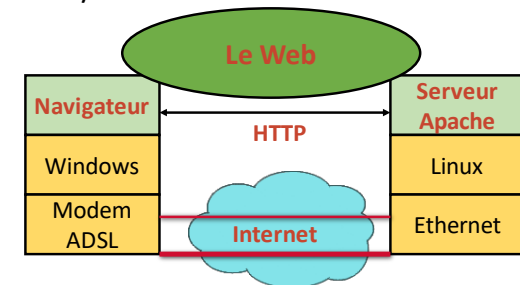
- Le web (**HTTP**)
- La connexion à distance (**telnet**, **ssh** et **X**)
- Le courrier électronique (**SMTP**, **POP**, **IMAP**, **Webmail**)
- La résolution des noms (**DNS**)
- Le transfert de fichiers (**FTP**)
- L'accès aux fichiers distants (**NFS**, **SMB**)
- L'annuaire fédérateur (**LDAP**)

Toutes ces applications fonctionnent selon le modèle Client/Serveur !

## Par exemple le Web...

- Une application d'Internet qui permet le partage de documents liés entre eux et appelés "pages web"
- Une page web peut contenir du texte, des images, des programmes, des liens vers d'autres pages web...
- Fonctionne en mode Client/Serveur au dessus de l'architecture TCP/IP

L'application est répartie sur le client et le serveur qui dialoguent selon un protocole applicatif spécifique



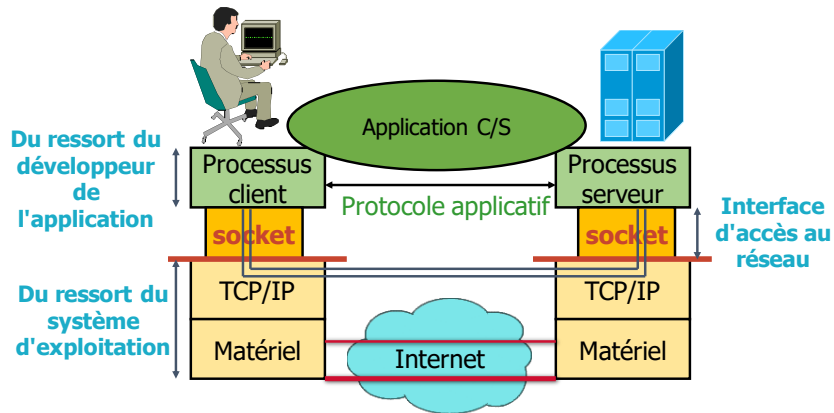
## Le modèle Client / Serveur

- Idée : l'application est répartie sur différents sites pour optimiser le traitement, le stockage...
- Le client
  - effectue une demande de service auprès du serveur (**requête**)
  - initie le contact (parle en premier), ouvre la session
- Le serveur
  - est la partie de l'application qui offre un service
  - est à l'écoute des requêtes clientes
  - répond au service demandé par le client (**réponse**)
- Le client et le serveur ne sont pas identiques, ils forment un système coopératif
- Un serveur peut répondre à plusieurs clients simultanément

## Le modèle Client / Serveur

- Une application Client/Serveur, c'est
  - **une partie cliente** qui exécute des requêtes vers un serveur
  - **une partie serveur** qui traite les requêtes clientes et y répond
  - **un protocole applicatif** qui définit les échanges entre un client et un serveur
  - **un accès via une API** (interface de programmation, généralement les sockets) à la couche de transport des messages
- Bien souvent les parties cliente et serveur ne sont pas écrites par les mêmes programmeurs --> rôle important des RFCs qui spécifient le protocole !

# Les sockets, interface d'accès au réseau

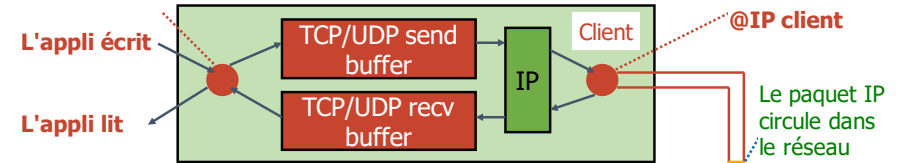


Une socket : interface locale à l'hôte, créée par l'application, contrôlée par l'OS  
 Porte de communication entre le processus client et le processus serveur

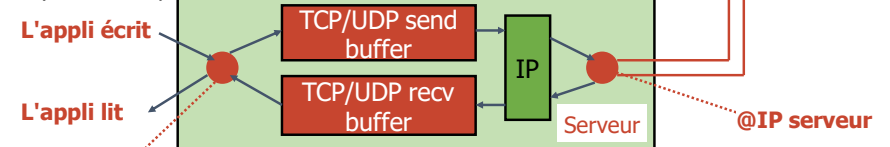
# Les sockets et les numéros de port

Un échange Client/Serveur = (@IP\_src,port\_src,@IP\_dest,port\_dest)

Port 5004 utilisé par le navigateur web



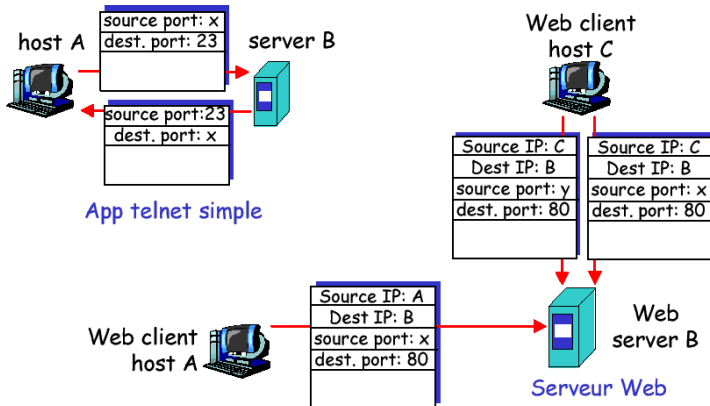
Une socket est un fichier virtuel local dans lequel l'application lit ou écrit. Elle est associée à un numéro de port local et des zones d'émission/réception de TCP ou UDP attribués par le système d'exploitation



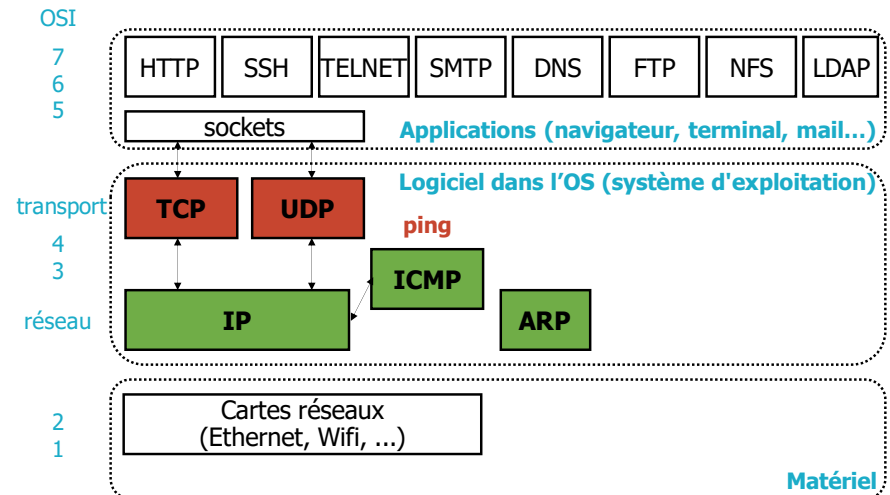
Port 80 utilisé par le serveur web

# Les numéros de ports et les adresses IP

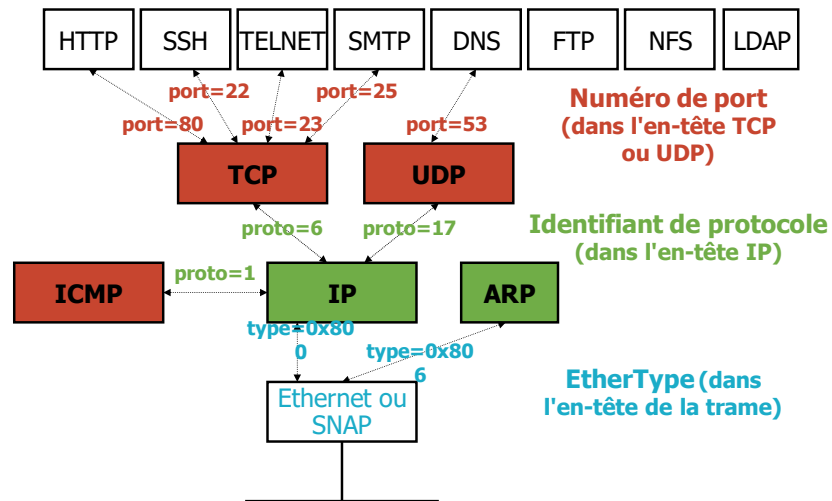
Les ports inférieurs à 1024 sont réservés pour les serveurs car le port du serveur doit être fixé et connu à l'avance pour que le client puisse faire sa demande.



