



INTERNET





INTERNET

- INTERNET = ensemble de réseaux (Autonomous Systems ou AS) utilisant le protocole TCP/IP et connectés entre eux
 - Internetworking = interconnexion de réseaux
 - AS = domaine administratif
- TCP/IP = Transmission Control Protocol/Internet Protocol
- 2 versions d'IP : IPV4(adresse sur 32 bits) et IPV6(adresse sur 128 bits)





Standardisation dans Internet (i)

- Internet Society (ISOC)

- ✓ Promotion de l'Internet comme infrastructure globale de recherche en télécommunication

- Internet Architecture Board (IAB)

- Organisme chargé de la supervision et de la coordination, prend en charge la qualité des standards et dépend de l'ISOC

- Internet Engineering Task Force (IETF)

- Élabore et développe les spécifications qui deviennent les standards Internet. Il comporte des « sous comités » par thème et est dirigé par Internet Engineering Steering Group (IESG)

- Internet research Task Force (IRTF)

- Mène les recherche sur des projets à long terme





Standardisation dans Internet (ii)

- Request For Comment (RFC)
 - à chaque protocole # un RFC = norme
RFC EDITOR (<http://www.rfc-editor.org/>)
- World Wide Web Consortium (W3C)
 - prend en charge le développement du Web
- Internet Corporation for Assigned Names and Numbers (ICANN)
 - association qui assume la responsabilité de l'allocation des adresses IP, la gestion du système des noms de domaines et celle du serveur souche.
 - L'allocation des adresses IP aux différents pays est faite par les registres continentaux: APNIC (Asie-Pacifique), ARIN (Amérique du Nord), RIPE (Europe), AfriNIC (Afrique), ...
- Enregistrement des noms de domaines: AFNIC (Association Française qui enregistre les noms de domaines en .fr), ...



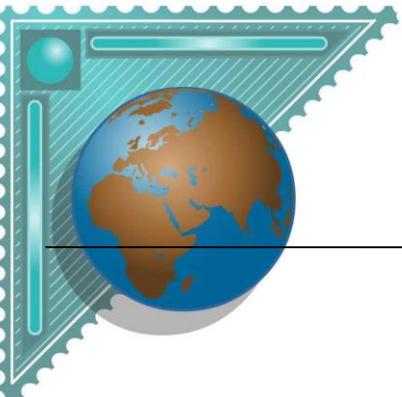


INTERNET

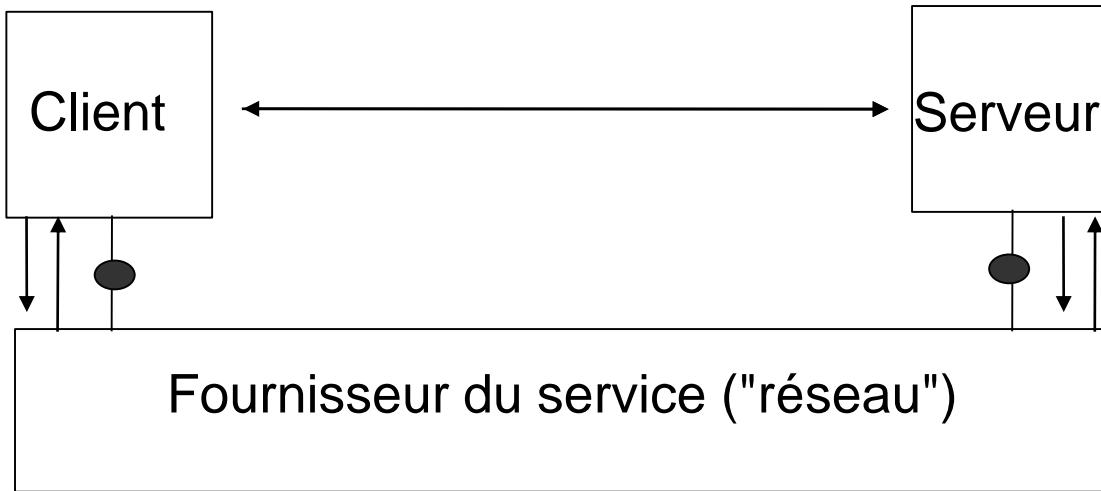


- Le modèle client serveur
- La résolution des noms en adresses
- Le courrier électronique
- Le world wide web





Modèle client-serveur

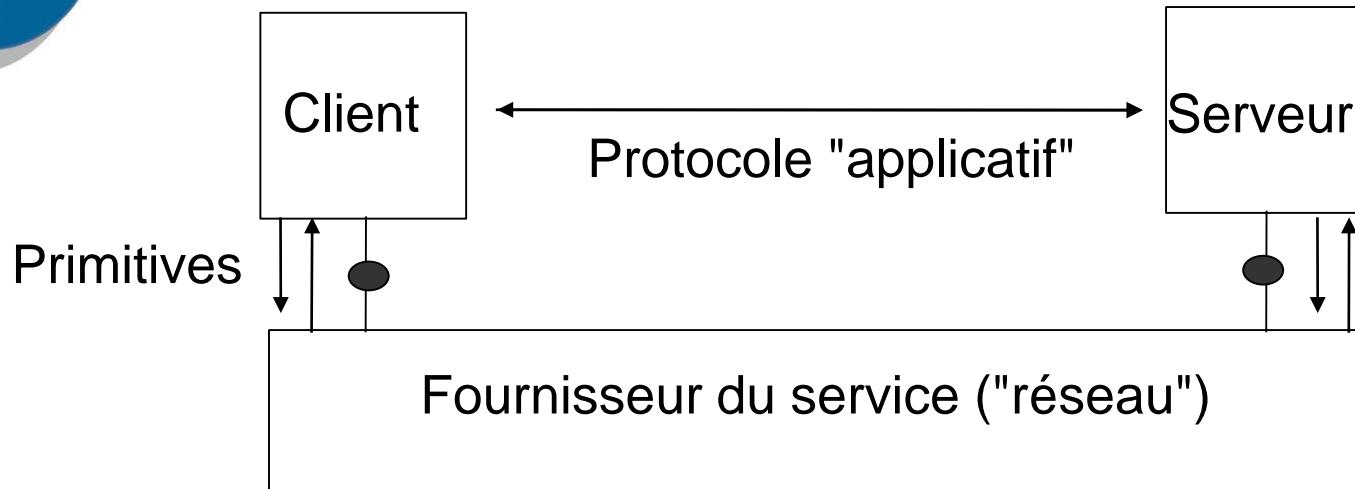


- Client
 - contacte le serveur via un service (TCP ou UDP)
 - envoie des demandes ou des commandes
- Serveur
 - répond aux demandes ou exécute les commandes des clients
 - plusieurs clients accèdent à un serveur
- Exemple : email, www, ...





Modèle client-serveur (2)



- Client interagit avec fournisseur du service
- serveur interagit avec fournisseur du service
- fournisseur donne au client l'illusion qu'il dialogue directement avec le serveur
- dialogue client-serveur possible uniquement si ils parlent le même "langage"
 - protocole "applicatif" : ensemble de règles syntaxiques et sémantiques qui définissent le langage du dialogue entre un client et un serveur

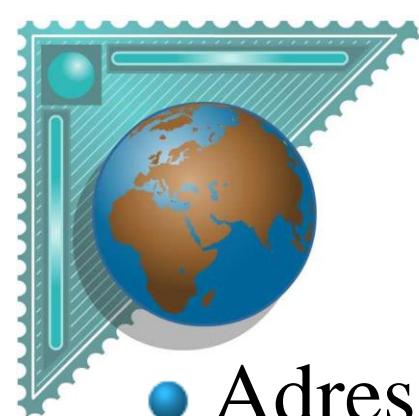




Applications Internet

- Le modèle client serveur
- ● La résolution des noms en adresses
- Le courrier électronique
- Le world wide web





Adresses et noms

- Adresse d'un serveur
 - Adresse IP de la machine sur laquelle il tourne
 - numéro de port (TCP ou UDP)
 - en général numéro bien connu pour chaque type de serveur
- Inconvénient
 - Adresse IP difficile à mémoriser par un humain
- Idée
 - remplacer l'adresse IP par un nom de machine
 - plus facile pour les humains





Adresses et noms (2)

- Fichier "hosts"

- contient la table adresse IP - nom de machine
- doit être distribué régulièrement

```
# Internet host table

127.0.0.1      localhost

138.48.32.99    babbage

138.48.32.100   leibniz

138.48.32.1     routeur

138.48.32.92    corneille

138.48.32.107   backus

138.48.20.152   arzach
```

- inutilisable dans un grand réseau





Noms de machines

- **Problème à résoudre**

- Eviter que deux machines différentes aient le même nom

- **Comment attribuer les noms de machines ?**

- Choisi par l'utilisateur

- risque d'avoir plusieurs fois le même nom
 - exemple : pc1, mac2, sun3, ...

