

Les serveurs *UE 103b*

Guillaume Burel

guillaume.burel@ens-lyon.org

http://www.loria.fr/~burel/empty_cours.html



Transparents réalisés principalement par Olivier Christmann

[Les grandes lignes]

- Le modèle client-serveur
- Les principaux serveurs
- Un peu d'adressage...
- Exemple 1 : le serveur web
- Exemple 2 : le serveur mail

[Le modèle client-serveur 1/8]

C'est comme au Mac(Quick)-drive...

- L'équipier [le serveur] attend votre commande
- Vous [le client] passez votre commande
- L'équipier [le serveur] traite votre commande et éventuellement d'autres commandes (si votre hamburger n'est pas encore prêt)
- L'équipier [le serveur] vous donne votre commande (à vous, [le client])
- Vous [le client] partez (après avoir payé...)
- L'équipier [le serveur] attend se remet en attente

[Le modèle client-serveur 2/8]

Le serveur et... le(s) client(s)

- Le **serveur** fournit un **service** (*a priori*, vous ne savez pas faire de hamburger...)
- Le **client** émet des demandes de service, appelées **requêtes**, en suivant un **protocole**

L'idée

- Utiliser les ressources de machines dédiées à des tâches bien particulières
- Permettre à plusieurs machines d'utiliser ces ressources distantes
- Structurer et centraliser les ressources
- Gagner en souplesse (par exemple pour les *MàJ*)
- Améliorer la sécurité

[Le modèle client-serveur 3/8]

Vocabulaire

- **Client** : processus demandant l'exécution d'une opération par envoi d'un message contenant le descriptif de l'opération à exécuter, et attendant la réponse de cette opération par message retour
- **Serveur** : processus accomplissant une opération sur demande d'un client, et lui transmettant le résultat
- **Requête** : message transmis d'un client à un serveur décrivant l'opération à exécuter
- **Réponse** : message transmis par un serveur à un client suite à l'exécution d'une opération, contenant le résultat de l'opération
- **Protocole** : spécification des échanges réalisables entre le(s) client(s) et le serveur

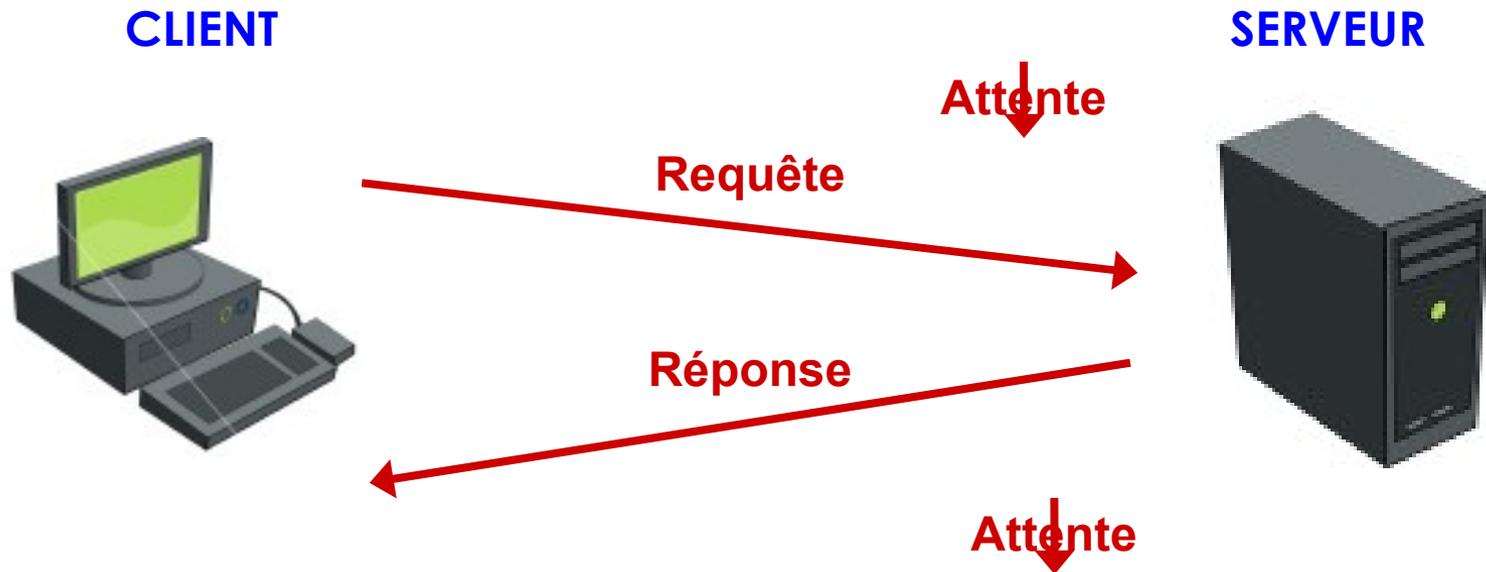
[Le modèle client-serveur 4/8]

Quelques propriétés

- **Partage des ressources, concurrence** : un serveur peut traiter plusieurs clients en même temps → un contrôle des ressources est donc nécessaire
- **Transparence** : le client accède à une ressource sans en connaître la localisation
- **Hétérogénéité** : indépendance des plates-formes matérielle et logicielle
- **Adaptabilité** : on peut modifier le serveur sans modifier le client (si le protocole reste le même). La réciproque est vraie → séparation présentation / couche métier
- **Modularité** : il est possible d'ajouter et de retirer des clients

Le modèle client-serveur 5/8

L'échange



→ Dialogue 2 à 2 entre processus

[Le modèle client-serveur 6/8]

Le client

- **Initie** le contact (parle en premier)
- **Émet** des requêtes à un serveur
- **Reçoit** des réponses du serveur

Le serveur

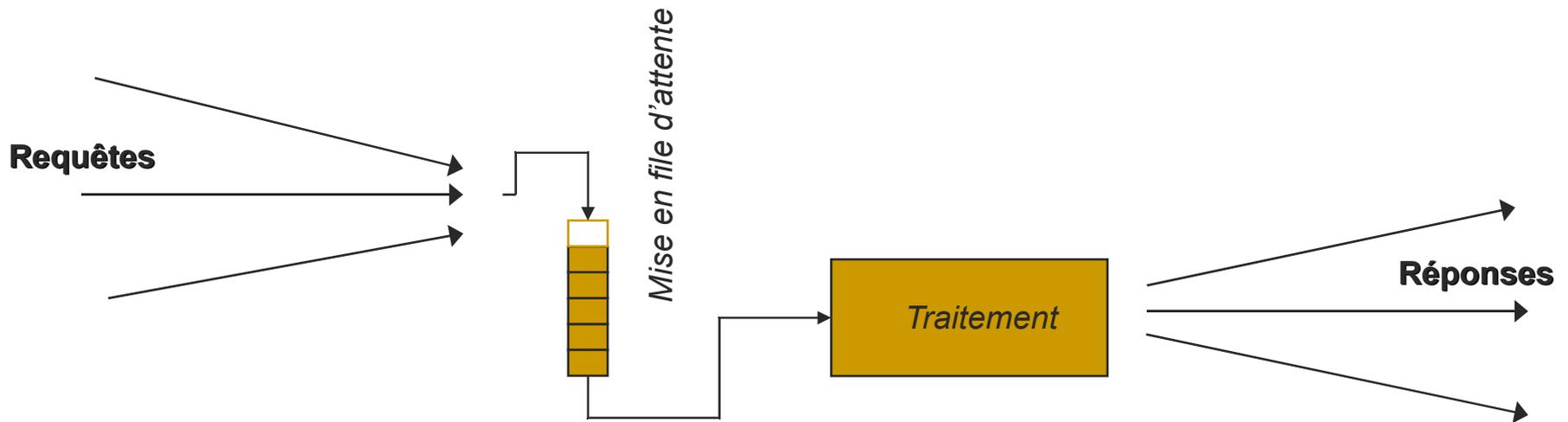
- **Attend** en permanence les requêtes de clients
- **Reçoit** des requêtes d'un ou plusieurs clients
- **Traite** les requêtes
- **Émet** les réponses au(x) client(s) correspondant(s)

+ un protocole applicatif

- Définit les échanges entre le client et le serveur

[Le modèle client-serveur 7/8]

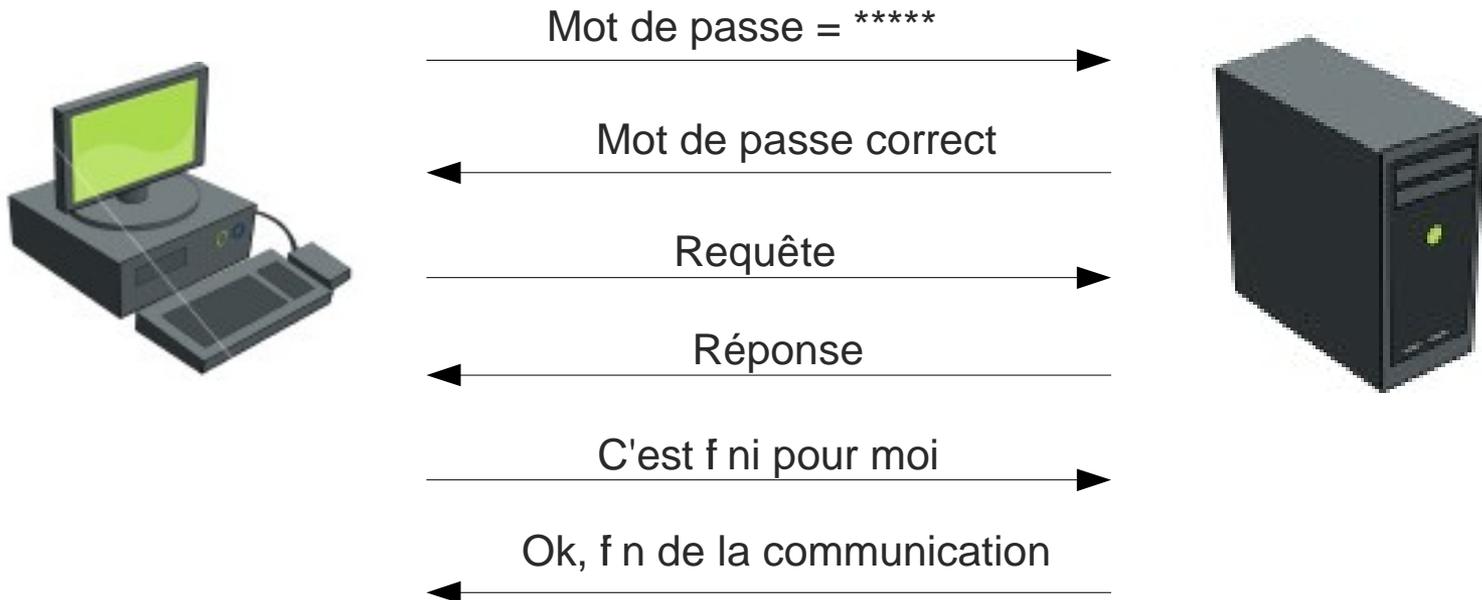
Vu du côté serveur



Le modèle client-serveur 8/8

■ Protocole

spécifie comment client et serveur communiquent



→ session de communication, gestion des erreurs, sécurité

[Les principaux serveurs 1/13]

Le serveur web

- Vous l'utilisez (indirectement) tous les jours sans le savoir (ou pas...)
- Aussi appelé **serveur http** par analogie avec le protocole du même nom...
- Désigne soit l'ordinateur, soit le logiciel en lui-même
- Les navigateurs web (*Internet Explorer, Mozilla Firefox*) jouent le rôle de clients
- Gère l'accès aux données (les pages web des sites hébergés + contenu)

D'après l'adresse (par ex. *www.google.fr*) que vous indiquez, le client se connecte au serveur, demande le code **HTML** de la page, et le reçoit en retour.