# L'architecture de TCP/IP



## La désencapsulation : identifier la couche >



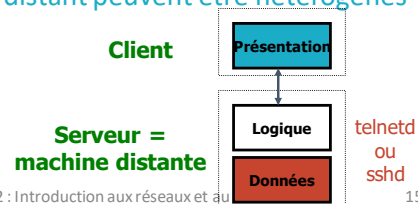
## La connexion à distance (telnet, ssh et X)

Connexion locale et distante  
 L'application telnet  
 L'application ssh  
 L'application X

## Qu'est-ce qu'une connexion à distance ?

- Application permettant à un utilisateur de se connecter à une machine distante pour en prendre partiellement le contrôle c'est à dire **exécuter des commandes** autorisées
  - à partir d'un terminal local et à condition que cet utilisateur dispose d'un accès autorisé à cette machine (login, mot de passe...)
- Les commandes saisies localement au clavier s'exécutent sur la machine distante
  - Les environnements local et distant peuvent être hétérogènes (windows-->unix, ...).

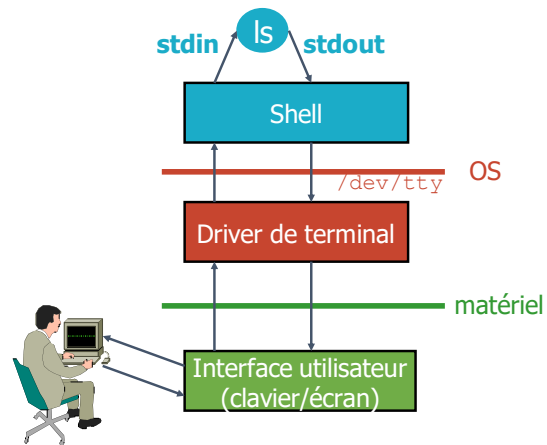
### Connexion à distance :



## La connexion à distance

- Plusieurs protocoles
  - telnet** : le standard (existe sur de nombreuses plate-formes)
  - rlogin** : uniquement entre machines unix
  - ssh** : sécurisé (authentification + chiffrement), peut transporter le DISPLAY c'est à dire gérer des fenêtres distantes
- La connexion à distance a besoin d'interactivité
  - Tout ce qui est tapé au clavier sur le client est envoyé au serveur à travers la connexion puis exécuté par lui.
  - Tout ce qui est envoyé par le serveur au client s'affiche dans le terminal sur l'écran de la machine cliente.

## Fonctionnement d'une connexion locale

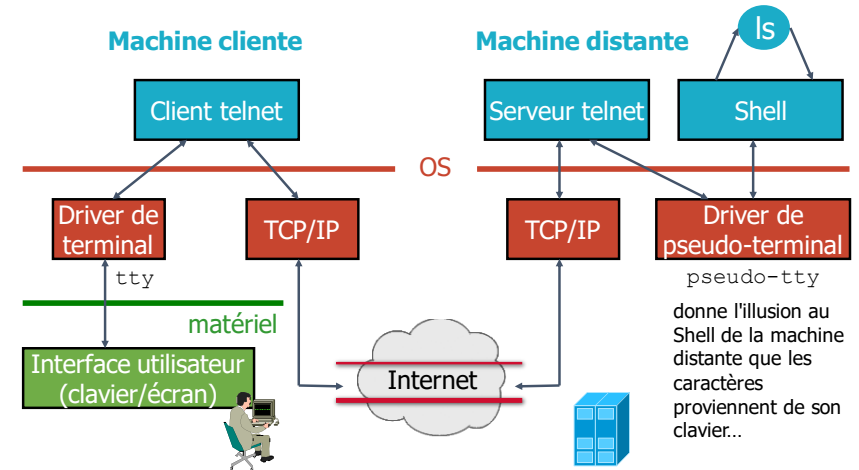


Olivier Glück

L1 Math-info UCBL - LIFASR2 : Introduction aux réseaux et au web

17

## Fonctionnement d'une connexion distante



Olivier Glück

L1 Math-info UCBL - LIFASR2 : Introduction aux réseaux et au web

18

## Telnet : un protocole ET une application

### TELecommunication NETwork protocol

- Un des premiers standard de l'Internet : RFC 854,855 (1983)
- Le serveur attend sur son port 23 les demandes de connexion TCP qui arrivent des clients
- Authentification par login/mdp sur le shell distant (attention : le mot de passe est transmis en clair)
- Quand un caractère est tapé au clavier, il est envoyé au serveur qui renvoie un "écho" du caractère ce qui provoque son affichage dans le terminal local
- Prise en compte de l'hétérogénéité entre le système local et le système distant
  - telnet d'une machine Windows vers une machine Unix

Olivier Glück

L1 Math-info UCBL - LIFASR2 : Introduction aux réseaux et au web

19

## Le client telnet

- Les différentes exécutions possibles (côté client)
  - sans argument (paramétrer sa connexion distante)  
`telnet`
  - par le nom de la machine distante (DNS+port 23)  
`telnet nom_du_serveur`
  - par l'adresse IP de la machine distante (port 23)  
`telnet adr_IP_du_serveur`
  - accès à un autre service (connexion sur un autre port)  
`telnet adr_IP_du_serveur numéro_port`

Olivier Glück

L1 Math-info UCBL - LIFASR2 : Introduction aux réseaux et au web

20

## telnet utilisé comme client web

```
lima
ogluck@lima:~$ telnet www.ens-lyon.fr 80
Trying 140.77.167.3...
Connected to pingouin.ens-lyon.fr.
Escape character is '^]'.
HEAD /index.html HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 01 Mar 2004 18:37:42 GMT
Server: Apache/1.3.26 (Unix) Debian GNU/Linux PHP/4.1.2
Last-Modified: Tue, 23 Oct 2001 08:59:37 GMT
ETag: "3fa02b-100e-3bd53179"
Accept-Ranges: bytes
Content-Length: 4110
Connection: close
Content-Type: text/html; charset=iso-8859-1

Connection closed by foreign host.
ogluck@lima:~$
```

L'application telnet est utilisée mais pas le protocole Telnet car il ne s'agit pas d'une connexion à distance

## SSH : un shell distant sécurisé

### Secure SHell

- Les communications sont cryptées
- Authentification à base de clés
- Un des seuls protocoles de connexion à distance qui passe les pare-feux de nos jours
- Permet de transporter des fenêtres graphiques via le tunnel SSH avec **ssh -X**
- Le serveur attend sur son port 22 les demandes de connexion TCP qui arrivent des clients
- Pas encore de RFC (ietf-internet-draft)

## Les commandes ssh et scp

- Connexions à distance

```
ssh -l user hostname
```

```
ssh user@hostname
```

- Exécution de commande à distance

```
ssh -l user hostname cmd
```

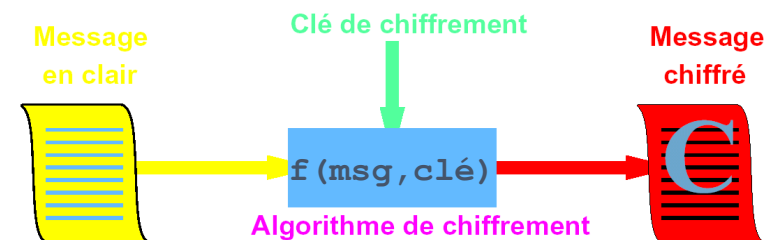
```
ssh user@hostname cmd
```

- Copie de fichiers à distance

```
scp file1 file2 user@hostname:
```

```
scp -r dir user@hostname:/tmp
```

## Principe du chiffrement



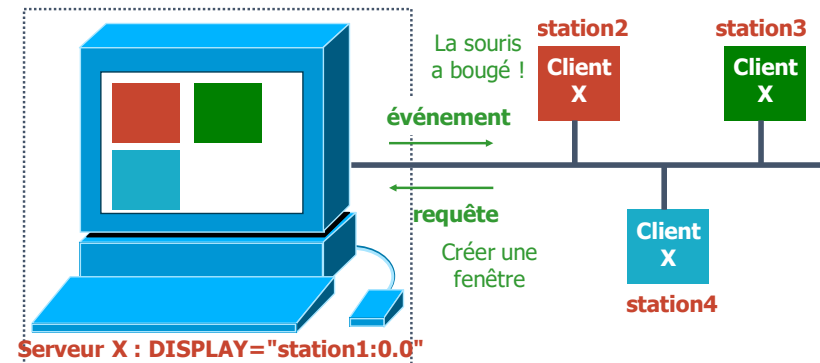
- La qualité de la sécurité dépend
  - du secret de la clé
  - de la longueur de la clé (plus il y a de bits, plus il est difficile d'essayer toutes les clés)
  - de la difficulté d'inversion de l'algorithme de chiffrement