- Au hasard

- exemple : GHTYUIOKSY
 - pas très user-friendly

- De façon hiérarchique

- nom d'une machine composé de deux parties
 - nom de la machine (à l'intérieur d'un domaine)
 - nom du domaine
 - Exemple : www.univ-tours.fr





Noms de machines (2)

Comment garantir l'unicité des noms hiérarchiques ?

- Attribuer les noms de domaine de façon hiérarchique
 - domaines de premier niveau
 - com, edu, net, gov, mil, int, ...
 - be, fr, de, uk, ...
 - chaque domaine de premier niveau peut déléguer des sous-domaines
 - exemple univ-tours.fr
 - chaque sous-domaine peut déléguer des sous-sous domaines ou des noms de machines
 - exemple
 - medecine.univ-tours.fr
 - www.cci.univ-tours.fr
 - desscci.univ-tours.fr
- Garantir l'unicité à chaque niveau pour unicité globale

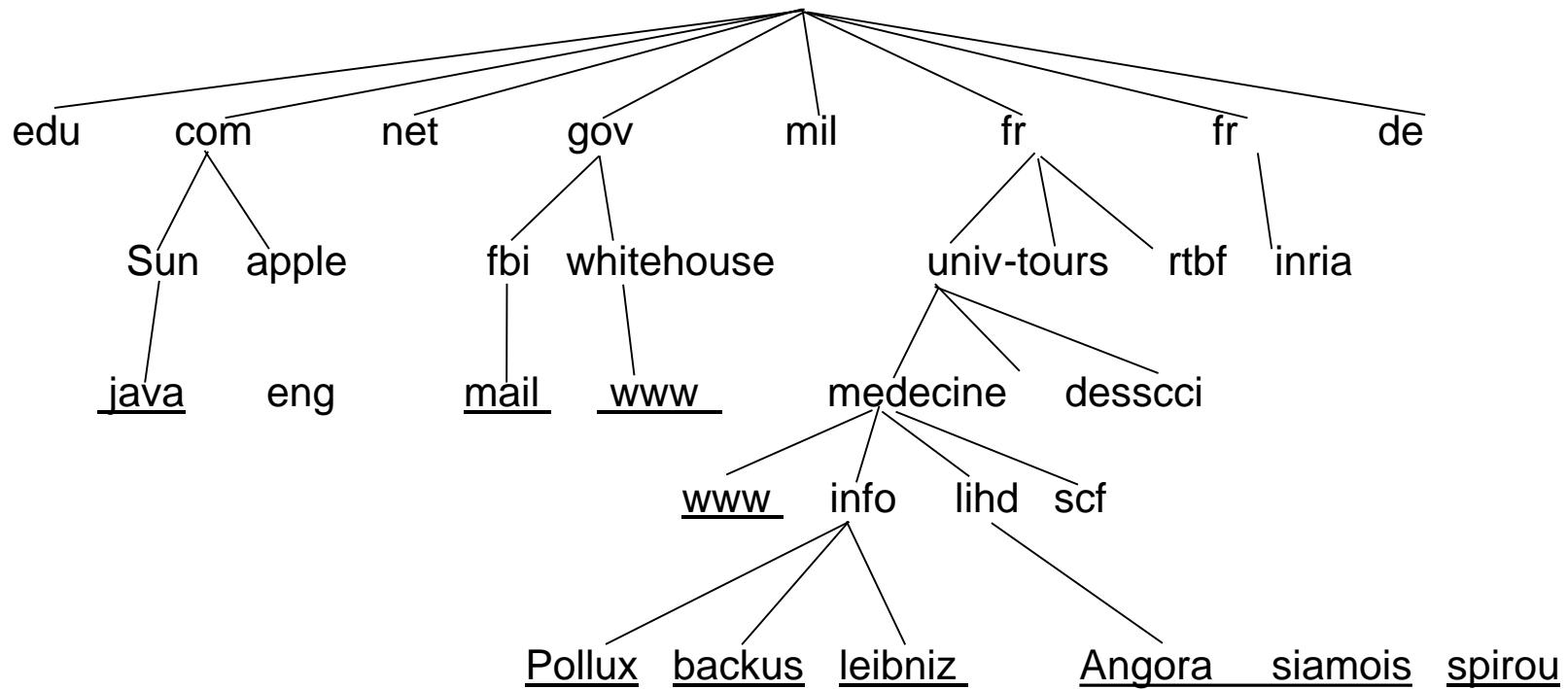




Noms de machines (3)

- Ensemble des noms de machines

- arbre

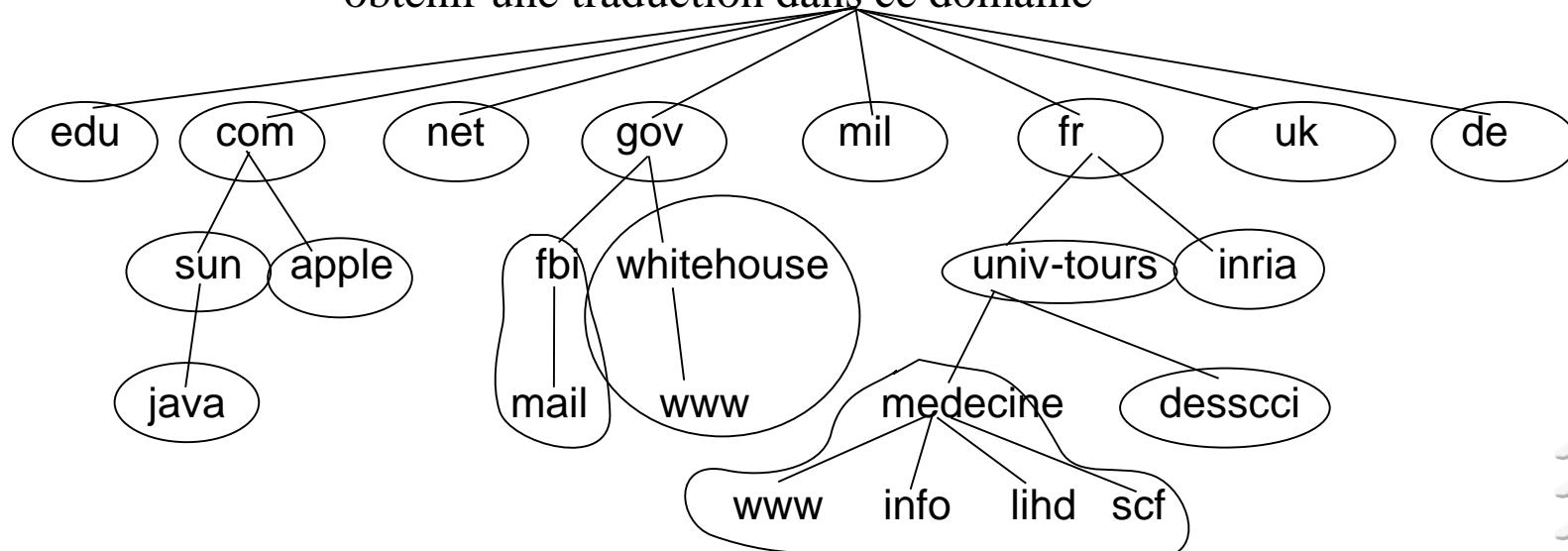




Traduction nom-adresse

Comment traduire un nom en une adresse ?

- Avec une BD centralisée
 - il y a plus de 50 millions de noms !!!
- Avec une BD distribuée
 - Comment distribuer la BD ?
 - DNS : Domain Name System
 - profiter de la hiérarchie des noms de machine
 - un serveur est responsable de chaque domaine et il faut le contacter pour obtenir une traduction dans ce domaine

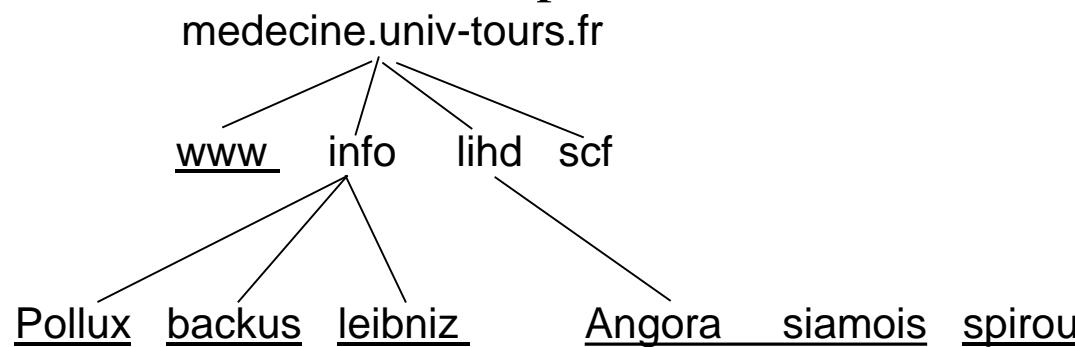




Traduction nom-adresse (2)

