



Entêtes HTTP

Entêtes HTTP

- ✓ **4 types de champs d'entête**
 - **Général**
 - Commun au serveur, au client ou à HTTP
 - **Requête du client**
 - formats de documents et paramètres pour le serveur
 - **Réponse du serveur**
 - informations concernant le serveur
 - **Entité**
 - informations concernant les données échangées

Entêtes Généraux

- ✓ **Cache-Control**
 - contrôle du caching.
- ✓ **Connection = listes d'option**
 - close pour terminer une connexion.
- ✓ **Date**
 - date actuelle (format RFC1123 mais aussi RFC850).
- ✓ **MIME-Version**
 - version MIME utilisé.
- ✓ **Pragma**
 - instruction pour le proxy.
- ✓ **Transfer-Encoding**
 - type de la transformation appliquée au corps du message.
- ✓ **Via**
 - utilisé par les proxys pour indiquer les machines et protocoles intermédiaires.
- ✓ **....**

Entêtes de Requêtes Client (1)

- ✓ **Accept**
 - type MIME visualisable par l'agent
- ✓ **Accept-Encoding**
 - méthodes de codage acceptées
 - compress, x-gzip, x-zip
- ✓ **Accept-Charset**
 - jeu de caractères préféré du client
- ✓ **Accept-Language**
 - liste de langues
 - fr, en, ...
- ✓ **Authorization**
 - type d'autorisation
 - BASIC nom:mot de passe (en base64) (donc en transmis en clair!)
 - NB : Préalablement le serveur a répondu un WWW-Authenticate
- ✓ **Cookie**
 - cookie retourné

Entêtes de Requêtes Client (2)

- ✓ **From**
 - adresse email de l'utilisateur
 - rarement envoyé pour conserver l'anonymat de l'utilisateur
- ✓ **Host**
 - spécifie la machine et le port du serveur
 - un serveur peut héberger plusieurs serveurs
- ✓ **If-Modified-Since**
 - condition de retrait
 - la page n'est transférée que si elle a été modifiée depuis la date précisée. Utilisé par les caches
 - indique si le document demandé peut être caché ou pas.
- ✓ **If-Unmodified-Since**
 - condition de retrait
- ✓ ...

Entêtes de Requêtes Client (3)

- ✓ **Max-Forwards**
 - nombre max de proxy
- ✓ **Proxy-Authorization**
 - identification
- ✓ **Range**
 - zone du document à renvoyer
 - bytes=x-y (x=0 correspond au premier octet, y peut être omis pour spécifier jusqu'à la fin)
- ✓ **Referer**
 - URL d'origine
 - page contenant l'ancre à partir de laquelle le visualisateur a trouvé l'URL.
- ✓ **User-Agent**
 - modèle du visualisateur

Entêtes de Réponses Serveur

- ✓ **Accept-Range**
 - accepte ou refus d'une requête par intervalle
- ✓ **Age**
 - ancienneté du document en secondes
- ✓ **Proxy-Authenticate**
 - système d'authentification du proxy
- ✓ **Public**
 - liste de méthodes non standards gérées par le serveur
- ✓ **Retry-After**
 - date ou nombre de secondes pour un ressay en cas de code 503 (service unavailable)
- ✓ **Server**
 - modèle de HTTPD
 - utilisé par Satan !!!!
- ✓ **Set-Cookie**
 - crée ou modifie un cookie sur le client
- ✓ **WWW-Authenticate**
 - système d'authentification pour l'URI

Entêtes d'Entité (1)

- ✓ **Allow**
 - méthodes autorisées pour l'URI
- ✓ **Content-Base**
 - URI de base
 - pour la résolution des URL
- ✓ **Last-Modified**
 - date de dernière modification du doc.
 - Utilisé par les caches
- ✓ **Content-Length**
 - taille du document en octet
 - utilisé par le client pour gauger la progression des chargements
- ✓ **Content-Encoding**
 - type encodage du document renvoyé
 - compress, x-gzip, x-zip
- ✓ **Content-Language**
 - le langage du document retourné
 - fr, en ...
- ✓ ...

Entêtes d'Entité (2)

- ✓ **Content-MD5**
 - résumé MD5 de l'entité
- ✓ **Content-Range**
 - position du corps partiel dans l'entité
 - bytes x-y/taille
- ✓ **Content-Transfer-Encoding :**
 - transformation appliqué du corps de l'entité
 - 7bit, binary, base64, quoted-printable
- ✓ **Content-Type**
 - type MIME du document renvoyé
 - utilisé par le client pour sélectionner le visualisateur (plugin)
- ✓ **Etag**
 - transformation appliqué du corps de l'entité
 - 7bit, binary, base64, quoted-printable

Entêtes d'Entité (3)

- ✓ **Expires**
 - date de péremption de l'entité
- ✓ **Last-Modified**
 - date de la dernière modification de l'entité
- ✓ **Location**
 - URI de l'entité
 - quand l'URI est à plusieurs endroits
- ✓ **URI**
 - nouvelle position de l'entité
- ✓ ...