## X : une application qui gère les fenêtres

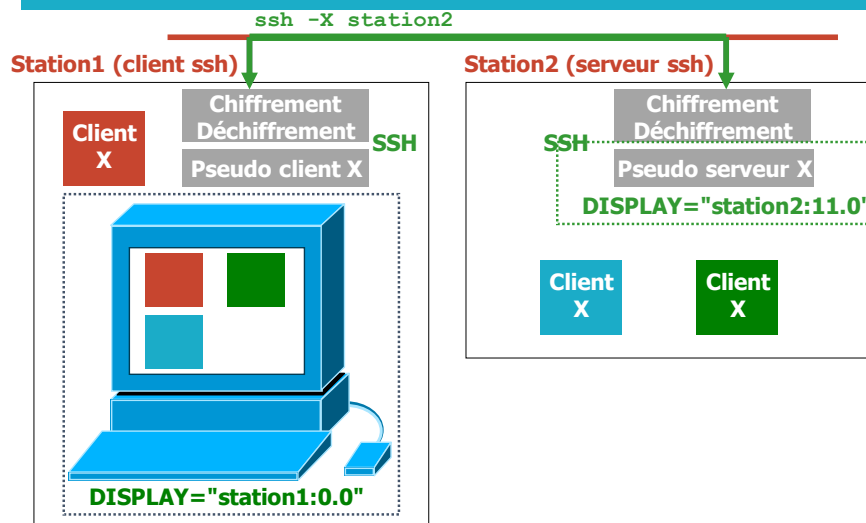
- Système de multi-fenêtrage sous Unix
  - appelé X ou X Window System ou X11
  - ensemble de programmes réalisant l'interface Homme/Machine basé sur l'utilisation des périphériques (clavier, souris, écran, ...)
- X est constitué de plusieurs entités
  - un **serveur X** : gère le matériel (clavier, écran, ...) et leur utilisation par les applications graphiques
  - des **clients X** : applications graphiques qui nécessitent un serveur X pour afficher les fenêtres (`xemacs`, `xterm`, `xcalc`, `xv`, ...)
  - le **protocole X** : fait communiquer les clients et le serveur

## X : une application qui gère les fenêtres

- Système réparti : permet de travailler sur plusieurs machines simultanément
  - les clients X peuvent s'exécuter sur des machines distantes (3 connexions TCP dans l'exemple)



## SSH -X : X11 forwarding



Université Claude Bernard Lyon 1



Faculté des Sciences et Technologies  
Dpt Informatique

## Le courrier électronique (SMTP, POP, IMAP, webmail)

- Les composants du courrier électronique
  - La transmission d'un courriel
  - Configuration d'un client mail
  - Les types MIME
- Les protocoles SMTP, POP et IMAP
  - Qu'est-ce qu'un Webmail ?
  - Format d'une adresse mail

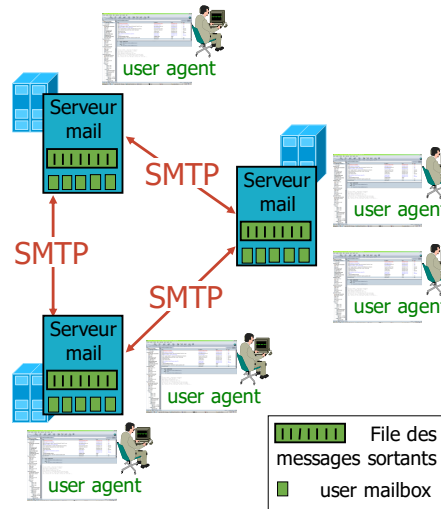
## Les composants du courrier électronique

### 4 composants principaux :

- des agents utilisateurs
- des serveurs de mail
- un protocole de transfert de mail : *Simple Mail Transfer Protocol (SMTP)*
- un protocole d'accès à la boîte aux lettres (POP, IMAP, ...)

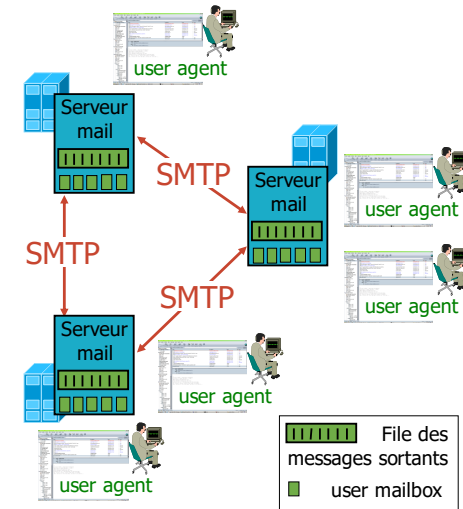
### Les agents utilisateurs :

- composition, édition, lecture du courrier électronique
- ex : Eudora, Outlook, elm, pine, Thunderbird
- un agent utilisateur dialogue avec un serveur pour émettre/recevoir des messages

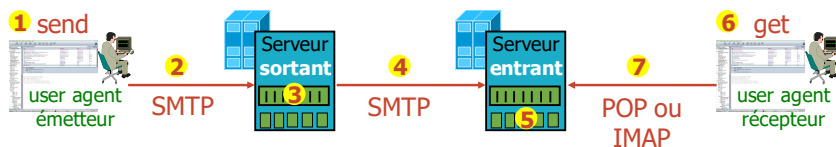


## Les composants du courrier électronique

- Les messages entrants et sortants sont stockés sur le serveur
- La boîte aux lettres de chaque utilisateur contient les messages entrants (à lire)
- File d'attente des messages mail sortants (à envoyer)
- Protocole SMTP entre les serveurs de mail pour l'envoi des messages
  - modèle C/S : Client (serveur de mail émetteur) - Serveur (serveur de mail récepteur)
  - le client se connecte sur le port 25/TCP du serveur pour transférer son message



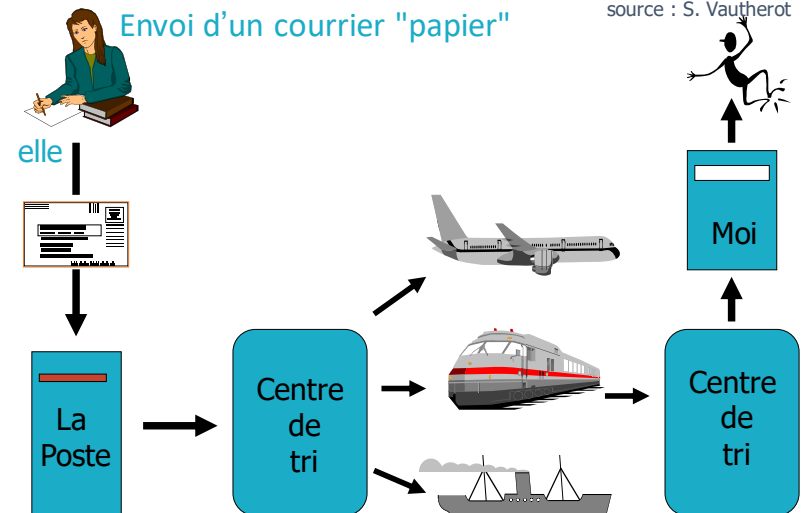
## La transmission d'un courrier électronique



### Les protocoles d'accès : consultation de sa boîte aux lettres (après authentification)

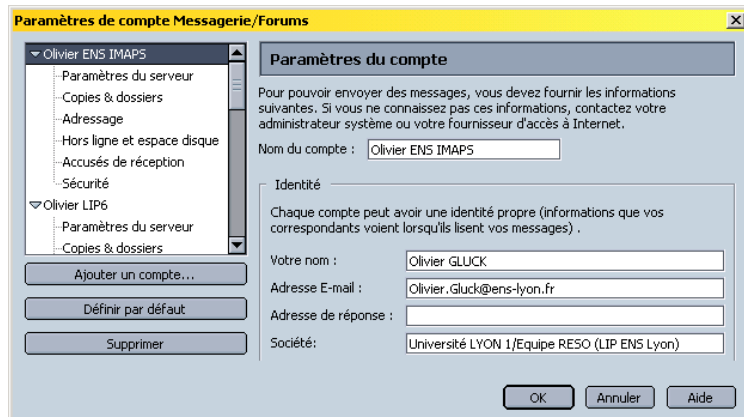
- POP3 : Post Office Protocol v3 [RFC 1939]**
  - autorisation (agent <--> server) et téléchargement
- IMAP4 : Internet Message Access Protocol v4 [RFC 3501]**
  - plus de caractéristiques, plus complexe, plus récent
  - manipulation de messages stockés sur le serveur
- HTTP (Webmail) : Hotmail , Yahoo! Mail, ...**