Serveur DNS

- Un serveur DNS est responsable d'un domaine
- Dans ce domaine il connaît
 - les adresses IP des machines du domaine
 - les adresses IP des serveurs DNS responsables de ses sous-domaines
- A partir d'un serveur DNS on peut "descendre" l'arbre

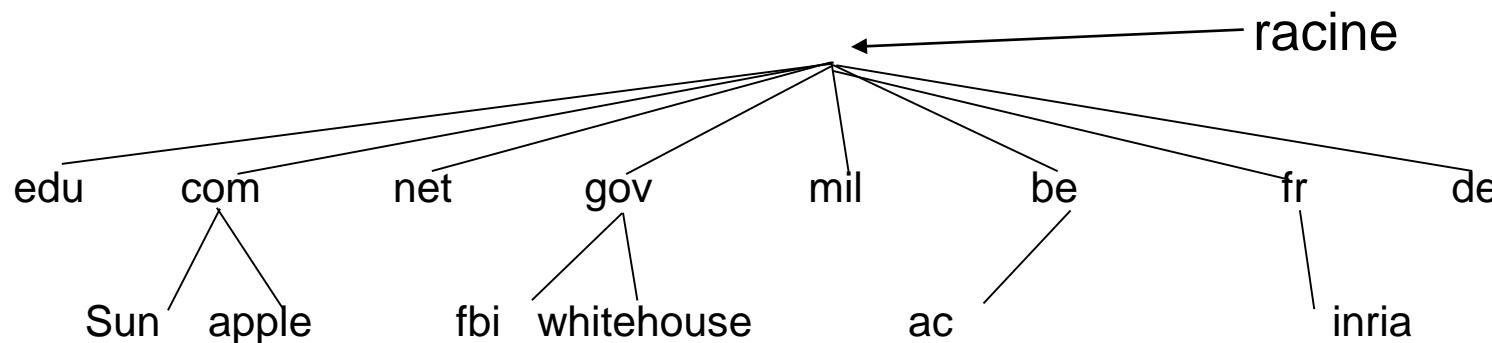




Localisation d'un serveur DNS

Comment localiser un serveur DNS ?

- A partir de n'importe quel serveur on peut descendre l'arbre des noms
- Serveurs racines (root servers)
 - un serveur racine connaît les adresses des serveurs DNS qui gèrent les domaines de premier niveau



- Plusieurs serveurs racines dont les adresses IP sont "bien connues"
- fichier "hosts" reprenant l'adresse des serveurs racines distribué régulièrement





Traduction d'un nom

- Pour traduire un nom, il faut
 - connaître les adresses IP des root-servers
 - pouvoir interroger un serveur DNS et parcourir l'arborescence de noms pour trouver la traduction
 - complexe si chaque machine doit supporter cela
- Solution choisie
 - serveur de noms
 - un serveur pour un ensemble de machines
 - réalise les traductions nom-adresse à la demande d'autres machines
 - le serveur de noms doit maintenir la liste des root-servers
 - machines "normales"
 - doit connaître l'adresse IP d'un serveur de noms





DNS : optimisations

- Réduire le risque de pannes
 - plusieurs root-servers
 - Plusieurs serveurs DNS responsables d'un domaine
 - machine peut interroger plusieurs serveurs de noms
- Améliorer les performances
 - éviter de demander successivement la même info
 - mémoire cache dans les serveurs de noms
 - mémoire cache contenant
 - traduction nom-adresse "récentes"
 - adresses IP des serveurs DNS "récemment" contactés
- Le DNS et les serveurs de noms utilisent UDP





DNS : resource records

- Chaque message DNS contient un ensemble de resource records (RR) qui est encodé sous la forme
 - < Nom, Valeur, Type, TTL >
- Types de RR
 - Type A (Address)
 - Nom est un nom de machine et Valeur l'adresse IP
 - Type NS (NameServer)
 - Nom est un nom de domaine et Valeur contient le nom de la machine qui gère le DNS de ce domaine
 - Type MX (Mail Exchange)
 - Nom est un nom de domaine et Valeur contient le nom du serveur SMTP pour les mails envoyés à ce domaine
 - Type CNAME

Durée pendant laquelle un RR peut être conservé en cache par un serveur





DNS : resource records

<http://www.kloth.net/services/nslookup-fr.php>





Applications Internet

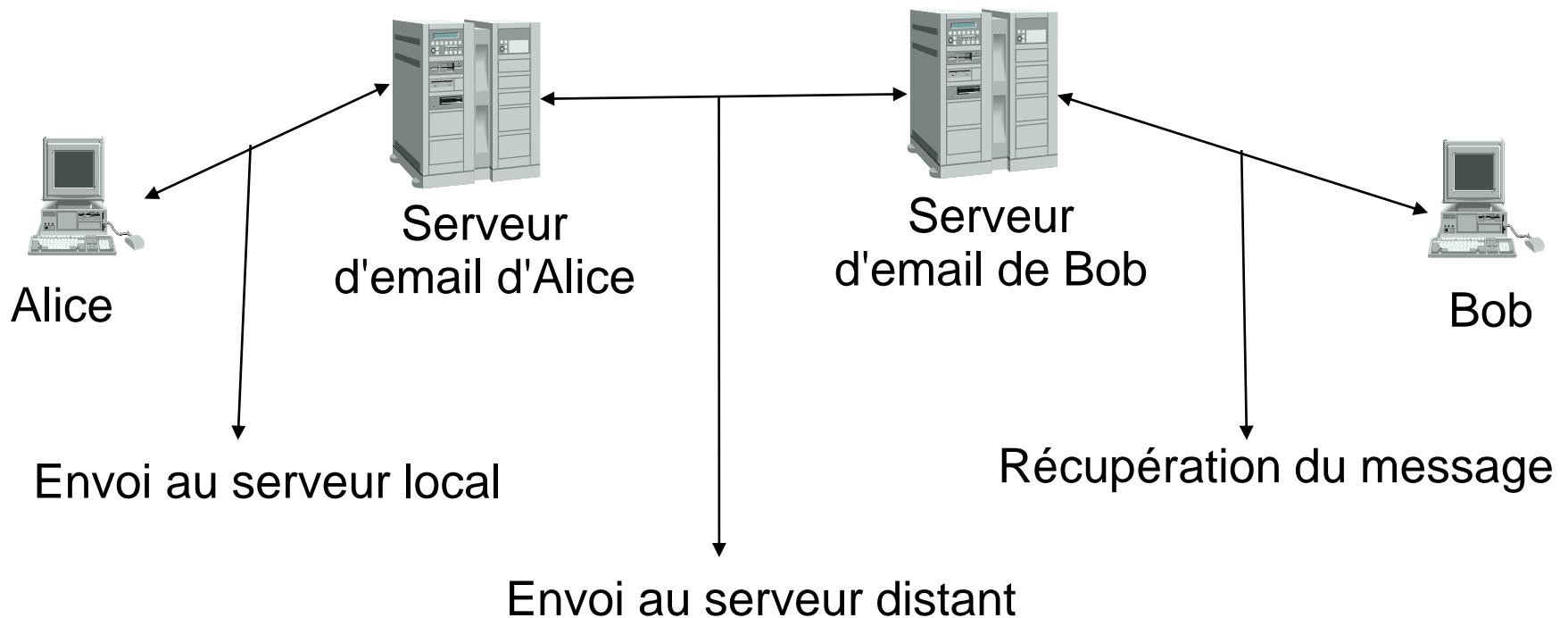
- Organisation du module
 - Le modèle client serveur
 - La résolution des noms en adresses
- - Le courrier électronique
 - Le world wide web





Courrier électronique

- Modèle simplifié
 - Envoie d'un message de Alice à Bob





Format des messages email

Exp: ABC S.A.
Rue de Fer 10
37000 Tours

DEF Corp.
Steel street 9
WA78 AX London
Grande Bretagne

From: teguia@univ-tours.fr
To: ceo@def.com
Subject: Bonjour
Date : 21 Sept. 2002 0901

Cher Monsieur,

Bla Bla Bla...

entête

Corps
du
message





Format des messages (2)

- Format de l'entête
 - composée de caractères ASCII
 - Seuls les caractères US-ASCII (7bits) sont autorisés
 - Minimum trois lignes terminées par <CRLF>
 - From: expediteur@domaine
 - To: destinataire@domaine
 - Date: <date de creation du message>
 - exemple : 08 Dec 2005 1445 EDT
 - Champs optionnels (terminés par <CRLF>)
 - Subject: sujet du message
 - cc: copie@domaine
 - Message-ID: <numero@domaine>
 - Received: information sur le chemin suivi par le message
 - In-Reply-To: <message-ID>
 - Entête se termine par ligne vide (<CRLF>)





Contenu des messages

- A l'origine
 - courrier électronique développé par des anglophones
 - message composé de lignes ASCII 7 bits
 - chaque ligne est terminées par <CRLF>
 - pas de limite a priori sur la taille des lignes ou des messages, mais nombreux serveurs en imposent
- Comment dans un email transmettre ?
 - des caractères accentués
 - autre chose que du texte ASCII
 - audio
 - video
 - programmes...