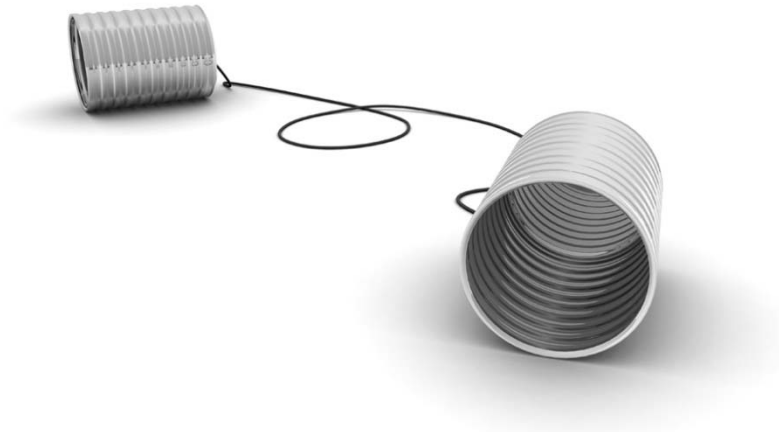
Internationalisation

✓ Langage Accepté

- fr, de, it, en, sq (albanais), ru, (russe), ja (japonais), zh (chinois), el (grec), he (hébreu), ca (catalan) ...

✓ Charset (table de caractère)

- par défaut ISO-8859-1 (Latin-1)
 - ISO-8859-2 (hongrois, albanais, ...)
 - ISO-8859- 4
 - ISO-8859-5, KOI8-R (russe, bulgare, polonais)
 - ISO-8859-7 (grec)
 - ISO-8859-8 (hébreu)
 - ISO-8859-9 (turc)
 - Shift_JIS, ISO-2022-JP, EUC-JP (japonais)
 - Big5 (chinois simplifié)
 - GB2312(chinois traditionnel - Taiwan)



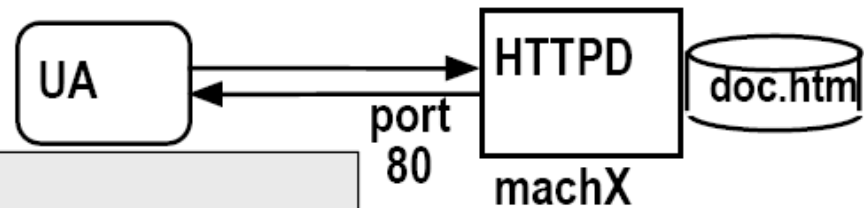
Echange de Documents

Réception et Envoi de Données

Récupération d'un Document Méthode GET

✓ GET /fichier

GET /doc.htm



le Client envoie

```
GET /doc.htm HTTP/1.0      méthode,chemin,version
Accept: www/source        documents acceptés
Accept: text/html
Accept: image/gif
User-Agent: Lynx/2.2 libwww/2.14
From: alice@pays.merveilles.net
* une ligne blanche *
```

le Serveur répond

```
HTTP/1.0 200 OK      ligne de status
Date: Wed, 02Feb97 23:04:12 GMT
Server: NCSA/1.1
MIME-version: 1.0
Last-modified: Mon,15Nov96 23:33:16 GMT
Content-type: text/html      type du document retourné
Content-length: 2345        sa taille
* une ligne blanche *
<HTML><HEAD><TITLE> ...
```