Le client (navigateur web) interprète ce code et affiche la page

Les principaux serveurs 2/13

Le serveur web

Les serveurs HTTP les plus utilisés sont :

- ***Apache HTTP Server*** de *Apache Software Foundation*
- ***Internet Information Services (IIS)*** de *Microsoft*
- ***Java System Web Server*** de *Sun Microsystems*

Le plus courant est ***Apache*** (voir UE 203b – semestre 2)



[Les principaux serveurs 3/13]

Le serveur DNS

- Vous l'utilisez tous les jours sans le savoir
- **DNS = Domain Name System** (système de nom de domaines)
- Permet d'établir la correspondance entre **numéro IP** (*Internet Protocol* → adresse d'un ordinateur sur le réseau) et un nom de site Internet
- Cela fonctionne dans les deux sens (***résolution inverse***)
- Maillage de serveurs DNS (chaque serveur possède une information partielle)

Par exemple, vous souhaitez accéder à ***www.google.fr***

Votre client se connecte au serveur DNS et envoie l'adresse comme **requête**

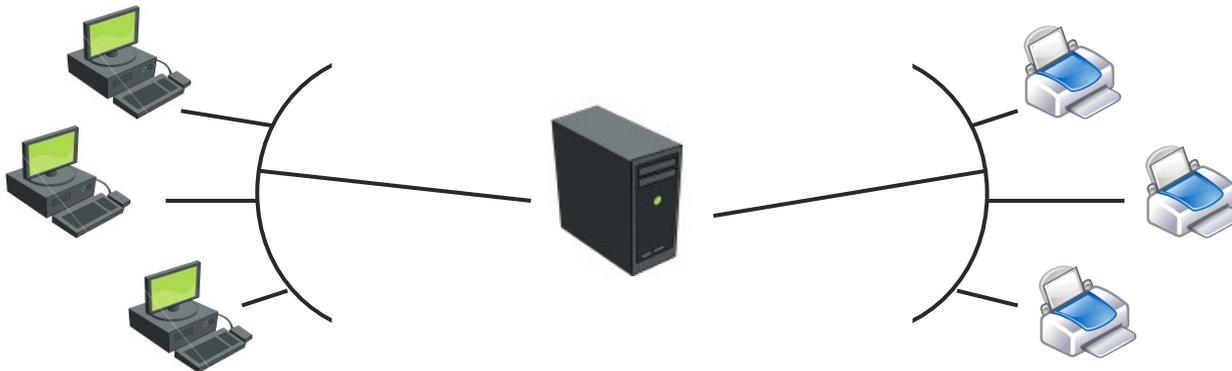
Il reçoit comme **réponse** l'adresse IP **209.85.135.47**

Les principaux serveurs 4/13

Le serveur d'impression

- Problématique : **partager** une imprimante entre plusieurs (dizaines/centaines) de d'ordinateurs
- L'imprimante ne peut satisfaire toutes les requêtes en temps réel → **mémorisation** des travaux à réaliser
- Gestion des ressources, **files d'attente**

Le plus courant sous Unix/Linux est **CUPS** (*Common **U**nix **P**rinting **S**ystem*)



[Les principaux serveurs 5/13]

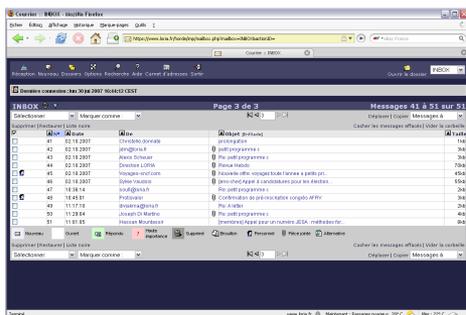
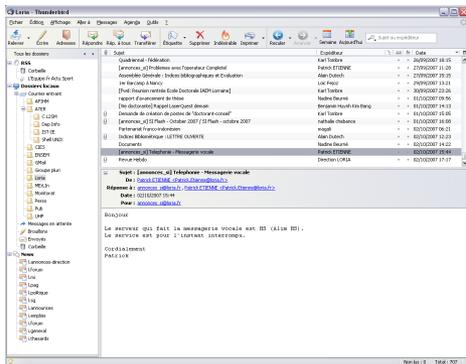
Le serveur de fichiers

- Permettre le partage des données à travers un réseau
- Avoir accès à ses données quelque soit l'ordinateur sur lequel on se connecte → entreprise, éducation
- Couramment, indique l'ordinateur à partir duquel on peut **monter** (mount) un lecteur disque ou un répertoire pour qu'il apparaisse directement sur l'ordinateur hôte
- L'accès aux données se fait par différents protocoles : WebDAV, SMB, **NFS**, AppleTalk, CIFS, ...

Les principaux serveurs 6/13

Le serveur mail

- Vous l'utilisez tous les jours (normalement)
- Gestion des mails (envoi/réception + stockage)
- Plusieurs protocoles
 - Émission : **SMTP**
 - Réception : **IMAP, POP**
- Émission et réception peuvent être dissociés physiquement
- 2 catégories de clients :
 - Client de messagerie (*Microsoft Outlook, Mozilla Thunderbird*) → logiciel
 - Webmail → interface web (*IMP/Horde, GMail, YahooMail*)



[Les principaux serveurs 7/13]

Le serveur mail

MTA = Mail Transfer Agent

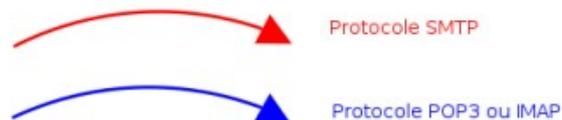
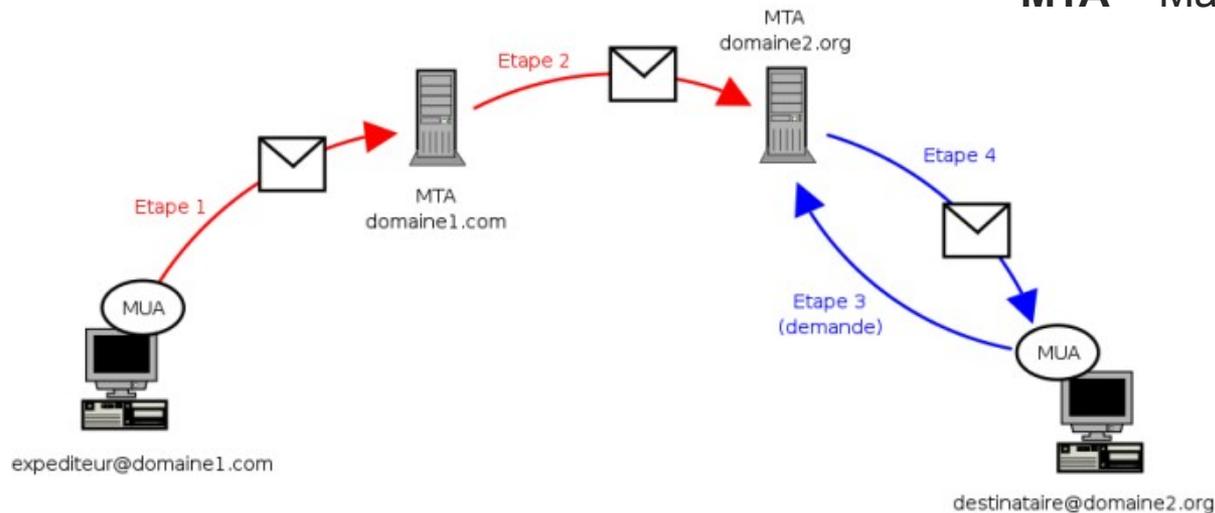


Image wikipedia

[Les principaux serveurs 8/13]

Le serveur mail

Logiciels de serveurs mail courants :

- **Sendmail** (*open source*)
- **Postfix** (*open source*)
- **Microsoft Exchange Server / Lotus Notes** → messagerie interne (entreprise principalement) avec fonctionnalités étendues (travail collaboratif)

Les principaux serveurs 9/13

Le serveur de nouvelles (Usenet)

- Possibilité de récupérer et de poster des articles, accessibles à tous
- Articles parfois échangés entre serveurs de nouvelles
- Regroupés par thèmes (*newsgroup*) de façon hiérarchique :
 - sci.math
 - sci.chem
 - comp.os.linux
 - fr.petites-annonces.telephones.mobiles
- Articles récupérés par des lecteurs de nouvelles (*Pan*) ou par les clients de messagerie (*Outlook*, *Thunderbird*)
- Rendu plus ou moins obsolète par les forums de discussion gérés par des serveurs webs

[Les principaux serveurs 10/13]

Le serveur d'applications

- Permet de centraliser les applications sur un poste
- Les utilisateurs accèdent aux applications via le réseau
- Les clients sont les ordinateurs des utilisateurs : des ordinateurs « normaux » ou des **clients légers** (ressources matérielles limitées → processeur peu rapide, pas de disque dur)
- Par extension, désigne également les applications basées sur le web



→ **Voir UE 203d – semestre 2**

Les principaux serveurs 11/13

Le serveur de base de données

- Permet la centralisation et l'administration des données
- Ensemble DONNEES + LOGICIEL + MATERIEL
- Les **données** sont regroupées dans des **tables** :

Enregistrements (= lignes)

Nom	Prénom	Sexe	Adresse	Ville	Code postal
Durand	Pierre	M	31 rue des champs	Uriage	38410
Chose	Stéphanie	F	2 place Stanislas	Nancy	54000
Trombe	Jean	M	18 cours de la libération	Grenoble	38001
etc.					