MIME

- Multipurpose Internet Mail Extensions
- Redéfinition du format des messages
 - objectifs
 - rester le + possible compatible avec d'anciens serveurs d'email
 - supporter des textes dans une autre langue que l'anglais
 - nécessité d'identifier le type d'encodage des caractères puisque ASCII 7 bits ne suffit plus
 - pouvoir transférer autre chose que du texte
 - nécessité d'identifier les différents types de contenu
 - pouvoir créer des messages composés de plusieurs parties
 - fichiers attachés
 - Solution choisie
 - ajouter de nouveaux champs optionnels dans l'entête du message
 - ajouter lorsque c'est nécessaire certains éléments dans le corps du message





MIME (2)

- Nouveaux champs dans l'entête

- MIME-Version:

- indique la version de MIME utilisée pour créer le message
- version actuelle : 1.0

- Content-Description:

- texte ASCII (commentaire) décrivant le contenu

- Content-Type:

- indique la nature du message

- Content-Transfer-Encoding:

- indique la façon dont le message a été encodé

- Content-Id:

- identificateur unique pour le contenu





MIME: Content-Type

- Content-Type composé de deux parties

- type de contenu

- text, image, video, application
 - multipart

- format du contenu

- text/plain , text/html
 - image/gif, image/jpeg
 - audio/basic
 - video/mpeg, video/quicktime
 - application/octet-stream, application/postscript
 - multipart/alternative

- message composé de plusieurs parties représentant le même contenu dans des formats différents (ex: deux formats d'image)

- multipart/mixed

- message composé de plusieurs contenus différents
 - exemple : partie texte et fichier attaché





Caractères accentués

- Support des caractères accentués dans les contenus de type text/plain
- paramètre indiquant le code de caractères utilisé pour encoder le message
 - Content-Type: text/plain; charset=us-ascii
 - norme définie fin des années 1960s aux USA
 - caractères ASCII 7 bits, défaut
 - Content-Type: text/plain; charset=iso-8859-1
 - code de caractère Latin1 pour Europe défini par l'ISO
 - 8 bits pour chaque caractère
 - les caractères 0-127 sont équivalents à US-ASCII
 - Content-Type: text/plain; charset=unicode
 - code de caractère universel, supporte toutes les langues
 - 16 bits pour chaque caractère
 - partiellement compatible avec ISO-8859-1

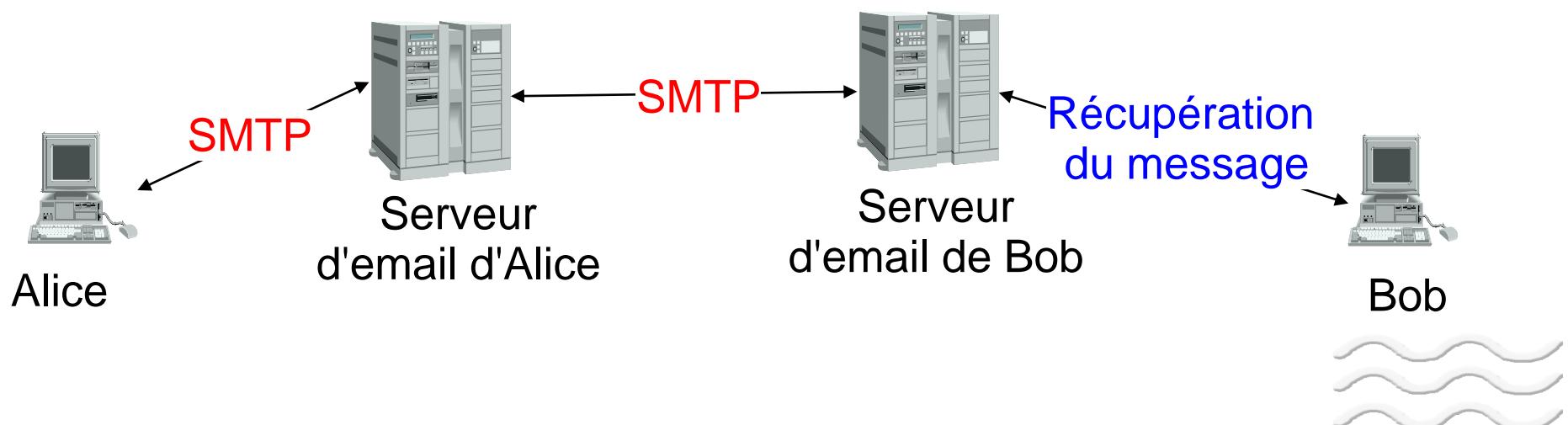




Transmission des messages

SMTP : Simple Mail Transfer protocol

- utilise le service TCP
- adresse du serveur SMTP
 - adresse IP de la machine + TCP + numéro de port : 25
 - Les RR de type MX du DNS permettent de localiser le serveur SMTP du domaine @abc.xyz





SMTP

Modèle client serveur

- Le serveur attend les messages à relayer
- Le client envoie les messages via le serveur
 - Fonctionnement du protocole
- le client envoie des commandes
 - commande paramètre <CRLF>
 - HELO
 - MAIL FROM:
 - RCPT TO:
 - DATA
 - QUIT
- le serveur répond par des messages
 - code_numérique description textuelle
 - 250 OK
 - 221 closing





SMTP (2)

Trois phases de SMTP

1. établissement d'une association SMTP

- ouverture de la connexion TCP
- message de bienvenue du serveur
- HELO

2. Transfert d'un message

- MAIL FROM: <user@domaine>
- RCPT TO: <user@domaine>
- DATA
 - transfert du message complet, entête comprise
 - caractère "." sur ligne vide pour terminer le message
- transfert éventuel d'autres messages

3. Fermeture de l'association

- QUIT





Récupération des messages

• Deux cas de figure

1. Machine destination avec serveur SMTP et connecté en permanence au réseau

- stocker les messages qui arrivent dans un fichier au nom du destinataire
- utiliser les permissions de l'OS pour s'assurer que seul le destinataire pourra les lire
 - /var/mail sur Unix

2. Machine destination sans serveur SMTP

- faire arriver les messages sur un serveur SMTP relais
- récupérer les messages de façon asynchrone
 - Post Office Protocol (POP)
 - Internet Mail Access Protocol (IMAP)





POP

- Objectif

- Permettre la récupération de messages email à distance avec authentification des utilisateurs

- Fonctionnement

- POP utilise le service TCP

- adresse du serveur POP

- adresse machine + TCP + numéro de port : 110

- client envoie des commandes

- commande : ligne ASCII terminée par <CRLF>

- USER, PASS, STAT, RETR, DELE, QUIT

- serveur répond avec

- +OK si réussite

- -ERR si erreur

- envoi des messages à la demande





POP (2)

• Trois phases du protocole

1. Autorisation : vérification de l'utilisateur

- USER
- PASS

2. Transaction

- récupération et suppression des messages
- STAT
 - liste des messages
- RETR <n>
 - récupération d'un message
- DELE <n>
 - message marqué pour suppression

3. Mise à jour

- Fin de la récupération des messages
 - messages marqués pour suppression sont effacés
 - fin de la connexion TCP





Applications Internet

- Le modèle client serveur
- La résolution des noms en adresses
- Le courrier électronique
- Le world wide web