Méthode GET conditionnelle

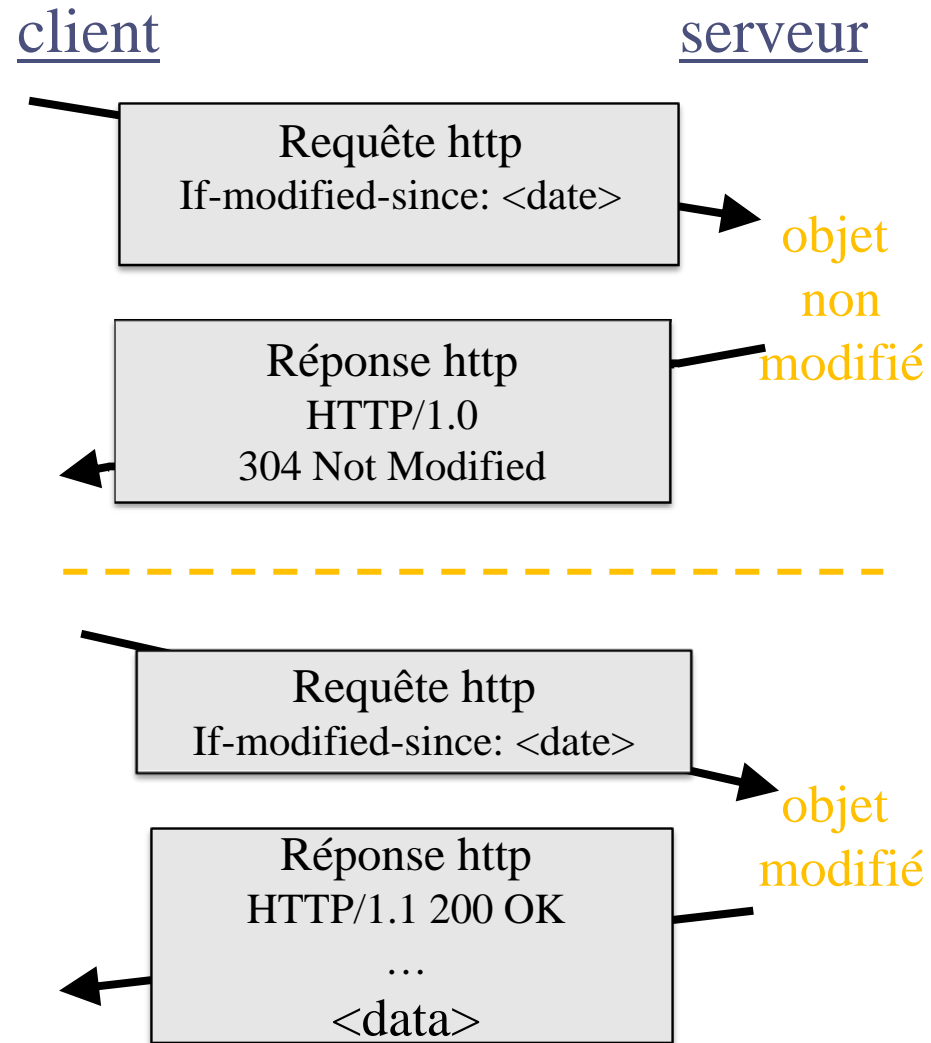
✓ **Objectif** : ne pas envoyer d'objet si le client à une version chargée à jour (en cache).

✓ **Client**: spécifie la date de la copie en cache dans la requête :

`If-modified-since: <date>`

✓ **Serveur**: la réponse ne contient pas de données si l'objet est à jour :