Champs (= colonnes)

Les principaux serveurs 12/13

Le serveur de base de données

- Le Logiciel (SGBD) permet d'accéder à la base de données (interrogation, ajout, mise à jour de données)
- Langage normalisé de manipulation de BDD : **SQL** (*Structured Query Language*)
- Beaucoup d'éditeurs :
 - Microsoft **Access** (bureautique) et **SQL Server** (grands systèmes)
 - MySQL (logiciel libre)
 - **Oracle** par Oracle Corporation
 - IBM **DB2**
 - ...

[Les principaux serveurs 13/13]

Le serveur de base de données

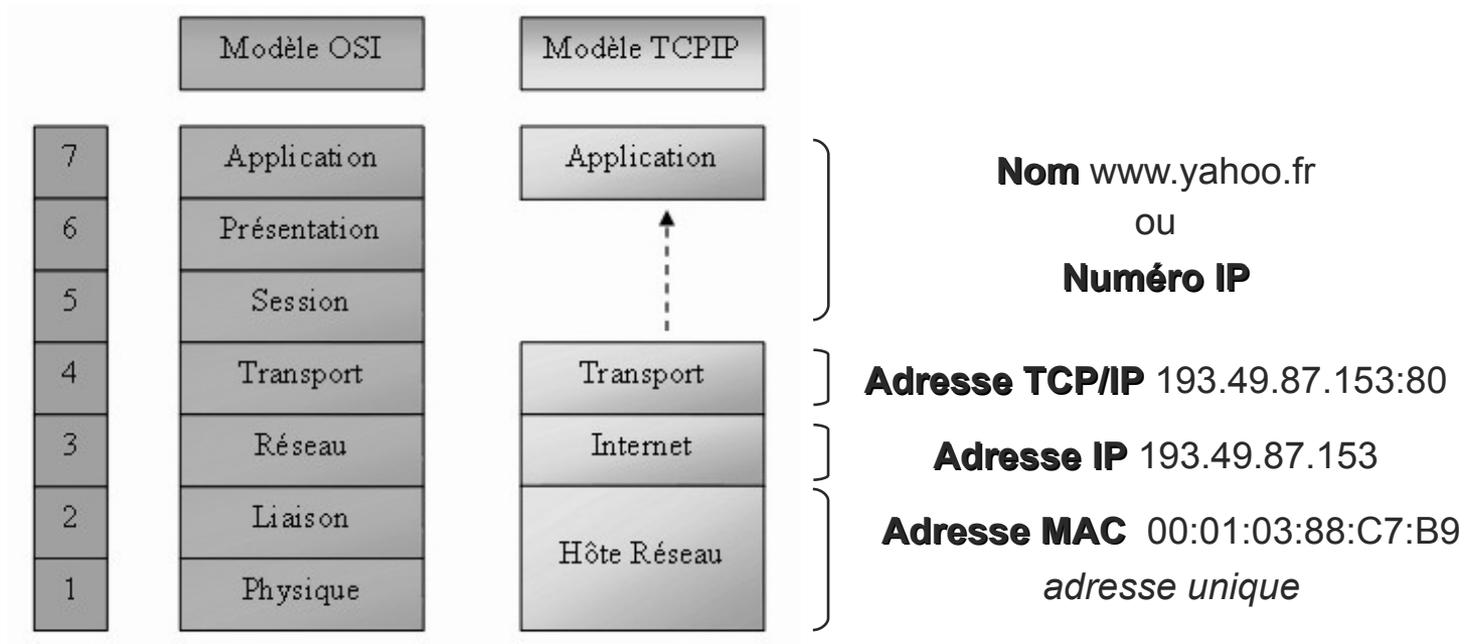
Le matériel doit permettre :

- Un accès rapide aux données (disque dur rapide)
- Un accès multiple (évidemment)
- Un accès fiable et sécurisé (comme les autres types de serveurs)

→ Voir UE 203c – semestre 2

[Un peu d'adressage... 1/13]

Le modèle OSI et TCP/IP



[Un peu d'adressage... 2/13]

De l'URL à l'adresse IP → **Domain Name System**

- Approche client/serveur
- Organisation hiérarchique et mondiale des données
- Mémorisation des informations recueillies (*cache*)

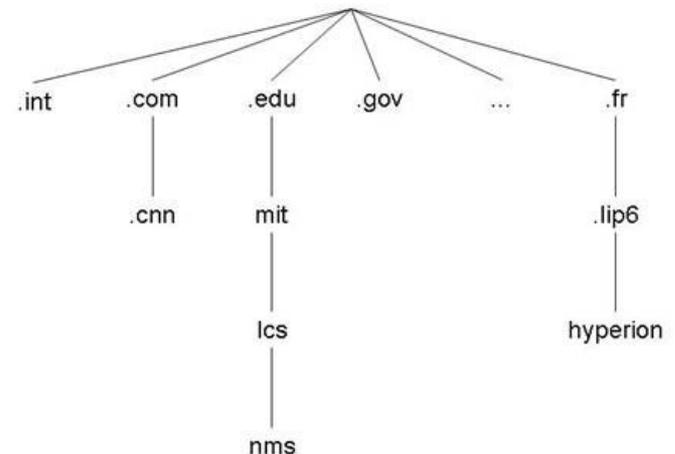
→ Espace de noms mondial, cohérent, indépendant des protocoles et des systèmes de communication

[Un peu d'adressage... 3/13]

Arborescence

- **Root** (racine)
- **Top Level Domains (TLD)** :
 - Historiquement : *.com*, *.edu*, *.gov*, *.mil*, *.net*, *.org*, *.int*
 - Organisations nationales : *.fr*, *.uk*, *.de*, *.it*, *.es* ...
 - Nouveaux : *.eu*, *.mobi*, *.travel*, *.cat*, ...
 - ...
- **Secondary Level Domain**, ...

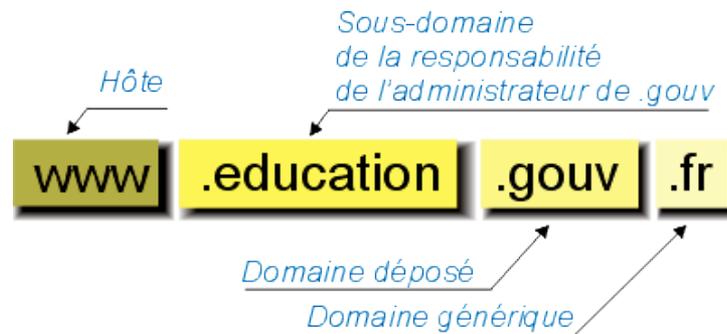
Label : chaîne commençant par une lettre, contenant uniquement des lettres, des chiffres et des tirets, d'au plus 63 caractères



[Un peu d'adressage... 4/13]

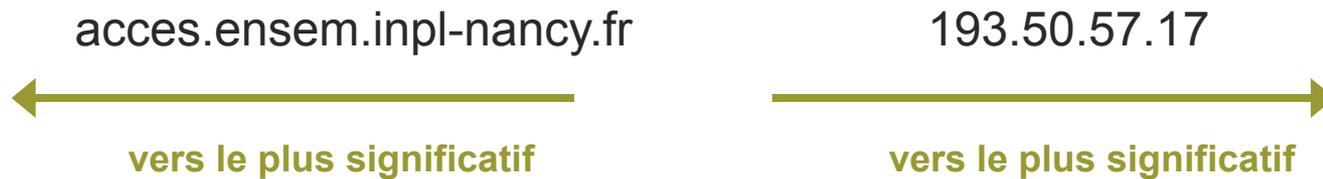
Spécificités

- 127 niveaux maximum, longueur maximum 255 caractères
- **Nommage relatif** : *ensem.inpl-nancy* relatif à *fr*
- Sous domaine : domaine inclus dans un autre
- Unicité des noms de domaine : à chaque niveau par le domaine englobant
- Sous-domaine = délégation
- Exemple :

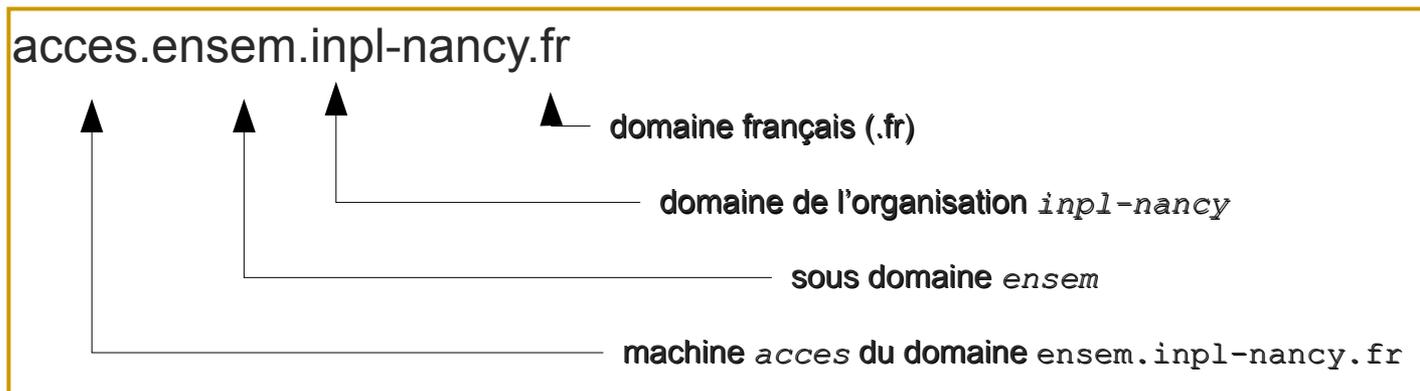


[Un peu d'adressage... 5/13]

Lecture des noms de domaines



Un autre exemple



[Un peu d'adressage... 6/13]

Et les adresses IP ?