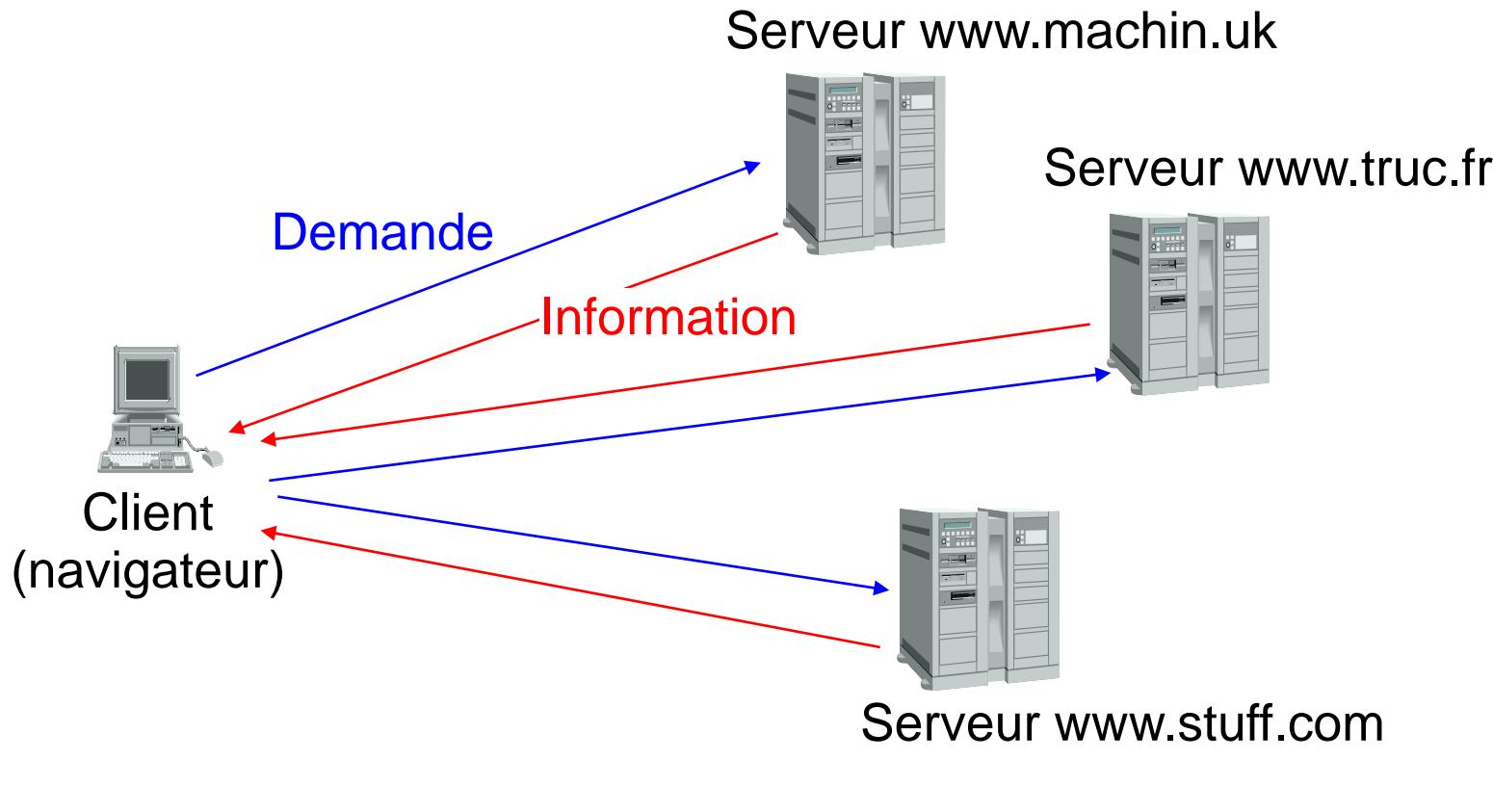




World Wide Web

Objectifs

- permettre la mise à disposition d'informations via des serveurs tout en autorisant des liens hypertextes entre informations situées sur des sites différents





Le World Wide Web (2)

- Initialement conçu comme une base documentaire
 - Très largement distribuée
 - Structure très hétérogène
 - Multi-langue
- Maintenant, mode d'interaction avec les SI dynamiques en Intranet ou en Extranet
 - Formulaires
 - Nouveau mode d'architecture: client léger et les architectures multi-tiers





World Wide Web (3)

- Les éléments clés du www

- un mécanisme d'adressage permettant d'identifier un document sur n'importe quel serveur
 - **URL** : Uniform Resource Locator
- un langage de description de documents qui permet d'inclure des liens hypertextes
 - **HTML** : HyperText Markup Language
- un protocole d'échange de documents HTML
 - **HTTP** : HyperText Transfer Protocol
- un logiciel serveur
- un logiciel client (navigateur)





Les serveurs Web

- NSCA HTTP (le premier)
- Apache (le plus répandu et multi-plaforme)
- IIS (Microsoft internet Server)
- Sun JavaServer
- Netscape Entreprise Server





Les clients HTTP

- Butineur ou arpenteur ou navigateur (browser)
 - Récupère un document
 - Puis le visualise

- NCSA Mosaic (le tout premier)
- Lynx
- Mozilla
- Opera
- Netscape communicator
- Javasoft HotJava
- MS Internet Explorer (le plus utilisé)





Le navigateur Web et les types de documents

Agit en fonction de ses capacités

De base, il sait « jouer »

Txt, html, gif, jpg

Sinon existe-t-il un « plug-in », qui « joue » le document dans le navigateur

.vrml plugin COSMO

.pdf plugin Acrobat Reader

Sinon c'est transmis à une application externe

.doc application MS word

.zip application WinZip

.ra application RealAudio





Les documents et le type MIME

Du texte

(.txt)

text/plain

Des hypertextes

(.html, .htm, .xml, .wml)

text/html, text/xml

Des sons

(.au, .snd, .wav, .mp3, ...)

sound/*

Des images

(.gif, .jpg, .jpeg, .bmp, ..)

images/*

Des scènes virtuelles (virtual reality)

(.vrml, .wrl, .3d, ...)

world/*

Des documents d'autres applications

(.ps, .pdf, .doc, .ppt, .zip, .gz)

application/*





Uniform Resource Locator (URL)

- Uniform Resource Locator (URL)

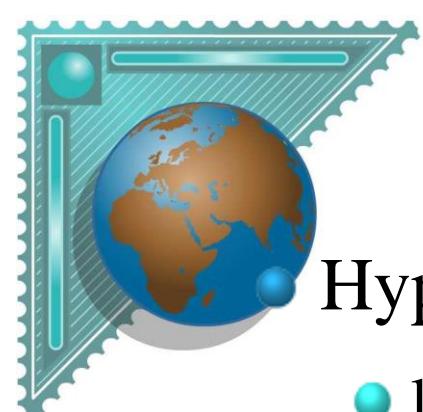
- forme générale : <protocole>://<document>

- protocole indique le protocole à utiliser pour récupérer le document du serveur
 - http est le plus courant mais d'autres protocoles sont utilisés
 - document spécifie le serveur qui contient le document et l'endroit où se trouve le document sur ce serveur

- forme générale <user>:<password>@<machine>:<port>/<chemin>

- <user> : nom d'utilisateur optionnel
 - <password> : mot de passe optionnel
 - <machine> : nom ou adresse IP du serveur contenant le document
 - <port> : numéro de port optionnel
 - <chemin> : identification du document sur le serveur
 - exemples
 - http://www.cci.univ-tours.fr
 - http://desscci.univ-tours.fr



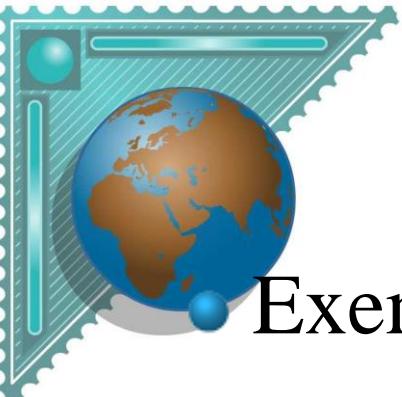


HTML

HyperText Markup Langage

- langage de marquage de documents utilisés sur le world wide web
- quelques mot clés du langage
 - <HTML> . . . </HTML>
 - <HEAD> . . . </HEAD>
 - <BODY> . . . </BODY>
 - <TITLE> . . . </TITLE>
 - . . .
 - <I> . . . </I>
 - <H1> . . . </H1>
 - <P>
 - <HR>
 - . . .
 - . . .
 -
 - *text anchor*





HTML (2)

Exemple

Entête

Corps

```
<HTML>
  <HEAD>
    <TITLE>Page de test en HTML</TITLE>
  </HEAD>
  <BODY>
    <IMG SRC="http://www.images.com/logo.gif">
    <H1>Serveurs www de l'université de Tours<P></H1>
    <HR>
    <UL>
      <LI><A HREF="http://www.medecine.univ-tours.fr">Medecine</A>
      <LI><A HREF="http://www.iut.univ-tours.fr">IUT</A>
      <LI><A HREF="http://www.cci.univ-tours.fr">DESS CCI</A>
    </UL>
  </BODY>
</HTML>
```

Image sur serveur distant à inclure

Titre de niveau 1

Lien hypertexte extérieur

- Lorsqu'un navigateur récupérera cette page sur le serveur où elle résidera, il contactera le serveur **www.images.com** pour obtenir le fichier **logo.gif** qu'il incluera dans le document.



HTML(3) : les frames

- Visualisation de plusieurs documents dans différents cadres (frames) dans la page du navigateur
- Les actions dans un cadre peuvent agir sur un autre cadre

```
<FRAMESET COLS=<< 50%,50% >>>
    <FRAME SRC=docgauche.htm>
<FRAMESET ROWS=<< 33%,66% >>>
    <FRAME SRC=docdroitehaut.htm>
    <FRAME SRC=docdroitebas.htm>
</FRAMESET>
</FRAMESET>
<NOFRAMES>
Document non supporté par votre browser
</NOFRAMES>
```





Les évolutions de HTML

● Qui normalise?