## Analogie avec le courrier "postal"



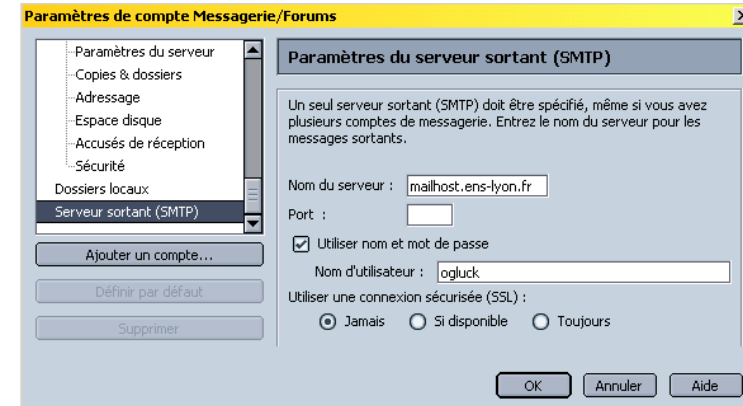


## Configuration d'un client mail



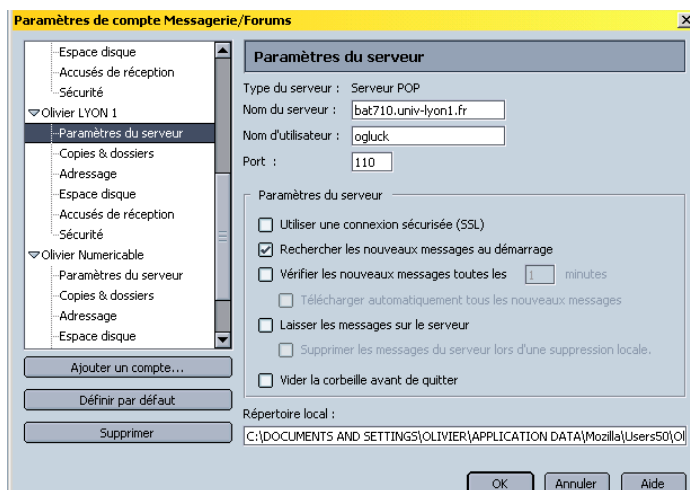
L'identité permet de renseigner une partie de l'en-tête des messages envoyés

## Configuration d'un client mail



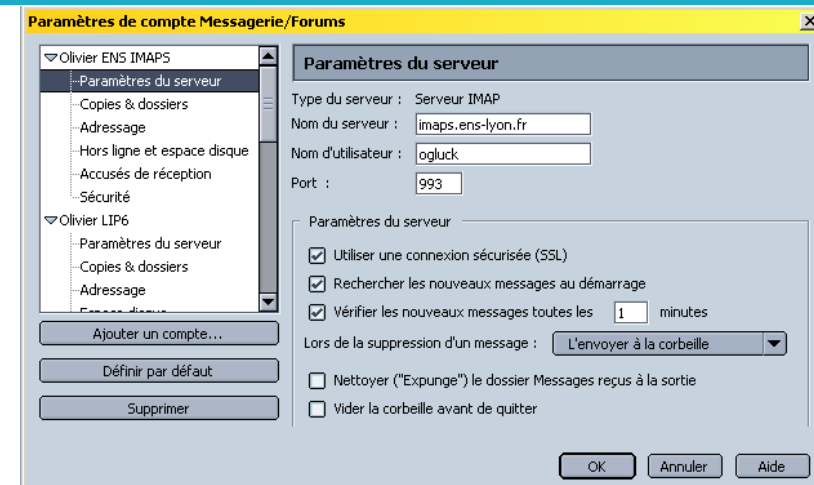
Paramétrage du serveur sortant

## Configuration d'un client mail



Oli **POP** : les messages sont rapatriés dans le répertoire local

## Configuration d'un client mail

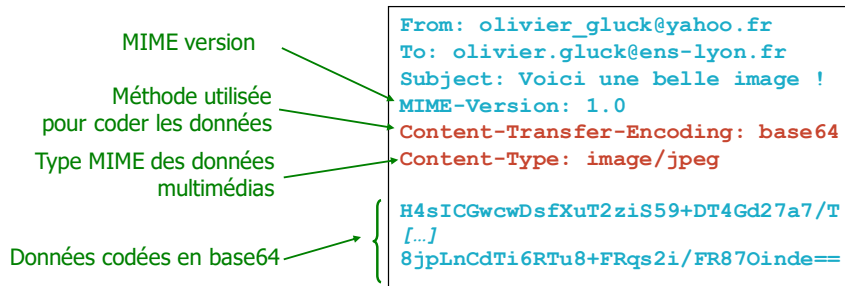


**IMAP** : les messages restent sur le serveur sauf s'ils sont supprimés, déplacés, ...

# Les types MIME [RFC 2045, 2056]

## Content-Type: type/subtype; parameters

- Lignes supplémentaires dans l'en-tête du message pour déclarer un type MIME et un encodage
- Content-type est généralement positionné à partir de l'extension du document demandé (/etc/mime.types)



# Un mail avec pièce jointe : type Multipart

```
From: olivier_gluck@yahoo.fr
To: olivier.gluck@ens-lyon.fr
Subject: Voici une belle image mais avec du texte !
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=98766789
```

```
--98766789
Content-Transfer-Encoding: quoted-printable
Content-Type: text/plain
```

```
Cher Olivier,
Voici une photo de nos dernieres vacances !
--98766789
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
H4sICGYRMTQAA3NsaWRlcy5wDsfXuT2ziS59+DT4Gd275a
56o7LlgSjBfNiWpSqsfw6rvLxPgSxIlVnk64i54ftrKi67/T
[...]
8jplnCdTi6RTu8+FRqs2i/RTuy56pLYbYVsa1fdvUjHrtV6g
RTf4/hy67fgIIVDFeR+rtYuNFR87Oinde==
--98766789--
```

# Les commandes SMTP

Commande	Description
<b>HELO</b> nom_client	identifie le client SMTP ; établit la connexion
<b>MAIL</b> From: <@exp>	identifie l'expéditeur du message
<b>RCPT</b> To: <@dest>	désigne le destinataire du message
<b>DATA</b>	indique le début du message (en-tête+corps)
<b>QUIT</b>	termine la connexion
<b>NOOP</b>	pas d'opération ; force le serveur à répondre
<b>RSET</b>	réinitialisation de la saisie de données (DATA)

```
ogluck@lima:~$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 lima.cri2000.ens-lyon.fr ESMTPEX 3.35 #1 Mon, 22 Mar 2004 11:57:58 +0100
HELP
214-Commands supported:
214- HELO EHLO MAIL RCPT DATA AUTH
214 NOOP QUIT RSET HELP
QUIT
221 lima.cri2000.ens-lyon.fr closing connection
Connection closed by foreign host.
ogluck@lima:~$
```

# Un échange SMTP

```
"Nice to meet you !"
... sender OK
... receiver OK
Début de l'en-tête (DATA)
Fin de l'en-tête (ligne vierge)
Fin du message (<CR><LF> . <CR><LF>)
```

ogluck@lima:~\$ telnet mailhost.ens-lyon.fr 25

Trying 140.77.1.22...

Connected to oceanite.ens-lyon.fr.

Escape character is '^]'.

220 oceanite.ens-lyon.fr ESMTPEX Postfix

HELO lima.ens-lyon.fr

250 oceanite.ens-lyon.fr

MAIL FROM: ogluck

250 Ok

RCPT TO: <olivier.gluck@ens-lyon.fr>

250 Ok

RCPT TO: <ogluck@bat710.univ-lyon1.fr>

250 Ok

DATA

354 End data with <CR><LF>.<CR><LF>

From: olivier\_gluck@yahoo.fr

To: Olivier <ogluck@bat710.univ-lyon1.fr>

Cc: GLUCK <olivier.gluck@ens-lyon.fr>

Subject: un dialogue SMTP

Voici un exemple d'échange !

250 Ok: queued as 119F332015D

MAIL FROM: ogluck

250 Ok

RCPT TO: <olivier\_gluck@yahoo.fr>

250 Ok

DATA

354 End data with <CR><LF>.<CR><LF>

From: olivier.gluck@ens-lyon.fr

To: olivier-gluck@yahoo.fr

Un deuxième mail à envoyer...

250 Ok: queued as 1A18A3200F6

QUIT

221 Bye

Connection closed by foreign host.

ogluck@lima:~\$

## Exemple d'en-têtes

Sujet	Expéditeur	Date	Etat
Cours SMTP	Olivier GLUCK	19:17	• Lire
Réunion d'information : les journées MIAGe sont à Lyon !	planification@jnm2004...	18/03/2004 13...	• Lire
Home page	admin@bat710.univ-ly...	18/03/2004 12...	• Répondu

```

Sujet: Cours SMTP
De: Olivier GLUCK <Olivier.Gluck@ens-lyon.fr>
Date: 19:17
A: Olivier GLUCK <ooluck@bat710.univ-lyon1.fr>, Olivier GLUCK <Olivier.Gluck@ens-lyon.fr>
X-UIDL: eb4d1caa77080000
X-Mozilla-Status: 0001
X-Mozilla-Status2: 00000000
Received: from ens-lyon.fr (gluck.cri2000.ens-lyon.fr [140.77.13.102]) by oceanite.ens-lyon.fr (Postfix) with ESMTMP id 3CC61320118; Fri, 19 Mar 2004 19:17:26 +0100 (CET)
X-From_: Olivier.Gluck@ens-lyon.fr Fri Mar 19 19:17:38 2004
Return-Path: <Olivier.Gluck@ens-lyon.fr>
Message-ID: <40583945.2050607@ens-lyon.fr>
Organization: Université LYON 1/Equipe RESO (LIP ENS Lyon)
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; fr-FR; rv:1.0.2) Gecko/20030208 Netscape/7.02
X-Accept-Language: fr-fr, fr
MIME-Version: 1.0
Content-Type: text/plain; charset=ISO-8859-15; format=flowed
Content-Transfer-Encoding: 8bit
X-Virus-Scanned: by AMaViS snapshot-20020222

Voilà un exemple !
    
```