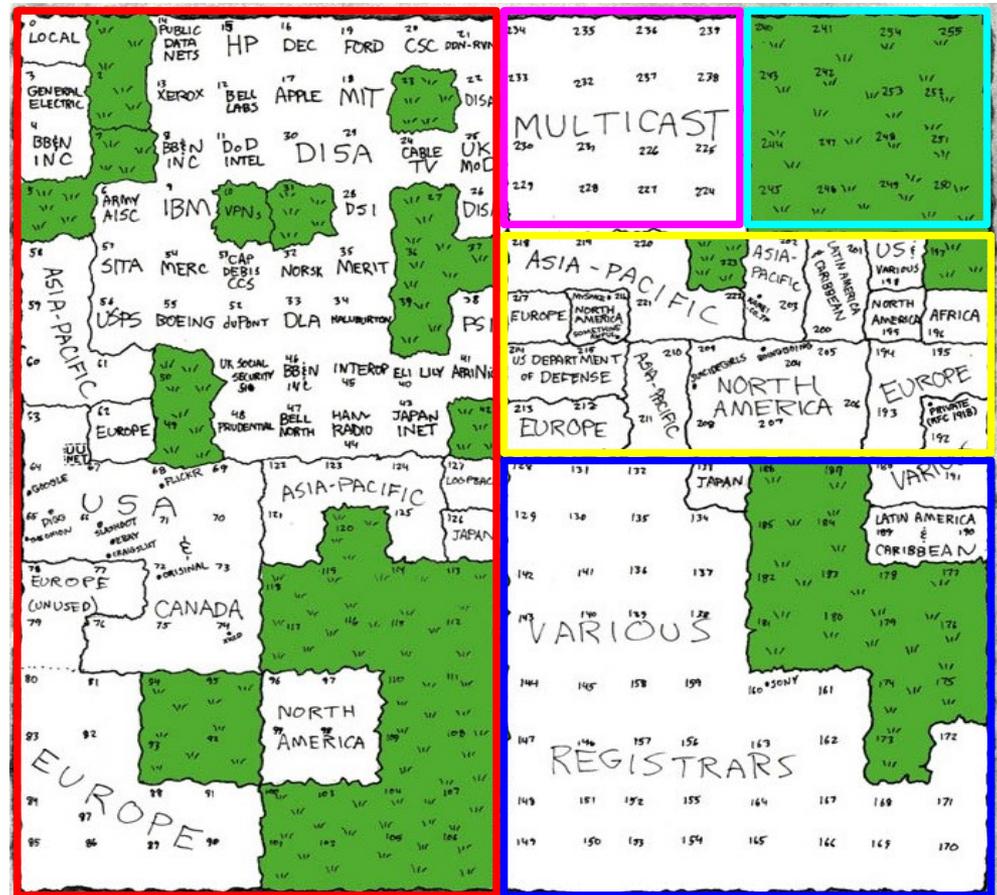
- Nombre codé sur 4 octets (un octet = 8 bits → de 0 à 255)
- Adresse habituellement écrite sous forme décimale W.X.Y.Z avec $W, X, Y, Z \in [0...255]$
- La partie réseau de l'adresse est en tête, la partie hôte est en queue → 2 adresses en 1...
- On a donc principalement 3 classes :
 - 1 octet réseau – 3 octets hôte (classe A)
 - 2 octets réseau – 2 octets hôte (classe B)
 - 3 octets réseau – 1 octet hôte (classe C)
- Comme à la poste !

Un peu d'adressage... 7/13

Récapitulatif des classes

Classe Nombre de réseaux/machines

A	<p><i>1.x.y.z à 127.x.y.z</i> 127 réseaux 16 777 216 machines (2^{24})</p>
B	<p><i>128.0.x.y à 191.255.x.y</i> 16 384 réseaux (2^{14}) 65536 machines (2^{16})</p>
C	<p><i>192.0.0.z à 223.255.255.z</i> 2 097 152 réseaux (2^{21}) 256 machines (2^8)</p>
D	<p><i>224.0.0.0 à 239.255.255.255</i></p>
E	<p><i>240.0.0.0 à 247.255.255.255</i></p>



Source de la carte : xkcd

Licence Creative Commons Attribution-NonCommercial 2.5

[Un peu d'adressage... 8/13]

Adresses particulières

- 0.0.0.0 : route par défaut dans les tables de routage
- 255.255.255.255 : adresse de diffusion à toutes les machines du réseau
- @réseau.255 : idem à 255.255.255.255
- 127.0.0.1 : boucle locale (la machine elle-même)
- 224.0.0.0 – 224.0.0.255 : réservées pour le multicast entre les équipements d'interconnexion

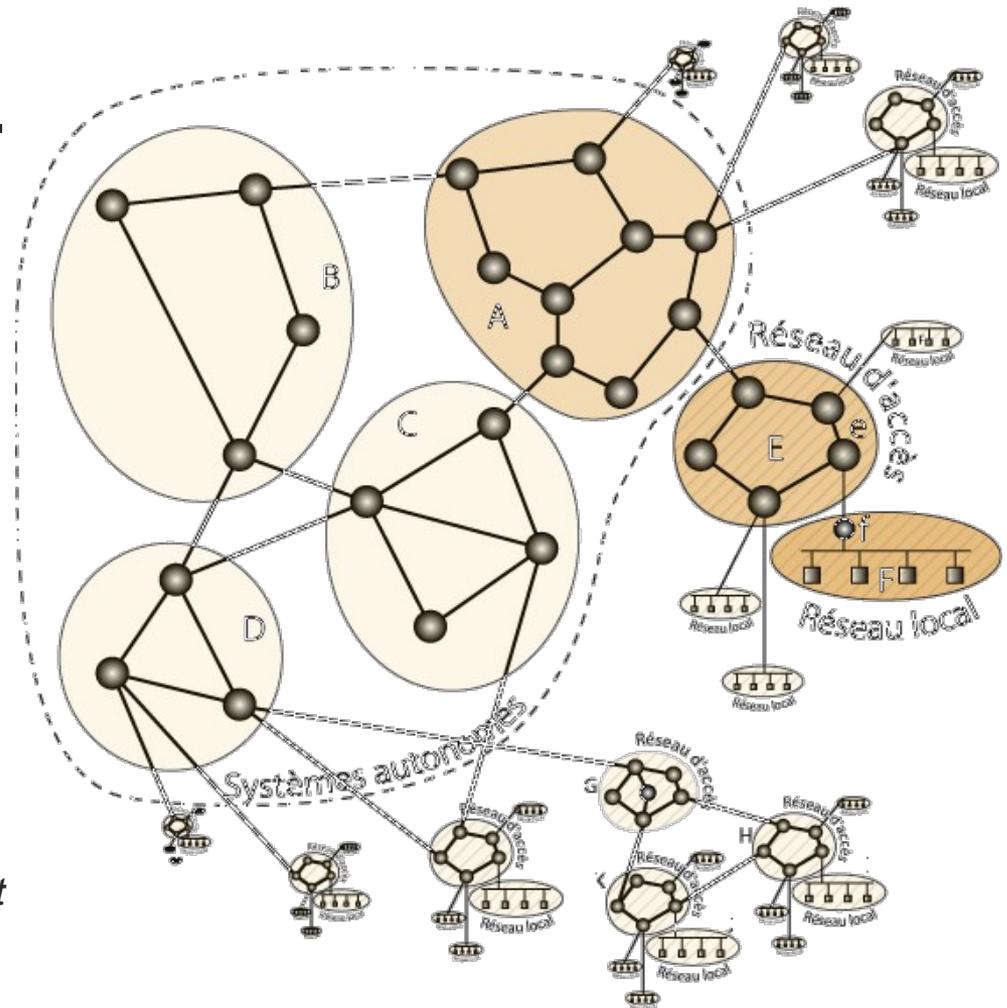
Un peu d'adressage... 9/13

Un réseau, en vrai...

Transport de paquets



routage



<http://interstices.info/upload/internet>

[Un peu d'adressage... 10/13]

Adresse de réseau, adresse d'interface

- Une adresse de réseau à tous les bits d'hôte positionnés à 0
- Par exemple pour une classe A :
 - 10.0.0.0 → adresse de réseau
 - 10.0.1.0 → adresse d'interface
- Par exemple pour une classe B :
 - 128.17.0.0 → adresse de réseau
 - 128.17.1.0 → adresse d'interface
- Par exemple pour une classe C :
 - 192.168.3.0. → adresse de réseau
 - 192.168.3.125 → adresse d'interface

[Un peu d'adressage... 11/13]

Les sous-réseaux

- À partir de l'adresse d'un réseau, faire un découpage permettant d'avoir plusieurs réseaux interconnectés, du point de vue local (réseaux logiques)
- → Gestion simplifiée des échanges et meilleure sécurisation des différentes classes d'utilisateurs (isolation)
- De manière pratique : on réserve 1 ou plusieurs bits d'interface, et on les interprète localement comme faisant partie des bits de réseau
- Mais comment distinguer @ réseau et @ hôte ? **masque**

[Un peu d'adressage... 12/13]

Les sous-réseaux : un exemple

- Prenons un réseau de classe C, par exemple **192.168.1.0**

Nombre de sous-réseaux	Nbre d'hôtes par réseau	Masque de réseau	
2	126	255.255.255.128	(11111111.11111111.11111111.10000000)
4	62	255.255.255.192	(11111111.11111111.11111111.11000000)
8	30	255.255.255.224	(11111111.11111111.11111111.11100000)
16	14	255.255.255.240	(11111111.11111111.11111111.11110000)
32	6	255.255.255.248	(11111111.11111111.11111111.11111000)
64	2	255.255.255.252	(11111111.11111111.11111111.11111100)

- Et au LORIA, par exemple ?
 - 2 sous-réseaux :
 - **recherche** 152.81.0.0 / 255.255.240.0 : les machines dont le n° IP est 152.81.x.y avec $x \in [1..15]$ et $y \in [0..255]$ → 4094 machines
 - **kiosque** 152.81.144.0 / 255.255.240.0 : les machines dont le n° IP est 152.81.x.y avec $x \in [144..159]$ et $y \in [0..255]$ → 4094 machines

[Un peu d'adressage... 13/13]

Masques et sous-réseaux

■ exemple : 192.44.77.79 / 255.255.255.192

192.44.77.79 → 11000000 . 00101100 . 01001101 . 01001111

255.255.255.192 → 11111111 . 11111111 . 11111111 . 11000000 *& logique*

@ sous-réseau → 11000000 . 00101100 . 10001101 . 01000000 = **192.44.77.64**

nombre de machines **62**

@ machine **001111** (15 en décimal)