- W3C (www.w3.org)

- HTML 4.0, XHTML

- CSS : Cascade Style Sheet
 - DOM : Document Object Model

Chaque élément (texte, graphique, style, ..) est un sous objet lui-même composite

- XML (eXtended Markup Language)

- DHTML (Dynamic HTML)

- DHTM = HTML + CSS + Layers + Scripting
 - Des zones d'une frame peuvent changer par scripting
 - Différent entre MSIE et NS communicator

- Layer

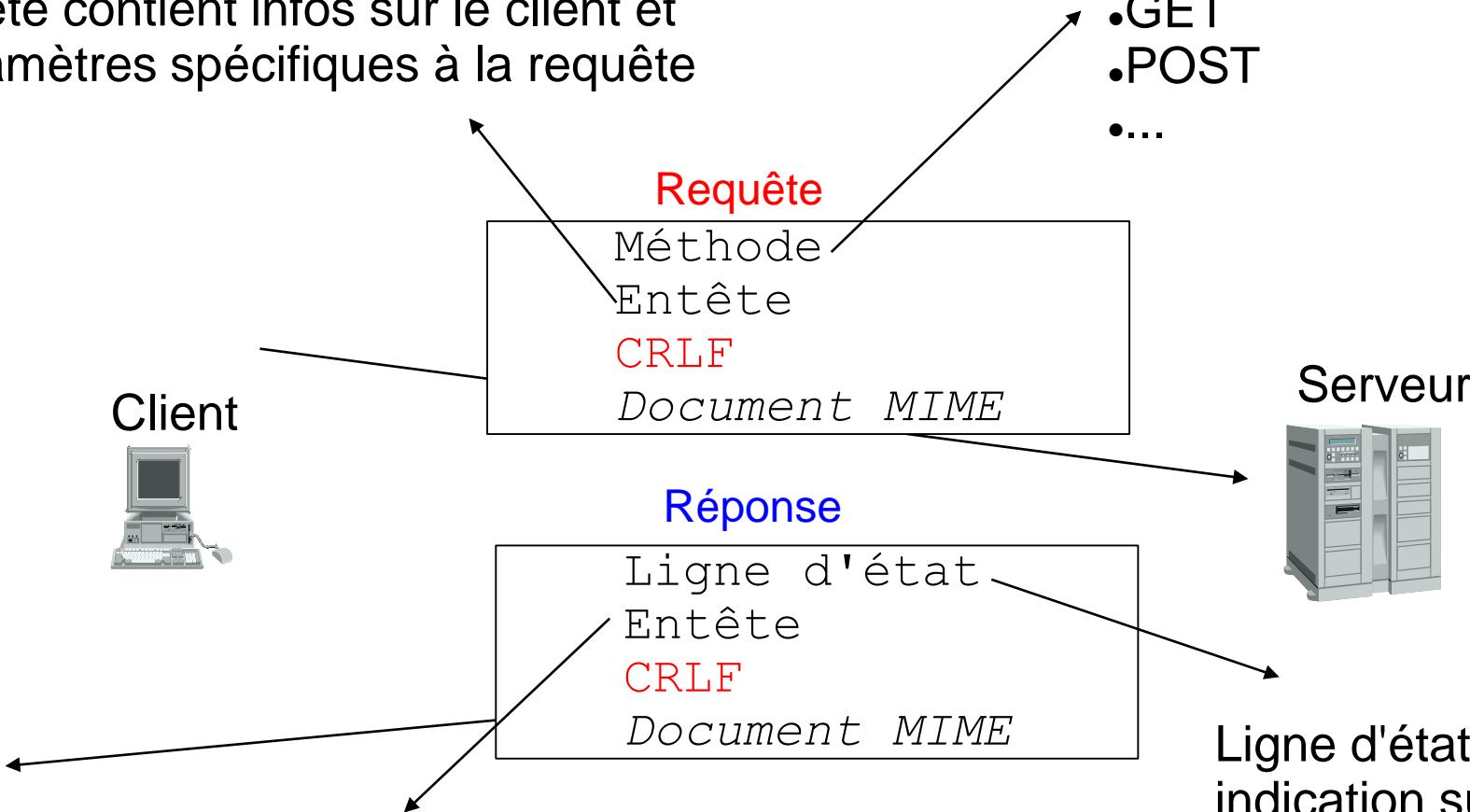
- Une frame contient plusieurs fenêtres flottantes





HTTP

Entête contient infos sur le client et paramètres spécifiques à la requête



Entête contient infos sur le serveur et paramètres spécifiques à la réponse

Méthode = Type de requête
• GET
• POST
• ...

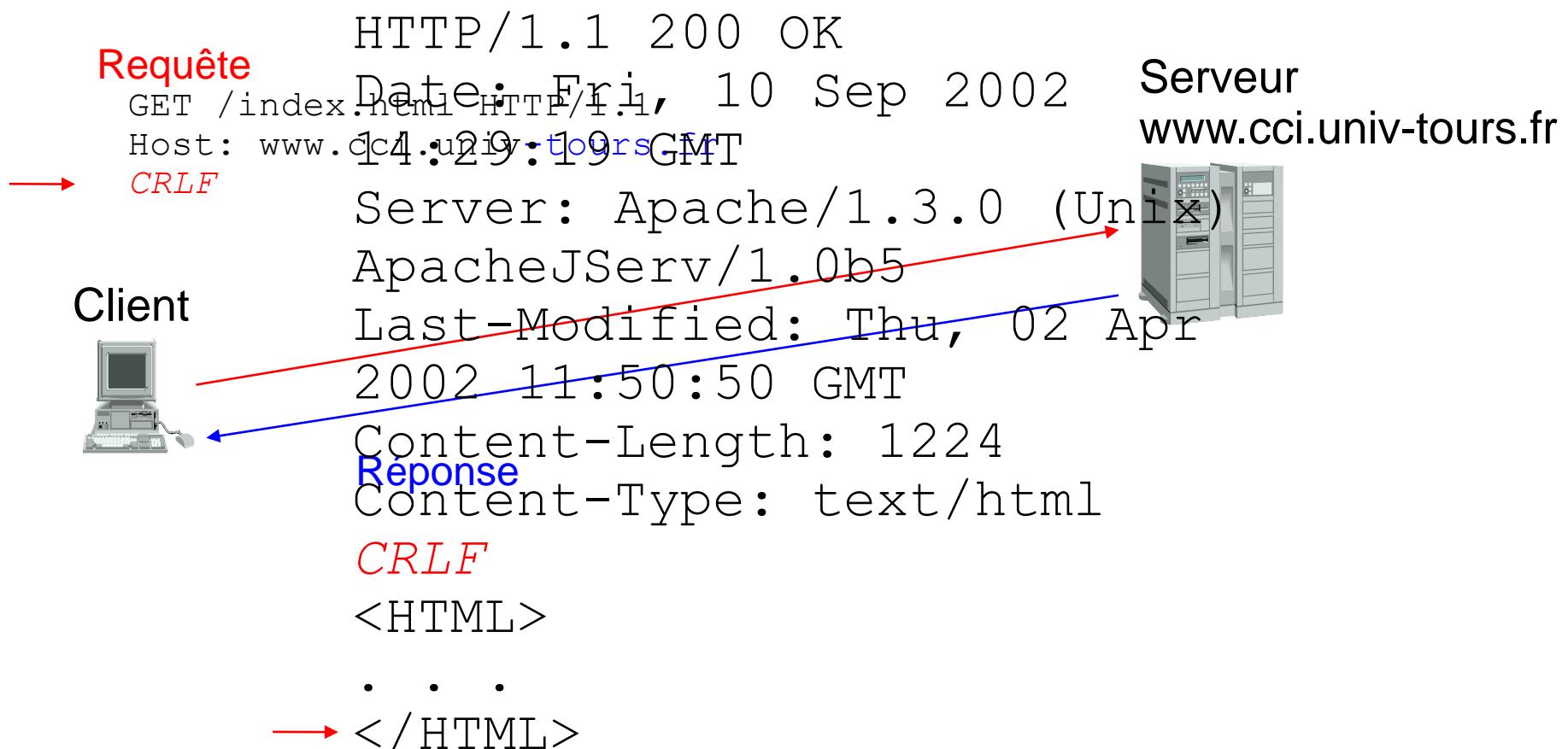
Ligne d'état
indication sur le succès ou l'échec de la requête