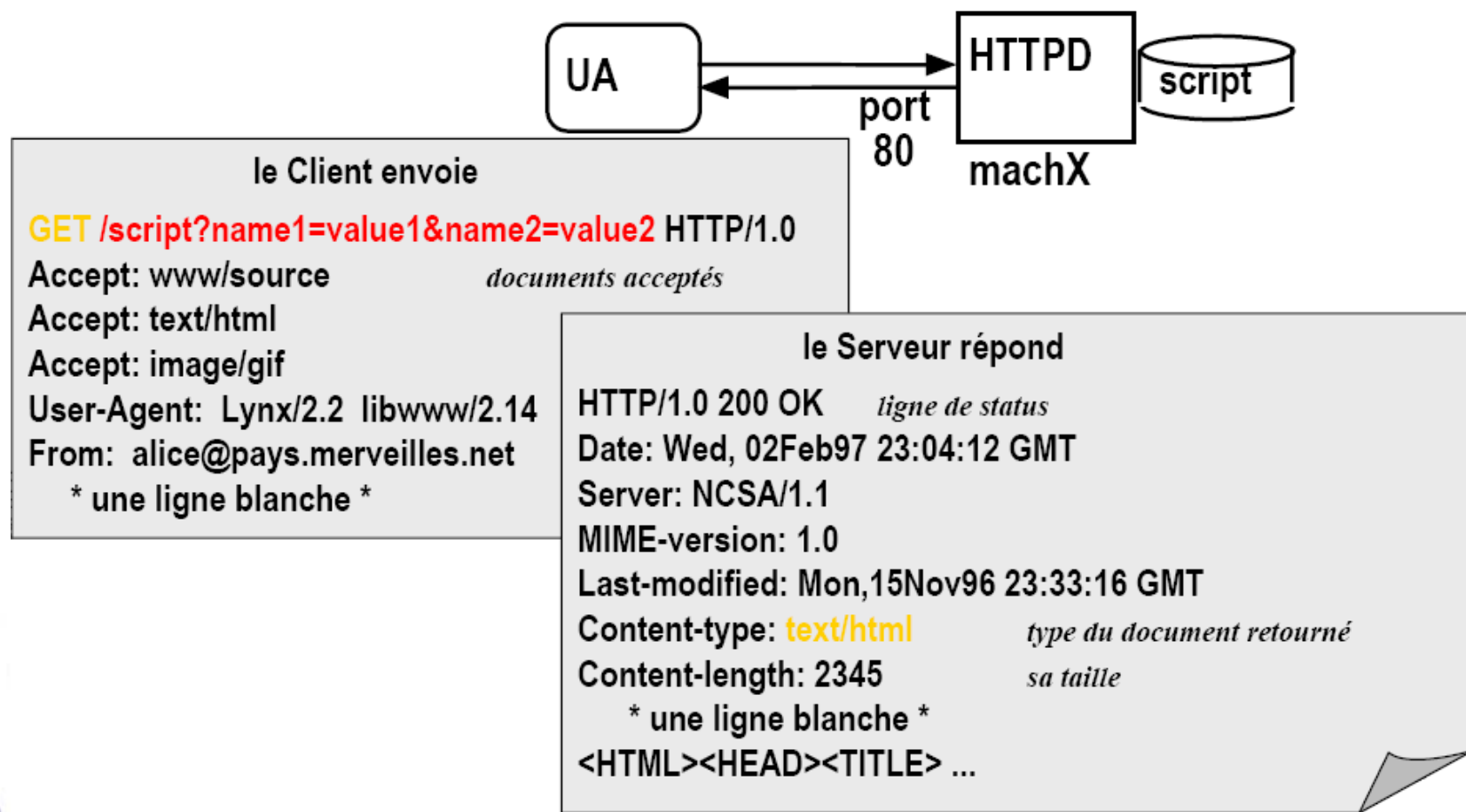
`HTTP/1.0 304 Not Modified`



Soumission d'un Formulaire

Méthode GET

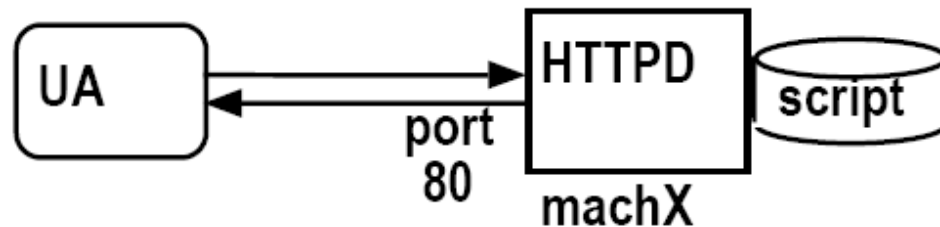
- ✓ GET/script?name1=value1&name2=value2
GET /script?name1=value1&name2=value2



Soumission d'un Formulaire méthode POST

✓ POST /script

POST /script



le Client envoie

```
POST /script HTTP/1.0
Accept: www/source
Accept: text/html
Accept: image/gif
User-Agent: Lynx/2.2 libwww/2.14
From: alice@pays.merveilles.net
  * une ligne blanche *
name1=value1&
name2=value2
```

le Serveur répond

```
HTTP/1.0 200 OK
...
Content-length: 2345
  * une ligne blanche *
<HTML><HEAD><TITLE> ...
```

Codage des « paramètres »

- ✓ Les valeurs passées (URL et contenu des entrées des formulaires) doivent être sur 7 bits et sans caractères spéciaux
- ✓ Format d'encodage : x-www-form-urlencoded
 - Espace ⇒ « + »
 - Tous les caractères spéciaux et accentués ⇒ % code ascii
 - @ %40
 - é %e9
 - Les entrées des formulaires sont encodés dans une chaîne composée de paires (nom de l'entrée)=(valeur de l'entrée) séparé par des &
- ✓ `nom=Dupont+Jean&adresse=3+rue+de+la+Gait%e9%0a75014+Paris`

Comportement du Client / type du document retourné

- ✓ **A partir du type MIME de Content-Type**
 - **Visualisation native**
 - la fonction de visualisation est dans le noyau (core) du client
 - `text/html`, `image/jpeg`
 - **Visualisation par plugin**
 - la fonction est présente dans une DLL, un SO ou un JAR
 - elle est liée dynamiquement pour réaliser la visualisation
 - `world/vrml`, `text/tex`
 - **Visualisation externe**
 - la fonction n'est pas présente dans le client
 - le client rapporte le document et le sauvegarde dans un fichier temporaire
 - `video/mpeg`, `application/postscript`