Olivier Glück L1 Math-info UCBL - LIFASR2 : Introduction aux réseaux et au web

41

## Exemple de contenu d'une bal

```

SunOs:/users/cao/glucko
glucko@ducas [SunOs] ~$ cat /var/mail/glucko
From Olivier.Gluck@numericable.fr Mon Mar 22 20:04:51 2004
Return-Path: <Olivier.Gluck@numericable.fr>
Received: from isis.lip6.fr (isis.lip6.fr [132.227.60.21])
  by asin.lip6.fr (8.11.6p3/8.11.6) with ESMTMP id i2MJ4pM14034
  for <Olivier.Gluck@asin.lip6.fr>; Mon, 22 Mar 2004 20:04:51 +0100 (CET)
Received: from oughtred.numericable.net (oughtred.numericable.net [80.236.0.153])
  by isis.lip6.fr (8.12.11/jtpda-5.4+vector) with ESMTMP id i2MJ4p13032651
  for <Olivier.Gluck@lip6.fr>; Mon, 22 Mar 2004 20:04:51 +0100
X-pt: isis.lip6.fr
Received: (qmail 15060 invoked from network); 22 Mar 2004 19:04:45 -0000
Received: from unknown (HELO numericable.fr) ([81.220.146.234])
  (envelope-sender <Olivier.Gluck@numericable.fr>)
  by oughtred.numericable.net (qmail-ldap-1.03) with SMTP
  for <Olivier.Gluck@lip6.fr>; 22 Mar 2004 19:04:45 -0000
Message-ID: <405F38D6.7020804@numericable.fr>
Date: Mon, 22 Mar 2004 20:04:54 +0100
From: Olivier GLUCK <Olivier.Gluck@numericable.fr>
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; fr-FR; rv:1.0.2) Gecko/20030208 Netscape/7.02
X-Accept-Language: fr-fr, fr
MIME-Version: 1.0
To: Olivier Gluck <Olivier.Gluck@lip6.fr>
Subject: Cours SMTP
Content-Type: text/plain; charset=ISO-8859-15; format=flowed
Content-Transfer-Encoding: 8bit
X-Scanned-By: isis.lip6.fr

Voilà le contenu d'une BAL contenant 1 seul message !
    
```

Une BAL n'est rien de plus qu'un fichier ! (généralement /var/mail/user\_login)

42

## Le protocole POP3 [RFC 1939]

### Phase d'autorisation

#### Commandes client :

**user** : déclare username  
**pass** : password

#### Deux réponses possible du serveur :

**+OK**  
**-ERR**

### Phase de transaction

**list** : liste les numéros de messages et leur taille

**retr** : rapatrie un message à partir de son numéro

**dele** : efface un message

**quit**

```

S: +OK POP3 server ready
C: user alice
S: +OK
C: pass hungry
S: +OK user successfully logged on

C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <contenu du message 1>
S: .
C: dele 1
C: retr 2
S: <contenu du message 2>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
    
```

Olivier Glück L1 Math-info UCBL - LIFASR2 : Introduction aux réseaux et au web

43

## Le protocole POP3 [RFC 1939]

### POP3 est extrêmement simple

- permet uniquement de télécharger des messages depuis le serveur en laissant éventuellement une copie de ceux-ci dans la BAL de l'utilisateur
- pas adapté aux utilisateurs nomades
  - impossible de gérer des répertoires sur le serveur
  - impossible de gérer les messages en les laissant sur le serveur

IMAP répond à cette problématique au prix d'un protocole beaucoup plus complexe

Olivier Glück

L1 Math-info UCBL - LIFASR2 : Introduction aux réseaux et au web

44

## Le protocole IMAP [RFC 3501]

- IMAP permet la gestion distante des messages
  - Associe un message à un répertoire distant sur le serveur
  - Permet à l'utilisateur de faire une recherche dans les messages sur le serveur
  - Permet de ne consulter que des extraits de messages (par exemple que l'en-tête ou que la partie texte d'un message *multipart*...)
  - Contrairement à POP3, IMAP conserve des informations d'état sur chaque utilisateur (noms des répertoires, listes des messages qu'ils contiennent...)

Plus d'infos : <https://tools.ietf.org/html/rfc3501>

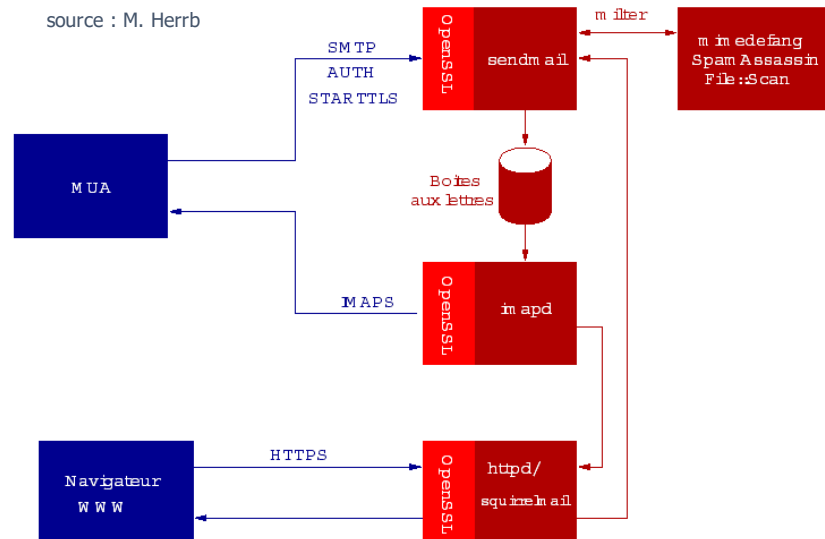
## Qu'est-ce qu'un Webmail ?

- L'utilisateur utilise un **navigateur Web** comme agent utilisateur pour consulter/envoyer ses courriers
- Le navigateur fait des requêtes **HTTP** (ou HTTPS) vers un serveur Web qui s'interface avec les serveurs SMTP/IMAP
  - Le serveur HTTP exécute des scripts qui font des requêtes
    - IMAP pour communiquer avec le serveur IMAP qui stockent les messages reçus par l'utilisateur
    - SMTP pour envoyer les messages de l'utilisateur
- Avantages du Webmail
  - adapté aux utilisateurs itinérants
  - pas besoin d'un agent utilisateur particulier, seule une connexion Internet avec Navigateur Web est nécessaire

## Format d'une adresse mail

- Adresse d'un destinataire : **bal@nom\_domaine**
- Problème :
  - **bal** n'est pas forcément le login de l'utilisateur : souvent de la forme *prenom.nom* qui est un alias vers le login
  - **nom\_domaine** n'est pas forcément le nom du serveur de mail contenant la boîte aux lettres pour avoir des adresses plus courtes et plus faciles à retenir
  - **bal** peut représenter plusieurs destinataires (liste de mail)
- Exemple : **Olivier.Gluck@ens-lyon.fr**  
Olivier.Gluck est un alias vers `/var/mail/ogluck`  
ens-lyon.fr pointe vers `mailhost.ens-lyon.fr`  
(enregistrement de type MX dans le DNS)

## Architecture d'un serveur MAIL





## La résolution des noms (DNS)

Les services fournis par le DNS  
Un système distribué  
Qu'est-ce qu'un domaine DNS ?  
Les serveurs racine  
Les messages DNS  
La commande host

## DNS : Domain Name System

- Les personnes ont plusieurs identifiants
  - Le nom mais aussi #sécu, #Passeport, #téléphone
- Les machines et les routeurs aussi
  - L'adresse IP (32 bits ou 128 bits) est l'équivalent du numéro de téléphone
  - Le nom de la machine est ce que l'on utilise dans l'URL par exemple :
    - `www.univ-lyon1.fr`
    - `www.education.gouv.fr`
- Le DNS fait le lien entre les adresses IP utilisées pour acheminer les paquets et les noms de machine utilisés par les utilisateurs ou les applications

## DNS : Domain Name System

- C'est une base de données **distribuée**
  - Il y a plein de serveurs de noms dans le monde. Chaque serveur stocke les noms et les adresses IP dont il est responsable.
- C'est un protocole applicatif comme HTTP, SMTP...
  - Les machines clientes et les serveurs de noms communiquent pour effectuer la traduction d'un nom en adresse IP.
  - Le DNS est utilisé par les applications clientes pour trouver les adresses IP des serveurs mais n'est pas utilisé directement par l'application comme SMTP...
  - Le DNS fonctionne selon le modèle Client/Serveur comme les autres applications.
  - Le serveur utilise le port 53/UDP (ou 53/TCP mises à jour)
  - RFC 1034, 1035, 2181, ...