On note parfois 192.44.77.79 / 26 (nombre de 1 dans le masque)

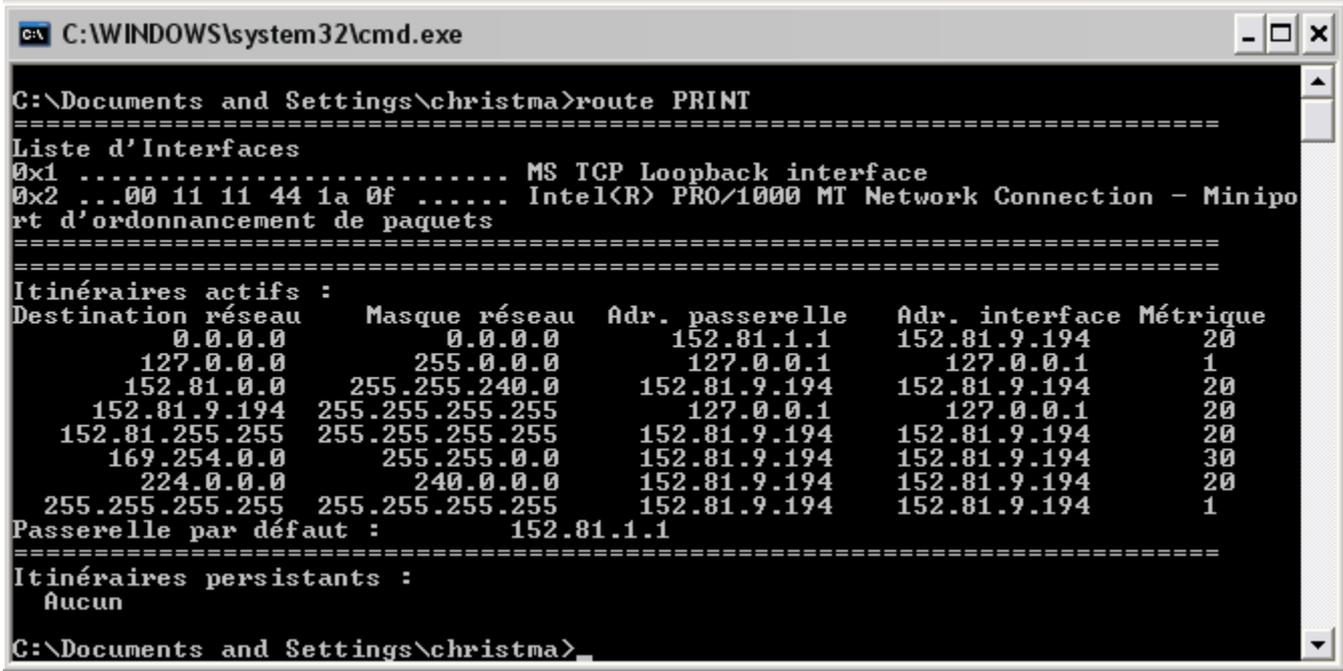
[Le routage... 1/2]

Les tables de routage

- Une station doit posséder une table de routage afin de savoir à qui elle doit envoyer un paquet IP
- Elles doivent représenter en permanence l'état du réseau
- Elles sont maintenues statiquement ou dynamiquement
- Chaque ligne comporte un coût
 - Algorithme de routage : chemin de coût minimal
- Routeur : machine avec plusieurs cartes réseaux dont chacune est reliée sur un réseau différent
- Table de routage = tableau avec correspondances
- Si réseau n'apparaît pas dans la table, utilisation ***d'une passerelle par défaut***

Le routage... 2/2

La commande route PRINT



```
C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\christma>route PRINT
=====
Liste d'Interfaces
0x1 ..... MS TCP Loopback interface
0x2 ...00 11 11 44 1a 0f ..... Intel(R) PRO/1000 MT Network Connection - Minipo
rt d'ordonnancement de paquets
=====

Itinéraires actifs :
Destination réseau      Masque réseau      Adr. passerelle    Adr. interface  Métrique
0.0.0.0                 0.0.0.0            152.81.1.1         152.81.9.194    20
127.0.0.0               255.0.0.0         127.0.0.1         127.0.0.1      1
152.81.0.0              255.255.240.0     152.81.9.194     152.81.9.194   20
152.81.9.194           255.255.255.255   127.0.0.1         127.0.0.1      20
152.81.255.255         255.255.255.255   152.81.9.194     152.81.9.194   20
169.254.0.0            255.255.0.0       152.81.9.194     152.81.9.194   30
224.0.0.0               240.0.0.0         152.81.9.194     152.81.9.194   20
255.255.255.255       255.255.255.255   152.81.9.194     152.81.9.194   1
Passerelle par défaut : 152.81.1.1
=====

Itinéraires persistants :
Aucun

C:\Documents and Settings\christma>
```

Sous Unix : netstat -r

[De IPv4 à IPv6... 1/2]

Pourquoi IPv6 ?

- La taille d'internet double tous les 12 mois
- 2 problèmes à résoudre :
 - Épuisement des adresses IP
 - Explosion des tables de routage
- Le nouveau protocole doit permettre :
 - Espace d'adressage plus grand
 - Routage plus efficace

[De IPv4 à IPv6... 2/2]

Caractéristiques ?

- Adresse plus longue : **128 bits** (16 octets au lieu de 4)
 - Plus de 340.10^{36} équipements (contre 4.10^9 pour IPv4)
 - L'adresse MAC peut en faire partie
 - Exemple : 1fff:0000:0a88:85a3:0000:0000:ac1f:8001
- 3 types d'adresses :
 - unicast : interface unique
 - multicast : groupe d'interfaces
 - anycast : un seul membre d'un groupe
- Compatible avec @ IPv4
- 15 % allouées initialement / 85 % réservées pour la croissance

[Pour finir... 1/2]

Petit exercice n°1

- Soit l'interface *129.33.145.131 / 255.255.255.240*
 - Classe de cette interface ?
→ **128.0.x.y < @ <191.255.x.y donc classe B 129.33**
 - Adresse de sous-réseau ?
→ **10000001 . 00100001 . 10010001 . 10000011 &
11111111 . 11111111 . 11111111 . 11110000
10000001 . 00100001 . 10010001 . 10000000
= 129.33.145.128**
 - Combien de machines ?
→ **4 bits pour l'@ de la machine, soit 16 possibilités = 14 interfaces**

[Pour finir... 2/2]

Petit exercice n°2

- Soit eth0 193.31.12.3 / 255.255.255.224, et les plans suivants :
 - 193.31.12.67 / 255.255.255.128
 - 193.31.12.120 / 255.255.255.192
 - 193.31.13.193 / 255.255.192.0
- Pour chaque plan, liste des @IP pouvant communiquer avec eth0
 - **machines visibles depuis eth0** : 193.31.12.1 à 193.31.12.30
 - **plan 2** : 193.31.12.1 à 193.31.12.126
 - **plan 3** : 193.31.12.65 à 193.31.12.126
 - **plan 4** : 193.31.0.1 à 193.31.63.254

[Le serveur mail... 1/13]

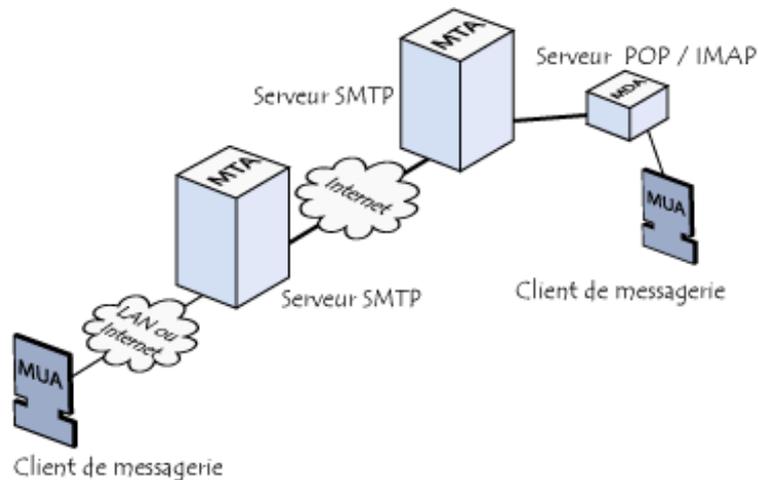
Protocoles

- Plusieurs protocoles :
 - Consultation (serveur entrant) : **Post Office Protocol 3** ou **Internet Message Access Protocol**
 - Envoi (serveur sortant) : **Simple Mail Transfert Protocol**
- Ces services peuvent être implantés sur des machines distinctes
- Font partie des standards de l'internet, et sont de ce fait enregistrés en tant que **RFC** (Request for Comments)

[Le serveur mail... 2/13]

Un peu de vocabulaire

- **Mail User Agent** : outil de consultation et d'envoi de mail du client
- **Mail Transfer Agent** : outil d'acheminement des mail de l'émetteur au destinataire
- **Mail Delivery Agent** : outil s'occupant de la distribution des mails dans les « boîtes aux lettres »



[Le serveur mail... 3/13]

POP3

- RFC 1939, protocole ancien (mais toujours largement utilisé) permettant à l'origine une utilisation hors ligne :
 - Connexion de la machine cliente au serveur
 - Transfert du courrier du serveur vers la machine cliente
 - Suppression des messages sur le serveur
 - Fin de la connexion
 - Puis, à intervalles réguliers, la machine cliente interroge le serveur pour « savoir » s'il y a de nouveaux messages
- Utilisation principalement avec des logiciels spécialisés (Thunderbird, Outlook, Seamonkey, Eudora,...)
- Il est possible également de taper les différentes commandes de manière interactive

POP3, principales commandes

- **USER** : votre nom de compte
- **PASS** : votre mot de passe
- **LIST** : liste les messages présents sur le serveur (pour chaque message, un n° et la taille en octets)
- **RETR** : télécharge le message indiqué
- **DELE** : efface le message indiqué
- **TOP** : affiche les premières lignes du message indiqué
- **QUIT** : termine la session en cours
- **APOP** : permet une authentification sécurisée

[Le serveur mail... 5/13]