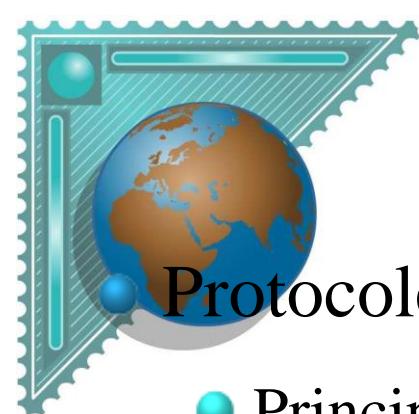
HTTP est stateless, le serveur ne maintient pas d'état d'une requête à l'autre
• contrairement à POP et SMTP par exemple





HTTP : Exemple





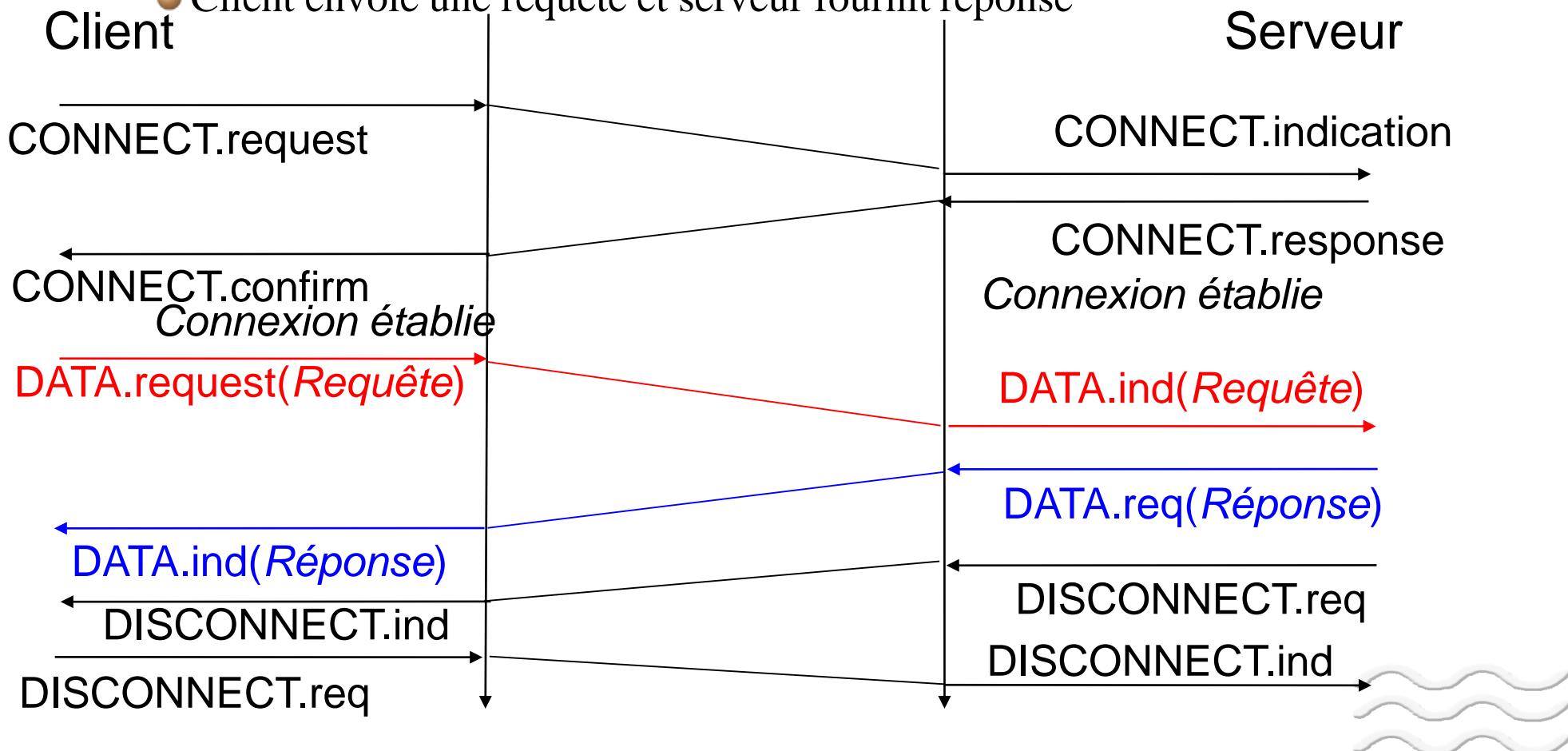
Transfert d'informations www

Protocole HTTP 1.0

Principe

- Utilise le service TCP (serveur utilise le port 80 par défaut)

- Client envoie une requête et serveur fournit réponse





HTTP : Méthodes

● Méthodes

● GET

- méthode utilisée pour demander le transfert d'un "document" stocké sur le serveur contacté
- format
 - GET <document> HTTP/1.0
- exemple
 - GET /index.html HTTP/1.0

● POST

- méthode utilisée pour transmettre un document au serveur
- dans ce cas, le document est annexé à la requête sous la forme d'un fichier au format MIME



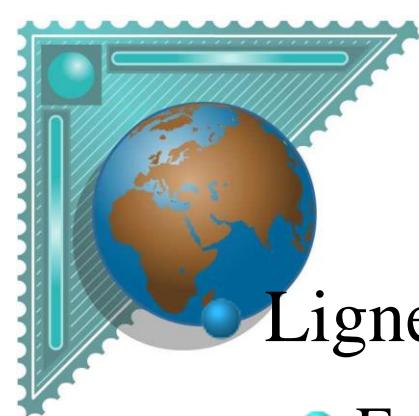


HTTP : Entêtes des requêtes

Entêtes

- Permettent au client de paramétriser la requête et de fournir de l'information sur ses capacités
- Host: <[name](#)>
 - **Nom** de la machine sur laquelle le document réside
- Authorization
 - pour réaliser du contrôle d'accès à certains URL
- If-Modified-Since: <[date](#)>
 - le serveur ne doit fournir le document demandé que si celui-ci est postérieur à la **date** spécifiée dans l'entête
- Referer: <[url](#)>
 - information, indique l'**URL** visité par le client précédemment
- User-Agent: <[agent](#)>
 - information, indique le client utilisé pour la requête
- Accept*
 - types de documents, encodages, langues supportés par client





HTTP : Ligne d'état des réponses

Ligne d'état

- Format : Version_HTTP **Code** **Commentaire**
- Codes d'erreur/succès
 - **1xx** : Pour information (inutilisé)
 - **2xx** : Réussite
 - Exemple : HTTP/1.0 200 OK
 - **3xx** : Redirection
 - pour être réussie, la requête doit être transmise ailleurs
 - Exemple : HTTP/1.0 301 Moved permanently
 - document MIME annexé contiendra l'URL à utiliser pour le document
 - **4xx** : Erreur imputable au client
 - exemples : erreur de syntaxe dans la requête, URL non accessible, requête non autorisée, ...
 - **5xx** : Erreur imputable au serveur
 - exemples : erreur interne, méthode non implémentée sur le serveur



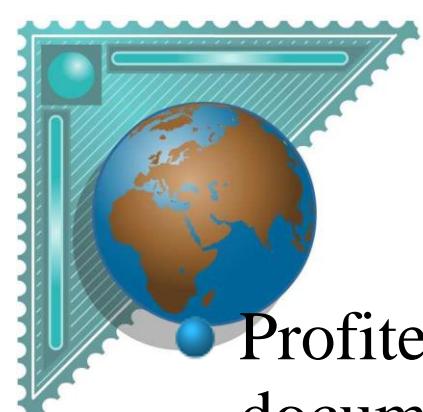


HTTP : Entête des réponses

Entête

- information complémentaire sur le serveur, la réponse ou le document annexé à la réponse
- Date
 - date du document qui est fourni dans la réponse
 - exemple : Date: Wed, 05 Sep 2002 13:27:34 GMT
- Server
 - identification du logiciel serveur http utilisé
 - exemple :
Server: Apache/1.3.20 (Unix) ApacheJServ/1.1.2
PHP/4.0.6
- Content-*
 - entête MIME du document annexé
 - exemple :
Content-Length: 5891
Content-Type: text/html





Amélioration des performances

Profiter du principe de localité et de la stabilité des documents

- de nombreux documents changent rarement
- un utilisateur peut visualiser plusieurs fois le même document
- deux utilisateurs proches ont de bonnes chances d'avoir de demander des documents similaires
- Solution
 - installer une cache (disque et mémoire) dans chaque navigateur
 - if-modified-since pour vérifier si un document en cache est toujours valide
 - installer des caches institutionnelles





Amélioration des performances (2)

Client1



ClientN



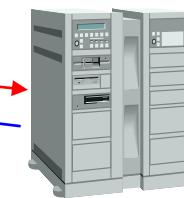
- Dialogue uniquement avec le proxy → Serveur

- ignore certains détails des clients

Proxy



Serveur



- Effectue toutes ses requêtes à travers le proxy

- connexion TCP du client au proxy

- permet de profiter de la cache

- utile aussi pour cacher ou contrôler les clients

- Effectue les requêtes demandées par les clients

- éventuellement traduction de protocole
 - gère la cache
 - attention aux cookies etc.