## Les services fournis par le DNS

- Le service principal : obtenir l'adresse IP d'un serveur  
**Requête DNS** : Quelle est l'adresse IP de `www.univ-lyon1.fr` ?  
**Réponse DNS** : `134.214.126.72`  

```
olivier.gluck@lifasr2:~$ host www.univ-lyon1.fr
www.univ-lyon1.fr is an alias for ksup.univ-lyon1.fr.
ksup.univ-lyon1.fr has address 134.214.126.72
```
- Autres exemples de services fournis par le DNS
  - Donner plusieurs noms à une machine (Alias)
  - Donner plusieurs adresses IP à un serveur (Répartition de la charge)
  - Trouver le nom d'un serveur mail (*Mail server aliasing*)  

```
olivier.gluck@lifasr2:~$ host univ-lyon1.fr | grep mail
univ-lyon1.fr mail is handled by 5 smtpbv.univ-lyon1.fr.
olivier.gluck@lifasr2:~$ host smtpbv.univ-lyon1.fr
smtpbv.univ-lyon1.fr has address 134.214.126.92
```

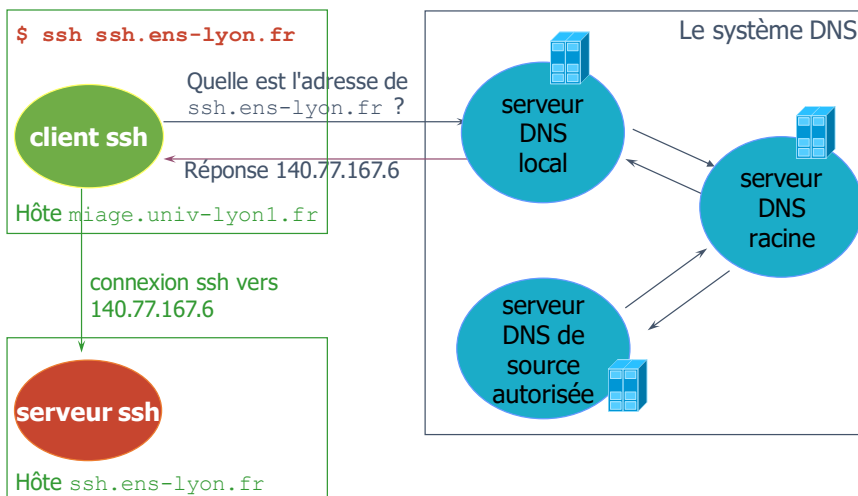
## Pourquoi le DNS est un système distribué ?

- Pour l'utilisateur, le DNS n'est qu'une boîte noire mais en réalité très compliquée
  - Une requête DNS peut impliquer plusieurs serveurs de noms répartis dans le monde entier
- Pourquoi pas de DNS centralisé ? Un seul serveur contiendrait toutes les correspondances requises par les applications de l'Internet
  - Dimension de l'Internet : trop de correspondances à gérer, nombre de requêtes au serveur trop important
  - Tolérance aux pannes : si le serveur DNS tombe, tout l'Internet aussi !
  - Volume de trafic impossible à supporter par un seul serveur
  - Délais de réponse : il faut faire en sorte que la réponse soit la plus proche possible du demandeur
  - Problème lié à la maintenance et aux mises à jour perpétuelles de la base

## Un système distribué

- Aucun serveur ne peut connaître toutes les correspondances nom <--> adresse IP
  - Si un serveur ne connaît pas une correspondance, il interroge un autre serveur jusqu'à atteindre le serveur détenant l'information souhaitée
- Trois types de serveur DNS
  - **Les serveurs de noms locaux** : c'est au serveur local que les applications clientes envoient toutes leurs requêtes
  - **Les serveurs de noms racine** : si un serveur local n'a pas la réponse, il transmet la requête à un serveur racine ; un serveur de noms racine connaît au moins les serveurs de source autorisée du premier niveau (.fr., ...)
  - **Les serveurs de noms de source autorisée** : un serveur de source autorisée contient les informations "officielles" de sa zone DNS ; il a autorité sur sa zone

## Un système distribué

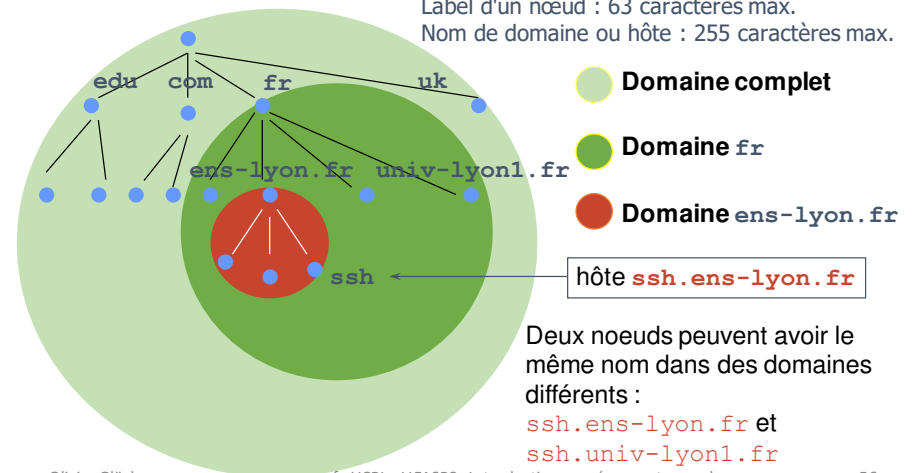


## Qu'est-ce qu'un domaine DNS ?

Un domaine est un sous-arbre entier de l'espace de nommage

Label d'un nœud : 63 caractères max.

Nom de domaine ou hôte : 255 caractères max.

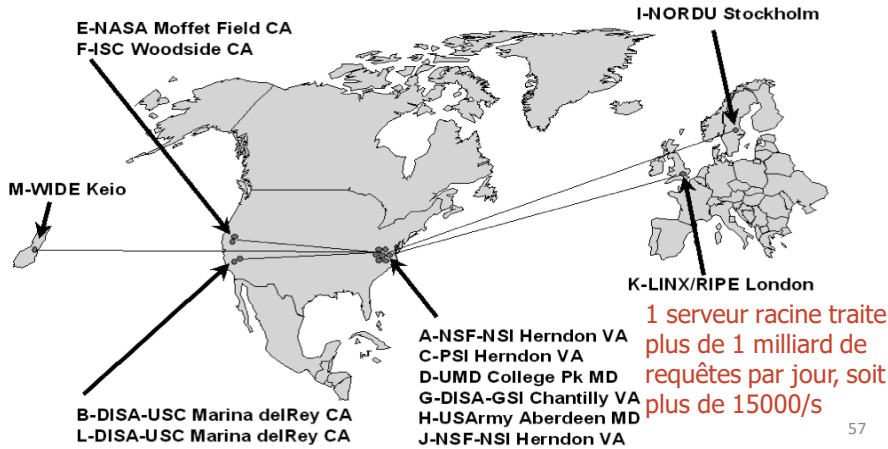


# Les serveurs racine

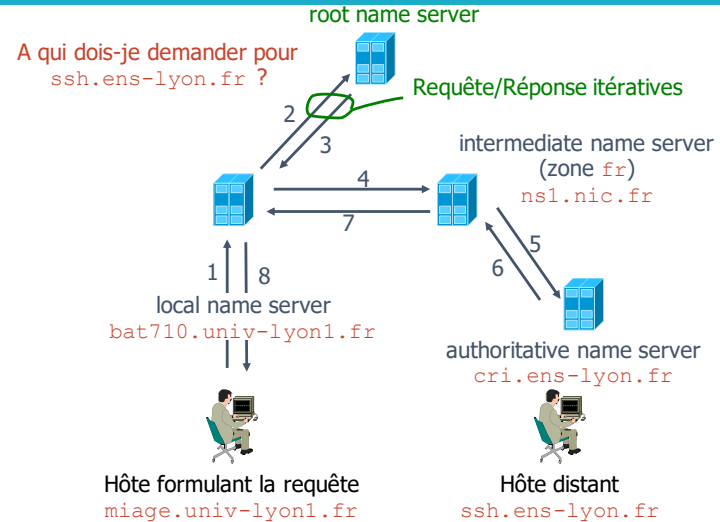
## DNS Root Servers

Designation, Responsibility, and Locations

1 primaire et 12 secondaires



# Principe d'une résolution de nom



# Les messages DNS [RFC 1034, 1035]

```

xterm
ogluck@lima:~$ host -a ssh.ens-lyon.fr
Trying "ssh.ens-lyon.fr"
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 11431
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;ssh.ens-lyon.fr.          IN      ANY     TTL  Classe  Type
;; ANSWER SECTION:
ssh.ens-lyon.fr.         7200   IN      CNAME   fulmar.ens-lyon.fr.
;; AUTHORITY SECTION:
ens-lyon.fr.            7200   IN      NS      cri.ens-lyon.fr.
ens-lyon.fr.            7200   IN      NS      imag.imag.fr.
ens-lyon.fr.            7200   IN      NS      ens.ens-lyon.fr.
;; ADDITIONAL SECTION:
cri.ens-lyon.fr.        7200   IN      A       140.77.1.32
imag.imag.fr.           172857 IN      A       129.88.30.1
ens.ens-lyon.fr.        7200   IN      A       140.77.1.183

Received 162 bytes from 140.77.1.32#53 in 0 ms
ogluck@lima:~$
    
```