POP3, une session

```
S: <en attente de la connexion TCP sur le port 110>
C: <ouverture de la connexion>
S: +OK POP3 server ready <1896.697170952@dbc.mtview.ca.us>
C: APOP mrose c4c9334bac560ecc979e58001b3e22fb
S: +OK mrose's maildrop has 2 messages (320 octets)
C: STAT
S: +OK 2 320
C: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
C: RETR 1
S: +OK 120 octets
S: <le serveur POP3 envoie le message 1>
S: .
C: DELE 1
S: +OK message 1 deleted
C: RETR 2
S: +OK 200 octets
S: <le serveur POP3 envoie le message 2>
S: .
C: DELE 2
S: +OK message 2 deleted
C: QUIT
S: +OK dewey POP3 server signing off (maildrop empty)
C: <close connection>
S: <en attente de la prochaine connexion>
```

[Le serveur mail... 6/13]

POP3, inconvénients

- Par défaut, le client récupère les messages les uns à la suite des autres : un message ne pouvant se télécharger (ou trop volumineux) bloque tous les autres...
- Les sauvegardes dépendent de l'utilisateur
- Le changement de client (logiciel) de messagerie peut s'avérer [très] complexe
- La consultation par **webmail** devient impossible après téléchargement

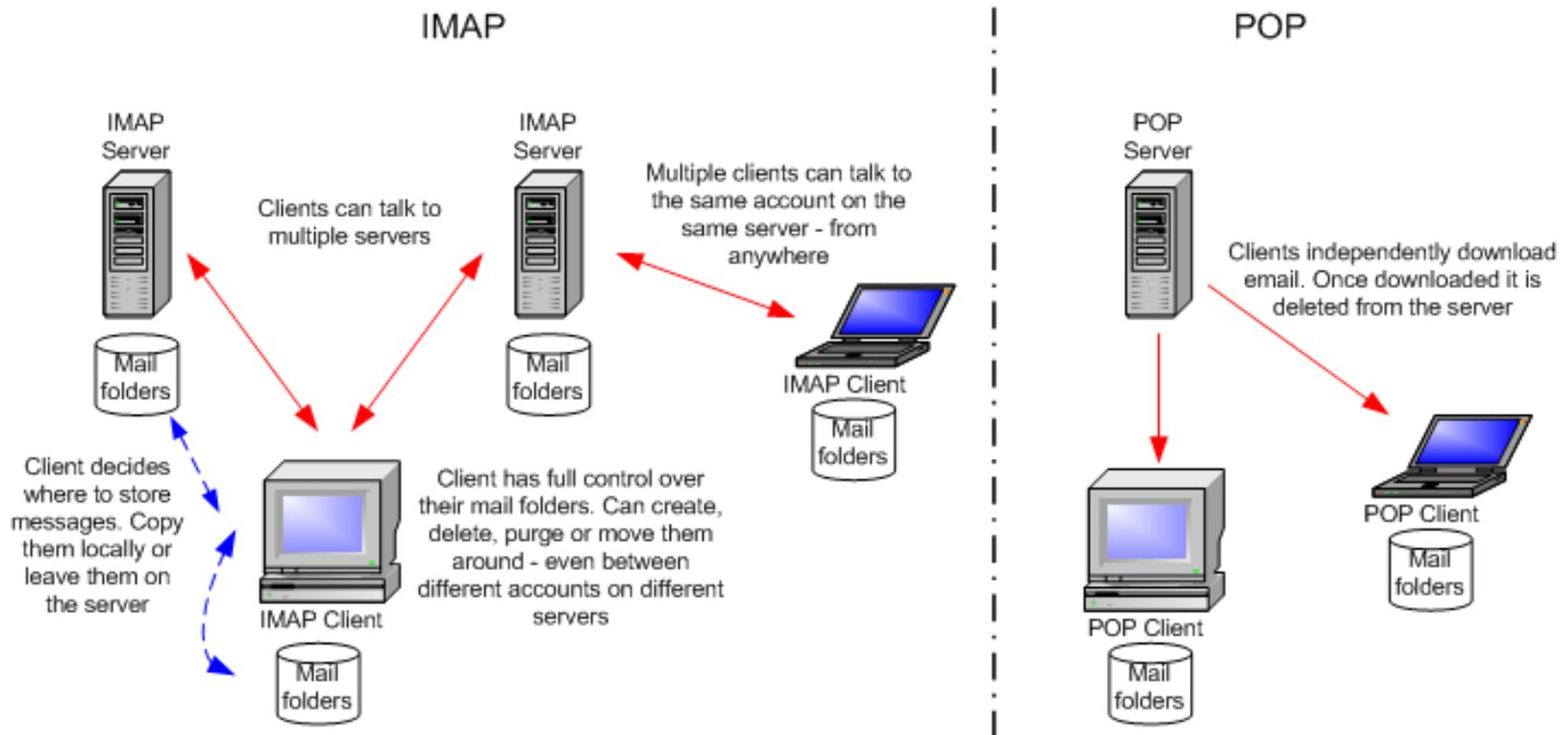
[Le serveur mail... 7/13]

IMAP

- RFC 3501, communication sur le port 143
- La liste des messages est également transmise au client (comme dans POP)
- Les messages sont alors gérés individuellement. On peut choisir :
 - quels messages restent sur le serveur
 - quels messages doivent être supprimés
 - quels messages doivent être téléchargés
- Gestion distante en « local »
- Inconvénients :
 - Temps de connexion
 - Utilisation des ressources serveurs

Le serveur mail... 8/13

Pour comparer



[Le serveur mail... 9/13]

SMTP

- RFC 2821, communication sur le port 25
- Pas d'authentification de l'expéditeur
- Gestion de la sécurité au niveau du serveur (MTA)
- Solution : **Extended SMTP**
 - Compatibilité avec SMTP
 - *HELO* → *EHLO*
 - Si le serveur supporte ESTMP, il annonce la liste des options qu'il supporte, sinon il répond *HELO*
- Dialogue avec le serveur en plusieurs étapes

[Le serveur mail... 10/13]

SMTP, l'échange

```
220 smtp.xxxx.xxxx SMTP Ready
HELO client
250 Hello client, pleased to meet you
MAIL FROM:<user@xxxx.xxxx>
250 <user@xxxx.xxxx> ... Sender ok
RCPT TO:<user2@yyyy.yyyy>
250 recipient ok.
DATA
354 Enter mail, end with "." on a line by itself
Subject: Test
From: user@xxxx.xxxx
To: user2@yyyy.yyyy

Ceci est un test pour un article de wikipédia portant sur SMTP.
.
250 Ok
QUIT
221 Closing connection
Connection closed by foreign host.
```

[Le serveur mail... 11/13]

Les aliases

- Adresse non basée sur le login
- Redirection interne sur le domaine, permettant, entre autres, d'avoir plusieurs comptes attachés à une boîte mail.
- Il devient possible d'avoir :
 - Un nom complet d'utilisateur
 - Une liste de diffusion
- C'est le serveur de mail qui redirige le message à l'adresse réelle

[Le serveur mail... 12/13]

Le SPAM

- **SPiced hAM** : l'histoire est trop longue à raconter
- Courrier non sollicité envoyé à plusieurs personnes (un peu comme les prospectus distribués dans les boîtes aux lettres)
- Les adresses sont récupérées via les News, les listes de diffusion, les pages web (analyse des champs **mailto**, recherche de texte avec le caractère @)
- Apparu avec l'explosion du nombre d'utilisateurs de l'Internet
- Commerce florissant : les fichiers d'utilisateurs sont revendus de sociétés en sociétés)
- Essaye de contourner les filtres :
vi@gr@ v|agra v i a g r a

Le SPAM – les solutions

- Filtrage au niveau personnel :
 - Manuellement
 - Via des logiciels dédiés
 - Via le logiciel de messagerie
- Filtrage au niveau d'un site
 - Liste noire des « spammeurs » reconnus
 - Refuser les adresses invalides
 - Refuser les adresses IP non valides des expéditeurs
- En amont :
 - Ne pas mettre son adresse mail « en clair » sur son site
 - Eviter de s'enregistrer sur n'importe quel site (et dans le meilleur des cas, avoir une adresse dédiée).

[Le serveur web... 1/17]

Le web...

- Architecture client-serveur
 - Le client web : **navigateur** ou *browser* (IE, Firefox, Opera,...)
 - Le serveur : machine qui attend les requêtes et envoie les réponses (Apache, par exemple)
- Accès à des documents liés entre eux, dispersés sur des milliers de machines
- Protocole utilisé : **HyperText Transfer Protocol**
- Simple système question/réponse
- Les demandes contiennent l'**Uniform Resource Locator** de la page à afficher
- Les réponses contiennent (entre autres) des documents au format **HyperText Markup Language**

[Le serveur web... 2/17]

I'URL

- Extension de la notion de nom de fichier sur un réseau
- But : retrouver et adresser tout objet sur le réseau
- Syntaxe : **protocole://adresse/fichier**
 - Protocole : HTTP, FTP, news,...
 - Adresse :
 - Spécifique au protocole
 - En général @ IP du serveur HTTP (ou le nom, qui sera résolu par appel au serveur DNS)
 - Fichier : nom du fichier à récupérer, facultatif (si omis, page par défaut : index.htm / index.html)
- **Exemples :**
 - `http://www.google.fr/`
 - `ftp://ftp.ciril.fr/pub/linux/mandrakelinux/official/iso/2008.0/xx.iso`

[Le serveur web... 3/17]

HTML

- C'est un **langage de description** (et non de programmation)
- C'est un langage qui évolue constamment
- Il permet de :
 - décrire la structure d'un document
 - d'y inclure des informations variées (textes, images, sons, animations)
 - d'établir des relations cohérentes grâce aux liens hypertextes
- Un document HTML est structuré
- La description d'un document HTML par l'utilisation de **balises** ou *tags* :
 - Une balise est délimitée par les signes `<` et `>`
 - la plupart du temps : balise ouvrante `<nom_balise>` et balise fermante `</nom_balise>`

[Le serveur web... 4/17]

HTML

```
<HTML>
<HEAD><TITLE>Un exemple simple de document HTML</TITLE>
<BODY>
<H1>Ceci est une section de premier niveau</H1>
<HR>
<B>Bienvenue dans le monde de HTML. </B>
Ceci est un paragraphe.<P>
Et ceci en est un second.<P>
</BODY>
</HTML>
```