- transmet les réponses aux clients



Personnalisation de l'information

Problème

- Comment personnaliser l'information fournie à un utilisateur en fonction de ses préférences ?
 - Exemple : choix de la langue sur un site, journaux, ...
- HTTP est un protocole stateless, le serveur ne se souvient donc pas des choix faits par le client
- Comment permettre au serveur de mémoriser certains choix du client ?
 - Forcer le client à envoyer un résumé de ses choix dans chaque requête
 - authentification
 - cookies

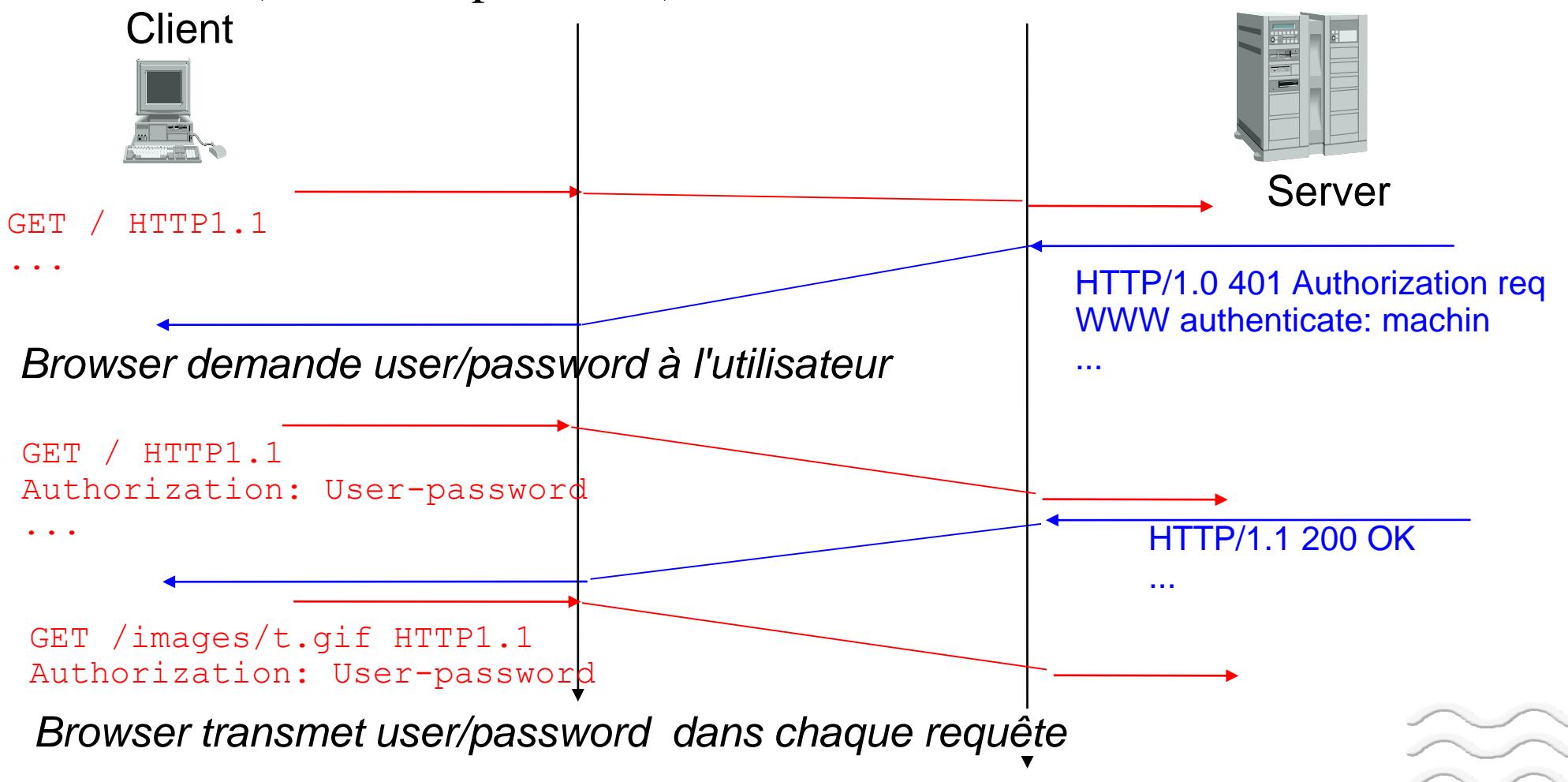




Authentification HTTP

Principe

- ajouter dans les requêtes http une authentification (username/password) de l'utilisateur

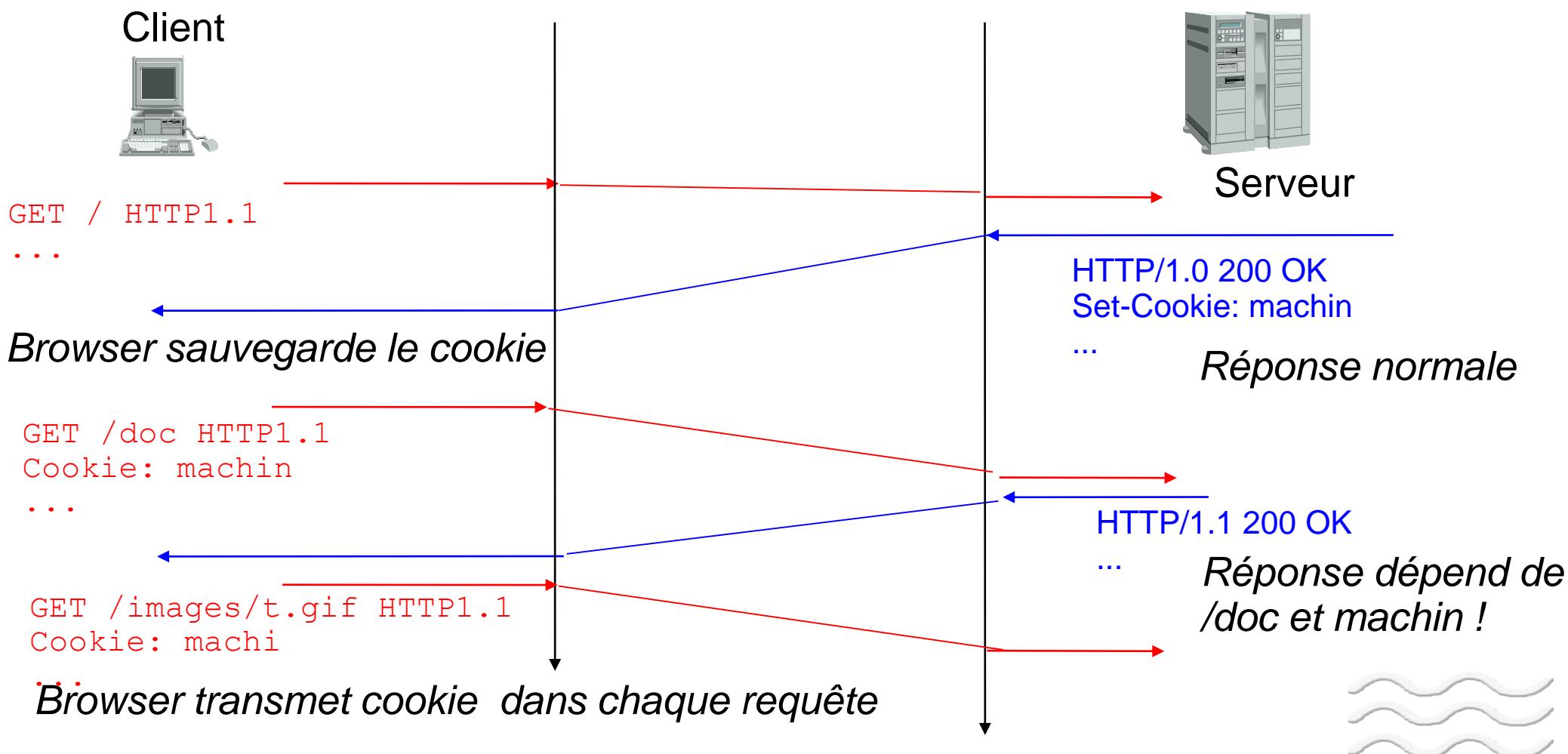




Cookies HTTP

Objectif

- obtenir le même résultat qu'avec l'authentification sans nécessiter une intervention de l'utilisateur





Résumé

- Modèle client-serveur
- Services UDP et TCP
- Protocoles applicatifs
 - DNS
 - utilise le service UDP, stateless
 - SMTP, POP
 - utilise le service TCP, statefull
 - HTTP
 - utilise le service TCP, stateless