12 octets d'en-tête

TTL Classe Type Valeur

Serveur primaire

Serveurs secondaires

# Les enregistrements stockés par les serveurs

- **Type=A** (val=1) : sert à décrire une correspondance  
Nom=nom d'hôte (canonique), Value=@IPv4
- **Type=AAAA** (val=28, RFC 1886) : idem mais adresse IPv6  
Nom=nom d'hôte, Value=@IPv6
- **Type=PTR** (val=12) : sert à la résolution inverse  
Nom=un nom de la zone arpa, Value=nom canonique (valeur pointée)
- **Type=NS** (val=2) : sert à associer un nom de domaine à un serveur de noms de source autorisée  
Nom=domaine, Value=nom du serveur de noms
- **Type=CNAME** (val=5) : sert à définir un alias pour un hôte  
Nom=un alias, Value=nom canonique (le vrai nom)

## Les enregistrements stockés par les serveurs

- **Type=MX** (val=15) : alias réservés aux serveurs mail permettant d'associer plusieurs serveurs de mail avec différentes priorités à une même adresse (RFC 974)  
Nom=un alias, Value=nom canonique d'un serveur de mail
- **Type=SOA** (val=6) : sert à donner des infos sur la zone  
Nom=nom d'une zone, Value=informations sur la zone
- **Type=ANY** (val=255) : utilisé dans les requêtes pour indiquer n'importe quel type (\*)
- **Type=AXFR** (val=252) : utilisé dans les requêtes pour demander le transfert d'une zone entière (mise à jour d'un serveur secondaire...)
- **Type=HINFO** (val=13) : sert à indiquer les CPU et OS du serveur interrogé

Olivier Glück

L1 Math-info UCBL - LIFASR2 : Introduction aux réseaux et au web

61

## Les enregistrements stockés par les serveurs

### Exemples :

```
ssh.ens-lyon.fr.      CNAME  fulmar.ens-lyon.fr.
ens-lyon.fr.         NS     cri.ens-lyon.fr.
ens-lyon.fr.         NS     ens.ens-lyon.fr.
cri.ens-lyon.fr.     A      140.77.1.32
relaissmtplib.ens-lyon.fr. CNAME  pluvier.ens-lyon.fr.
ens-lyon.fr.         MX     20 pluvier.ens-lyon.fr.
ens-lyon.fr.         MX     30 pluvier2.ens-lyon.fr.
listes.ens-lyon.fr. MX     20 pluvier.ens-lyon.fr.
fulmar.ens-lyon.fr.  A      140.77.167.6
6.167.77.140.in-addr.arpa. PTR  fulmar.ens-lyon.fr
```

Olivier Glück

L1 Math-info UCBL - LIFASR2 : Introduction aux réseaux et au web

62

## La commande host

```
xterm
ogluck@lima:~$ host
Usage: host [-aCdLrTuv] [-c class] [-n] [-N ndots] [-t type] [-W time]
          [-R number] hostname [server]
-a is equivalent to -v -t *
-c specifies query class for non-IN data
-C compares SOA records on authoritative nameservers
-d is equivalent to -v
-l lists all hosts in a domain, using AXFR
-n Use the nibble form of IPv6 reverse lookup
-N changes the number of dots allowed before root lookup is done
-r disables recursive processing
-R specifies number of retries for UDP packets
-t specifies the query type
-T enables TCP/IP mode
-v enables verbose output
-w specifies to wait forever for a reply
-W specifies how long to wait for a reply
ogluck@lima:~$
```

```
ogluck@lima:~$ host ssh.ens-lyon.fr
ssh.ens-lyon.fr is an alias for fulmar.ens-lyon.fr.
fulmar.ens-lyon.fr has address 140.77.167.6
```

Olivier Glück

L1 Math-info UCBL - LIFASR2 : Introduction aux réseaux et au web

63

```
ogluck — ssh -X lifasr2.univ-lyon1.fr -l olivier.gluck — 81x28
olivier.gluck@lifasr2:~$ host -a etu.univ-lyon1.fr
Trying "etu.univ-lyon1.fr"
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 8179
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 7

;; QUESTION SECTION:
;etu.univ-lyon1.fr.                IN      ANY

;; ANSWER SECTION:
etu.univ-lyon1.fr.                432000 IN      A      134.214.126.72
etu.univ-lyon1.fr.                3600   IN      MX     5 smtpbv.univ-lyon1.fr.

;; AUTHORITY SECTION:
univ-lyon1.fr.                    432000 IN      NS     dnsi.univ-lyon1.fr.
univ-lyon1.fr.                    432000 IN      NS     dns2.univ-lyon1.fr.
univ-lyon1.fr.                    432000 IN      NS     dns.univ-lyon1.fr.

;; ADDITIONAL SECTION:
smtpbv.univ-lyon1.fr.             1800   IN      A      134.214.126.92
dns.univ-lyon1.fr.                432000 IN      A      134.214.100.6
dns.univ-lyon1.fr.                432000 IN      AAAA   2001:660:5001:100::6
dns2.univ-lyon1.fr.               432000 IN      A      134.214.100.245
dns2.univ-lyon1.fr.               1800   IN      AAAA   2001:660:5001:100::245
dnsi.univ-lyon1.fr.               432000 IN      A      134.214.100.9
dnsi.univ-lyon1.fr.               432000 IN      AAAA   2001:660:5001:100::9

Received 278 bytes from 10.10.10.10#53 in 2 ms
olivier.gluck@lifasr2:~$
```

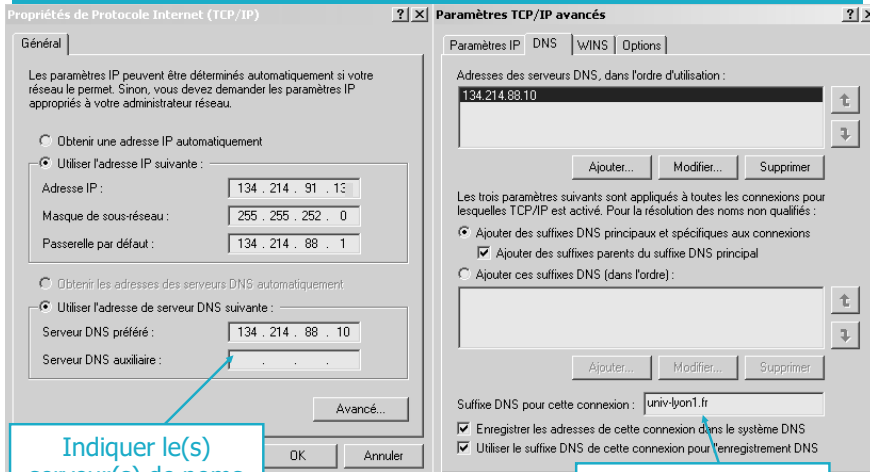
Olivier Glück

L1 Math-info UCBL - LIFASR2 : Introduction aux réseaux et au web

64



# Configuration d'un poste de travail



Indiquer le(s) serveur(s) de noms locaux

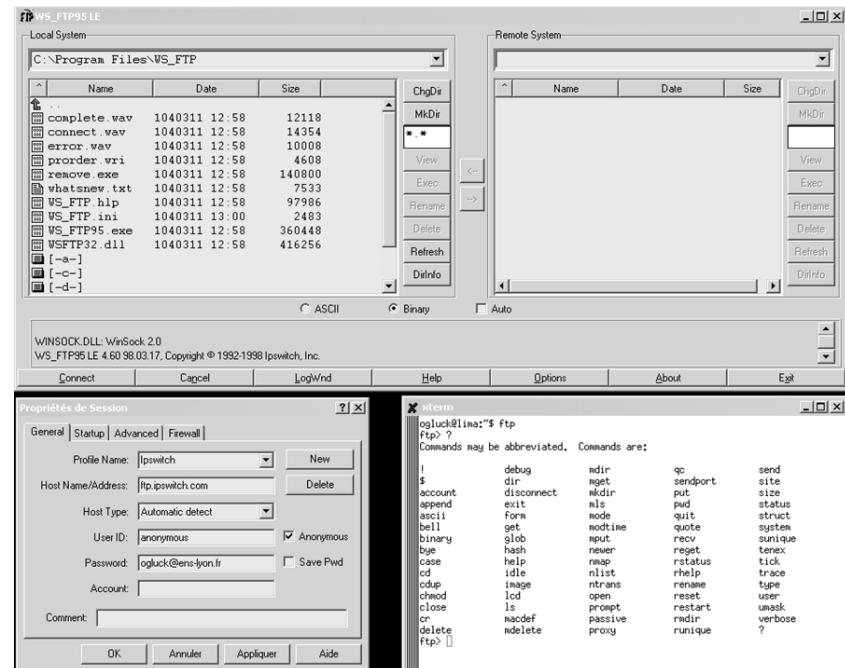
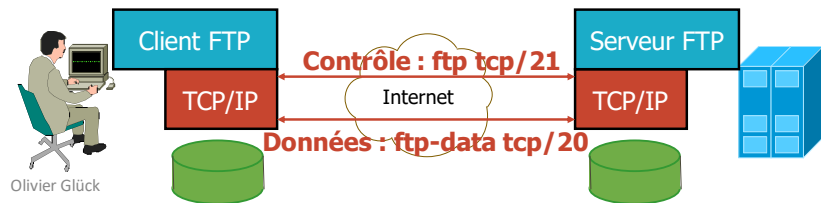
Suffixe DNS principal pour cette connexion