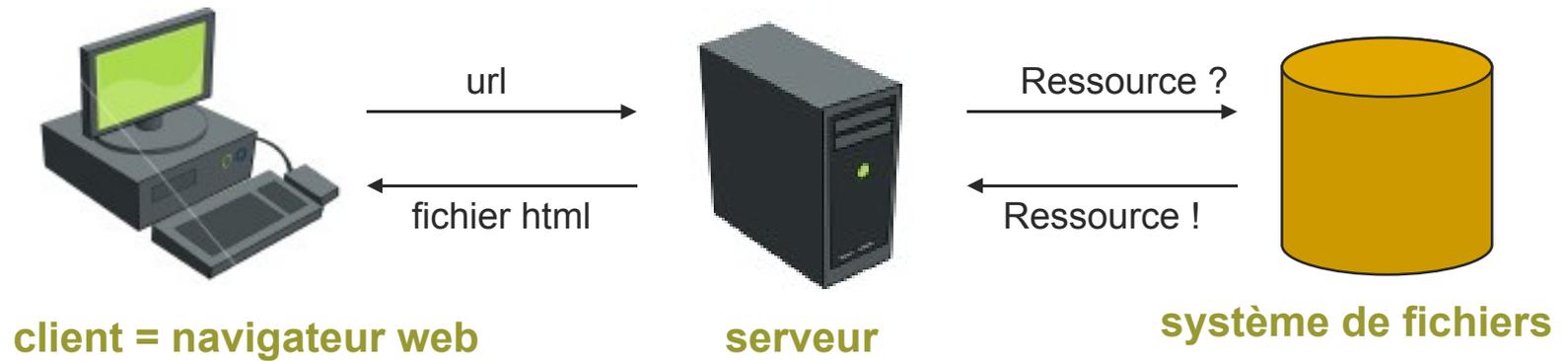
[Le serveur web... 5/17]

MIME

- On peut également inclure des types **Multipurpose Internet Mail Extensions** dans les pages web
 - Codages de caractères différents d'ASCII
 - Contenus non textuels : images, audio, videos, application
- Au début, pour les courriels, pour les autres types de codages, principalement latins (caractères accentués par exemple)
- MIME est extensible, on peut enregistrer de nouveaux types de contenus
- Voir <http://www.iana.org/assignments/media-types/> pour les différents types enregistrés par l'IANA

[Le serveur web... 6/17]

Principe de base



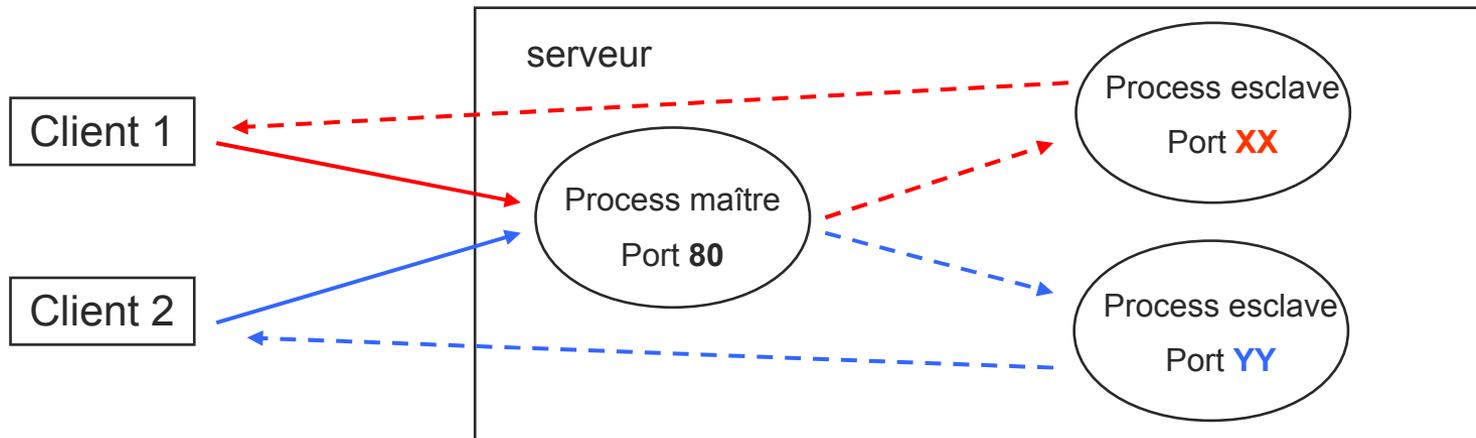
Un échange requête/réponse par document

+ un échange pour tout élément inclus (même site ou autre site)

[Le serveur web... 7/17]

Par exemple, un serveur Apache

- Écoute du serveur « maître » sur le port 80
- Lors d'une connexion, le serveur « maître » passe le relais un à serveur « esclave »
- Cela libère le port 80, et utilise donc un autre port pour l'échange avec le client (traitement de la requête et réponse)
- À la fin, la connexion est fermée



[Le serveur web... 8/17]

L'échange [au microscope]

- HTTP = protocole en mode « texte »
- **1.** Le client se connecte au serveur (adresse + n° de port 80)
- **2.** Il envoie une requête HTTP :
 - **GET** : ressource située à l'URL spécifiée
 - **HEAD** : entête de la ressource située à l'URL spécifiée
 - **POST** : envoi de données à l'URL spécifiée
 - **DELETE** : suppression de la ressource à l'URL spécifiée
 - **TRACE, PUT, OPTIONS, CONNECT**
- **GET** est la requête la plus utilisée, la requête ne contient pas de message

[Le serveur web... 9/17]

La requête *GET*

- La syntaxe est la suivante :
GET *ressource* **HTTP**/*version*
- La ressource est le nom de la page web à charger
- La version correspond à la version du protocole HTTP :
 - **0.9 : GET**. De nombreux inconvénients, liés à la limitation à un seul échange :
 - Une nouvelle connexion pour chaque document à télécharger, y compris pour les images contenus dans les pages web
 - Saturation du réseau / serveur
 - Pas de gestion de cache
 - **1.0 : HEAD, POST** + cache + entêtes + cookies / mais toujours les connexions multiples
 - **1.1 : OPTIONS, PUT, DELETE, TRACE, CONNECT** + connexion persistante

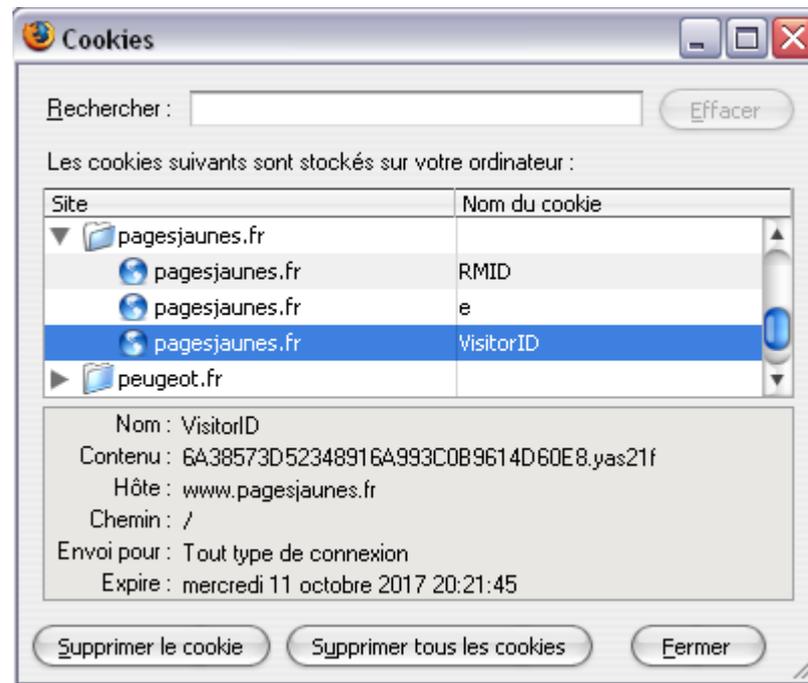
Petite parenthèse : les cookies (1)

- **Définition** : Élément d'information qui est transmis par le serveur au navigateur lorsque l'internaute visite un site Web, et qui peut être récupéré par ce serveur lors de visites subséquentes.
- Fichiers *texte* enregistrés sur l'ordinateur du client par le navigateur web
- Durée de vie variable
- Permet d'enregistrer des informations sur l'utilisateur, comme par exemple le parcours sur le site web, ou les produits consultés (pour un site *e-commerce*).
- Se trouvent dans l'entête HTTP des réponses :
Set-Cookie : NOM=VALEUR; domain=NOM_DE_DOMAINE; expires=DATE
- Se trouvent dans l'entête HTTP des requêtes :
Cookie : NOM1=VALEUR1; NOM2=VALEUR2;

[Le serveur web... 11/17]

Petite parenthèse : les cookies (2)

- Un serveur ne peut sauver plus de 20 cookies chez le client
- Taille maximale = 4 Ko



[Le serveur web... 12/17]

La réponse du serveur...

- Une fois que le serveur a reçu la requête, et si celle-ci est valide, il effectue le traitement.
- **3.** Le serveur envoie la réponse :
 - **une ligne de statut** (version du protocole)
 - **Un code de succès** (ou d'échec)
 - **Informations** (relatives au serveur et au message)
 - **Corps du message**
- **4.** Le serveur termine la connexion

Petite parenthèse : TelNet

- Protocole de type client-serveur, fonctionnant sur les réseaux de type TCP/IP
- Permet d'obtenir une communication bidirectionnelle entre deux machines distantes
- Les commandes sont exécutées sur la machine distante comme si elles étaient tapées sur la console du serveur
- Fonctionne sur la plupart des systèmes d'exploitation
- Utilisation du port 23
- Possibilité d'utilisation avec les serveurs web (protocole HTTP), les serveurs mail (protocoles SMTP, POP, ...), ...
- Inconvénient majeur : tout passe en clair sur le réseau !