# Les autres applications (FTP, NFS, LDAP...)

Le transfert de fichiers (FTP)  
L'accès aux fichiers distants (NFS, SMB)  
LDAP : un annuaire fédérateur

# Le transfert de fichiers

- Copie intégrale d'un fichier d'un système de fichiers vers un autre en environnement hétérogène
  - copie de fichiers à distance : **r**cp, **s**cp
  - protocole de transfert de fichiers avec accès aux systèmes de fichiers local et distant : **f**tp, **t**ftp, **s**ftp
- Ne pas confondre avec les protocoles d'accès aux fichiers distants : NFS (RPC), SMB (Microsoft)
- Le serveur FTP maintient un "état" : répertoires courants local et distant, username



## Requêtes du protocole FTP

### **RETR** <filename>

Déclanche la transmission par le serveur du fichier <filename> sur le canal de données.

### **STOR** <filename>

Déclanche la réception d'un fichier qui sera enregistré sur le disque sous le nom <filename>. Si un fichier avec le même nom existe déjà il est remplacé par un nouveau avec les données transmises.

### **APPE** <filename>

Déclanche la réception d'un fichier qui sera enregistré sur le disque sous le nom <filename>. Si un fichier avec le même nom existe déjà, les nouvelles données lui sont concaténées.

### **REST** <offset>

Redémarrage en cas d'échec d'un transfert précédent. L'offset précise le numéro du dernier octet reçu.

**ABOR** : abandon d'un transfert en cours.

## Requêtes du protocole FTP

**PWD** : impression du répertoire courant.

**LIST** : catalogue du répertoire courant (canal donnée).

**NLIST** : catalogue succinct (canal donnée).

**CWD** <repname> : changement de répertoire courant pour <repname>.

**MKD** <repname> : création du nouveau répertoire <repname>.

**RMD** <repname> : suppression du répertoire <repname>.

**DELE** <filename> : suppression du fichier <filename>.

**RNFR** <filename1> : définit le nom actuel d'un fichier à renommer.

**RNTO** <filename2> : définit le nouveau nom d'un fichier à renommer.

**STAT** : status courant de la session FTP.

**STAT** <repname> : équivalent à LIST mais réponse sur le canal de contrôle.

**HELP** : affiche l'aide sur les opérations du site.

**NOOP** : no operation.

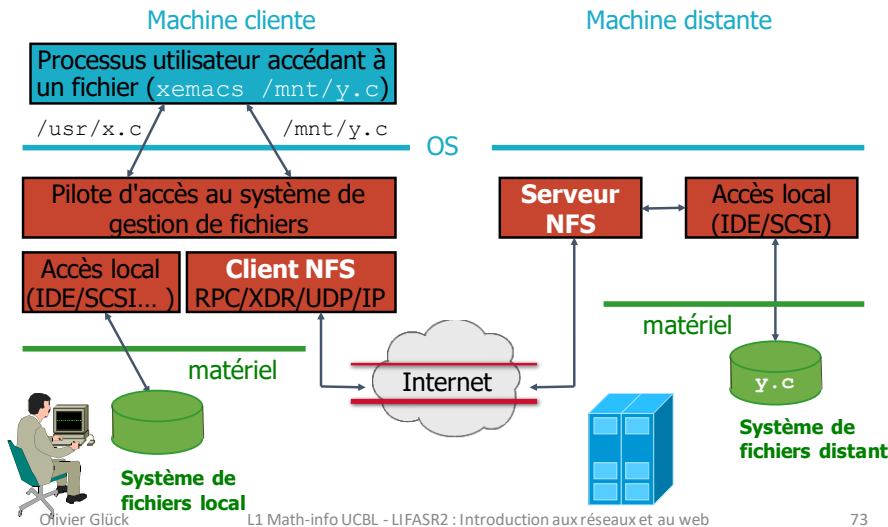
## Exemples de réponses FTP

- 125 Data connection already open
- 150 Opening BINARY mode data connection
- 200 Command successful
- 214 Help message
- 220 lima.cri2000.ens-lyon.fr FTP server (Version 6.4/OpenBSD/Linux-ftp-0.17) ready
- 226 Transfer complete
- 230 User ogluck logged in
- 331 Passwd required for ogluck
- 425 Can't open data connection
- 452 Error writing file
- 500 Command not understood
- 550 No files found

## L'accès aux fichiers distants

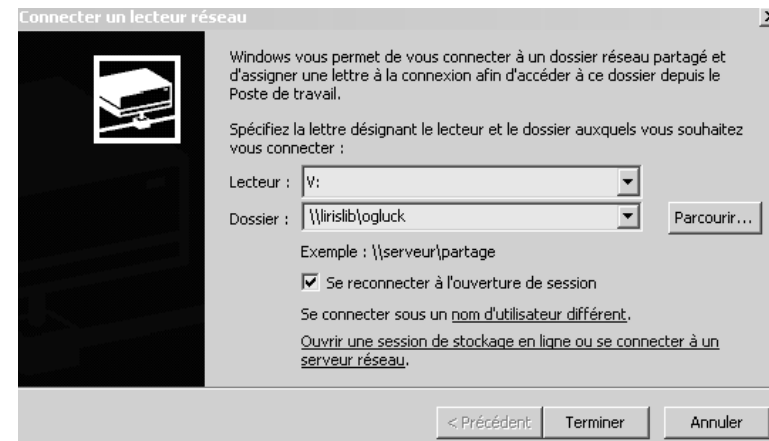
- Différences avec le transfert de fichiers
  - L'accès aux fichiers distants est complètement transparent pour l'utilisateur
  - Tout se passe comme si le système de fichiers distant était local
  - L'utilisateur peut éditer le fichier, le modifier, ... ; les modifications seront répercutées sur le système de fichiers distant
- Les deux principaux protocoles
  - NFS : *Network File System* (Unix/Sun-RPC)
  - SMB : *Server Message Block* (issu du monde Microsoft)

## NFS : principe de fonctionnement



## SMB : Server Message Block

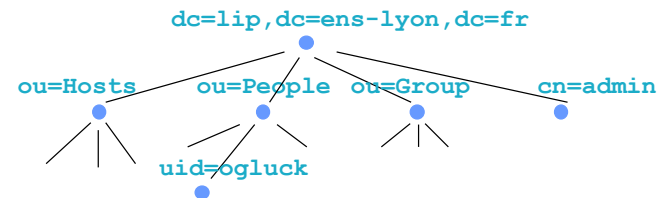
- Protocole de Microsoft et Intel permettant le partage de ressources (disques, imprimantes...) à travers un réseau (1987)



## LDAP : un annuaire fédérateur

- Permettre la fusion de multiples BD dans un unique annuaire informatique
  - base Microsoft Excel du personnel administratif
  - base Microsoft Access du personnel enseignant
  - base Microsoft Excel des numéros de téléphone
  - base /etc/passwd des comptes Unix des utilisateurs
  - base /etc/aliases (ou Sympa) de listes de Mail
  - base Samba des utilisateurs Windows
  - autres bases MySQL, Oracle, maps NIS,...
- Comment envoyer un mail à l'ensemble du personnel administratif sachant que l'administrateur système recevra uniquement une liste de (Nom, Prénom) ?

## LDAP : les objets stockés



```
dn: uid=ogluck,ou=People,dc=lip,dc=ens-lyon,dc=fr
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
uid: ogluck
uidNumber: 44132
gidNumber: 200
homeDirectory: /home/ogluck
cn: Olivier GLUCK
loginShell: /bin/bash
```

# Les URLs LDAP [RFC 1959]

The screenshot shows a Netscape browser window displaying LDAP search results for the domain 'lip'. The search criteria is 'uid, cn?sub?(cn=Olivier\*)'. The results list several users: Olivier AUMAGE, Olivier BEAUMONT, Olivier BODINI, Olivier RIFFAULT, and Olivier GLUCK. An 'Address Book' window is also open, showing a search for 'lip' and a list of contacts.

L'annuaire LDAP permet la mise à jour du carnet d'adresses

ldap://lip-ldap-master.ens-lyon.fr:389/dc=lip,dc=ens-lyon,dc=fr?uid,cn?sub?(cn=Olivier\*)

# LDAP : liens avec les autres applications