Le serveur web... 14/17

Exemple de session

<pre>-bash-3.00\$ telnet www.loria.fr 80 Trying 152.81.144.22... Connected to www.loria.fr (152.81.144.22). Escape character is '^]'. GET /~christma/index.htm HTTP/1.0</pre>	Connexion
	Connexion acceptée
	Requête
<pre>HTTP/1.1 200 OK Date: Mon, 15 Oct 2007 17:32:26 GMT Server: Apache/2.0.61 (Unix) mod_ssl/2.0.61 DAV/2 PHP/4.4.6 Last-Modified: Mon, 30 May 2005 10:55:10 GMT ETag: "2c6140-7e3-b8b3df80" Accept-Ranges: bytes Content-Length: 2019 Connection: close Content-Type: text/html; charset=ISO-8859-1</pre>	Entête Réponse
<pre>..... <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> <html> <HEAD> <title>Olivier Christmann Home Page</title> </HEAD> </html></pre>	Contenu
<pre>Connection closed by foreign host.</pre>	Connexion terminée

[Le serveur web... 15/17]

Codes de retour

Renseigne sur le succès ou l'échec de la requête :

- **Information : 100 et 101** (demande de la suite de la requête, changement de protocole)
- **Succès : 200 à 206**
 - 200 / OK : requête traitée avec succès
 - 204 : requête correctement traitée, mais pas de contenu en retour
- **Redirection : 300 à 305 + 307**
 - 301 : changement d'adresse définitif
 - 302 : changement d'adresse temporaire
- **Erreurs du client : 400 à 417**
 - 400 : mauvaise requête
 - 401 : authentification nécessaire
 - 403 : interdit
 - **404 : non trouvé**
- **Erreurs du serveur : 500 à 505**
 - 500 : erreur interne du serveur
 - 503 : service indisponible

Le serveur web... 16/17

Type de documents

- MIME

audio/x-wav	Fichiers audio Wave	wav
image/gif	Images gif	man
image/jpeg	Images JPEG	jpg, jpeg, jpe
image/png	Images PNG	png
image/tiff	Images Tiff	tiff, tif
image/x-portable-bitmap	Fichiers Bitmap PBM	pbm
image/x-portable-graymap	Fichiers Graymap PBM	pgm
image/x-portable-pixmap	Fichiers Pixmap PBM	ppm
multipart/x-zip	Fichiers archive zip	zip
multipart/x-gzip	Fichiers archive GNU zip	gz, gzip
text/css	Feuille de style	css
text/csv	Fichiers texte avec séparation des valeurs	csv
text/html	Fichiers HTML	htm, html
text/plain	Fichiers texte sans mise en forme	txt, g, h, c, cc, hh, m, f90
text/richtext	Fichiers texte enrichis	rtx
text/rtf	Fichiers texte au format Rich Text Format	rtf
text/tab-separated-value	Fichiers texte avec séparation des valeurs	tsv
text/xml	Fichiers XML	xml
video/h264	Vidéos H.264	h264
video/dv	Vidéos au format DV	dv
video/mpeg	Vidéos MPEG	mpeg, mpg, mpe
video/quicktime	Vidéos QuickTime	qt, mov
video/msvideo	Vidéos Microsoft Windows	avi

[Le serveur web... 17/17]

Et la sécurité...

- Aucune garantie de sécurité : tout circule « *en clair* » sur le réseau avec HTTP
- Avec le *e-commerce*, ou pour les échanges *officiels*, il a fallu disposer de plus de garanties :
 - confidentialité
 - intégrité
 - authentification
- Association de **HTTP** et de **Secure Socket Layer** → **HTTPS**
- Utilisation du port 443
- Dans le navigateur :
 - apparition de **https** dans la barre d'adresse
 - apparition d'un cadenas fermé dans le bas de l'écran



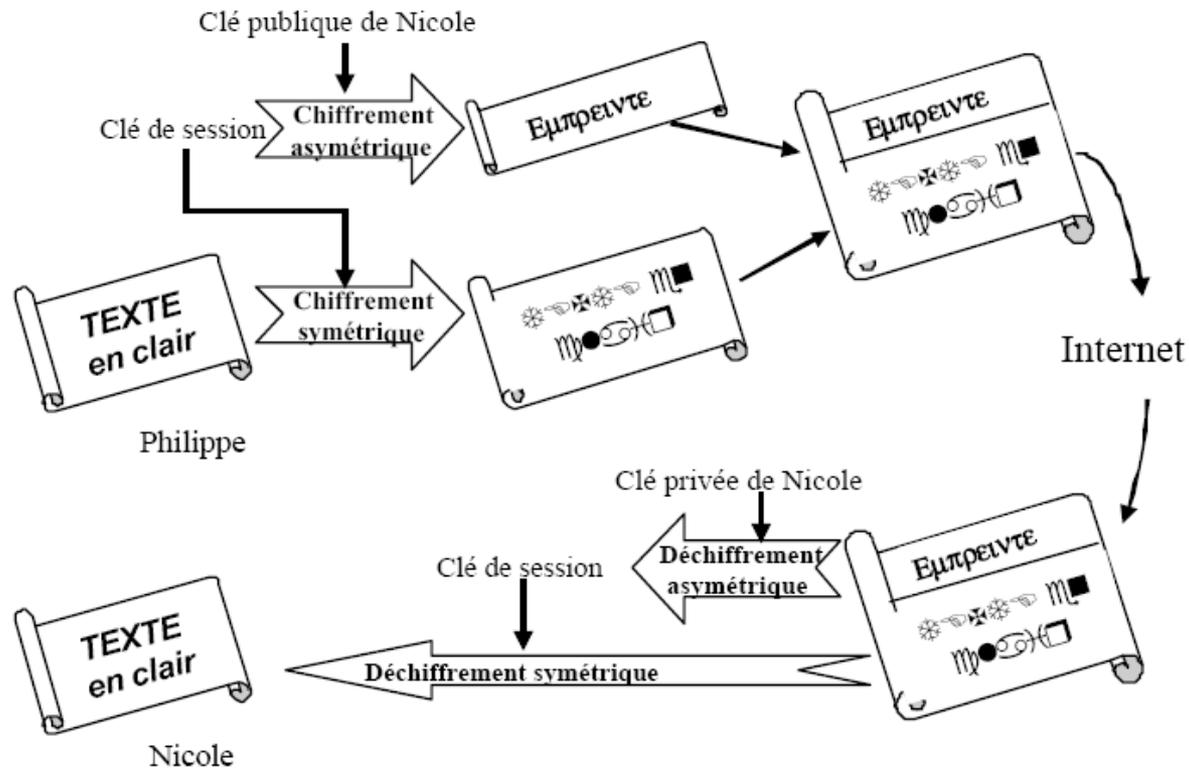
[La sécurité... 1/4]

Le chiffrement : confidentialité

- Pour assurer la sécurité d'un document électronique, on le chiffre
- Le chiffrement consiste à appliquer une fonction mathématique sur le fichier. La variable est appelée **clé de chiffrement**
- 2 types d'algorithmes :
 - Symétriques : clé de chiffrement = clé de déchiffrement (DES, AES) → pb : chaque clé doit être secrète et unique
 - Asymétriques : clé de chiffrement \neq clé de déchiffrement (RSA). Ces algorithmes sont aussi appelés à **clé publique**
 - création d'un couple de clés
 - mécanisme de clé publique (chiffrement) diffusée, et de clé privée (secrète)

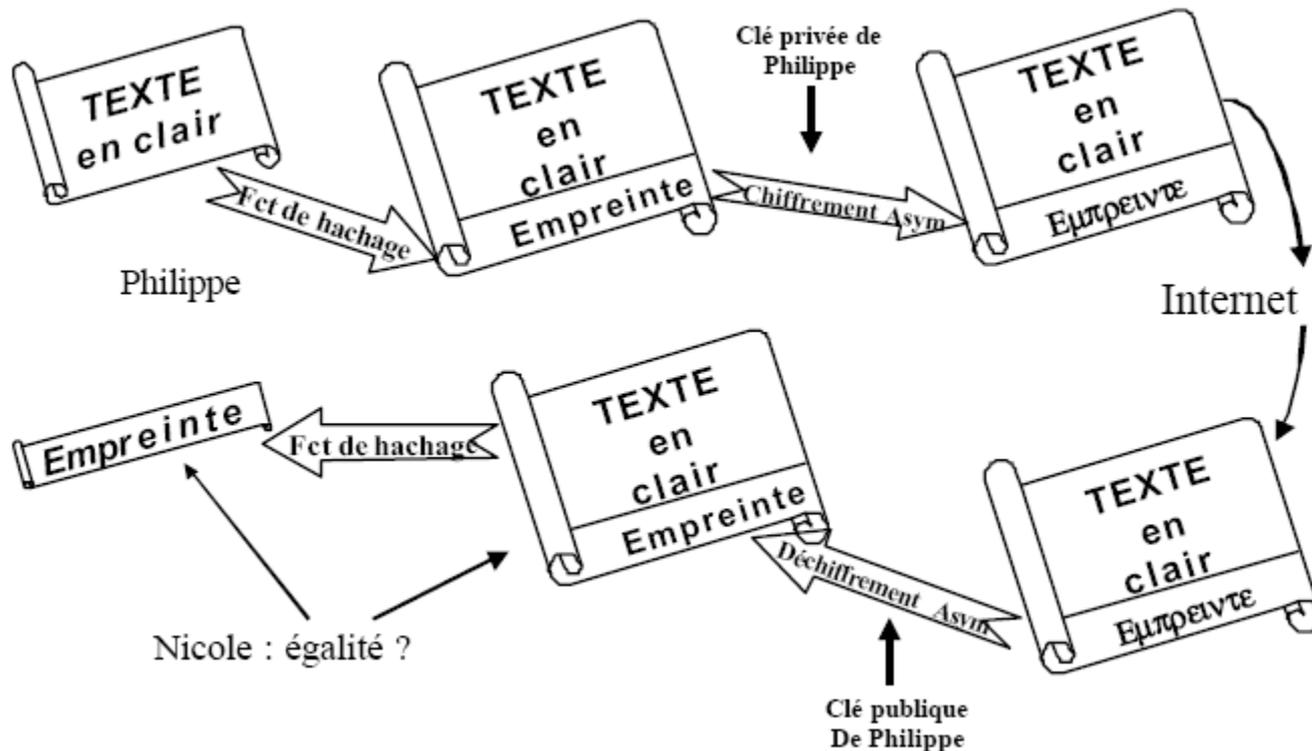
La sécurité... 2/4

Clé de session : confidentialité



La sécurité... 3/4

Signature électronique : authentification, intégrité



[La sécurité... 4/4]

Certificats : l'assurance

- Problème si la clé que l'on croit être d'une personne appartient à autre personne (malveillante)
- Il faut être sûr que l'on peut avoir confiance
- Mécanisme de certificats
- Autorité de certification



Exercice (rappel) 1/1

198.116.233.242 / 255.255.255.248

- Adresse du sous-réseau ?
 $248 = 128 + 64 + 32 + 16 + 8 \rightarrow 11111000$
 $255 - 7 (4 + 2 + 1)$
 $242 = 255 - 13 (8 + 4 + 1) \rightarrow 11110010$
- Nombre d'ordinateurs sur le sous-réseau ?
- Plage d'adresses du sous-réseau ?
- L'ordinateur peut-il communiquer avec celui ayant pour adresse **198.116.233.246** et pour masque **255.255.255.252** ?