Requête Multi-parties (multipart)

✓ Motivation

- Requête multi-document [RFC1867]
- formulaire HTML contenant des Upload de fichiers
 - `<FORM ACTION="/servlet/UploadTest" ENCTYPE="multipart/form-data" METHOD=POST>`
 - `Your name? <INPUT TYPE=TEXT NAME=submitter>
`
 - `Your first file to upload? <INPUT TYPE=FILE NAME=file1>
`
 - `Your second file to upload? <INPUT TYPE=FILE NAME=file2>
`
 - `<INPUT TYPE=SUBMIT>`
 - `</FORM>`
- Remarque : Mail multi-documents
 - (fichiers attachés, mail enrichi d'images, audio-mail ...)

Requête Multi-parties (multipart) Codage de la requête

```
Content-Type : multipart/form;boundary=End9989822  
--End9989822
```

```
Content-Disposition; form-data;name="file1";  
filename="test.htm"  
Content-Type : text/html  
<HTML><BODY> Ceci est un fichier de  
test !</BODY></HTML>  
--End9989822
```

```
Content-Disposition; form-data; name="file2";  
filename="test2.txt"  
Content-Type : text/plain  
Ceci est un deuxième fichier de test !  
--End9989822
```

Réponse Multi-parties

✓ Codage

- Content-Type : multipart/x-mixed-replace;
- Frontière entre les parties
 - Déclaration : boundary=chaîne_aléatoire
 - Séparateur : -chaîne_aléatoire

✓ Comportement

- le navigateur affiche le sous-document suivant dès qu'il commence à le recevoir après avoir effacer la fenêtre.

Réponse Multi-parties

Exemple

```
Content-Type : multipart/x-mixed-replace;  
boundary=End65577565679001838
```

Le serveur définit une chaîne séparateur des documents

```
--End65577565679001838
```

```
Content-Type : text/html
```

```
<HTML><BODY><H1>Trois ... </H1><BODY></HTML>
```

Le serveur attend 1 seconde avant de renvoyer la suite : le client affiche « Trois... »

```
--End65577565679001838
```

```
Content-Type : text/html
```

```
<HTML><BODY><H1>Deux ... </H1><BODY></HTML>
```

Le serveur attend 1 seconde avant de renvoyer la suite : le client affiche « Deux... »

```
--End65577565679001838
```

```
Content-Type : text/html
```

```
<HTML><BODY><H1>Un ... </H1><BODY></HTML>
```

Le serveur attend 1 seconde avant de renvoyer la suite : le client affiche « Un... »

```
--End65577565679001838
```

```
Content-Type : text/html
```

```
<HTML><BODY><H1>Partez ! </H1><BODY></HTML>
```

Le serveur clôt la connexion TCP/IP avant de renvoyer la suite : le client affiche finalement « Partez ! »

```
--End65577565679001838
```



Sessions avec HTTP

Comment réaliser le suivi de sessions ?

Suivi de Sessions avec HTTP (1)

✓ Motivations :

- La notion de session est importante dans une application conversationnelle
 - commerce électronique
 - « j ’ajoute ce produit à mon panier (existant)»
- Cependant HTTP est un protocole «stateless»
 - le serveur ne maintient pas d ’informations liées aux requêtes précédentes d ’un même client.
 - HTTP est donc « sessionless »
- Comment implanter la notion de session sur plusieurs requêtes HTTP
 - documents, CGI, Servlet, ASP

Suivi de Sessions avec HTTP (2)

✓ Méthodes

- Le serveur génère un identificateur de session et associe un état (et une date limite de validité) à une session
- Le client renvoie l'identificateur de session à chaque requête HTTP vers le serveur

✓ Echange et Stockage de l'identificateur de session

- Input HIDDEN dans les formulaires
- Réécriture des URLs EXTRA_PATH
- Cookies (désactivable)
- Identificateur de session SSL (Secure Socket Layer)

Suivi de Sessions avec HTTP (3)

- ✓ **Une session s'étend sur plusieurs requêtes**
 - documents, CGI, SSS, Servlet, ASP
 - le serveur maintient un contexte de session et y associe un identifiant de session

- ✓ **3 solutions de suivi**
 - input HIDDEN
 - contient l'identifiant de la session
 - la Ré-écriture d'URL
 - l'identifiant dans chaque URL (dans les documents)
 - les Cookies
 - information positionnée par le serveur sur le client la durée de vie du cookie dépasse la session
 - puis envoyée par le client à chaque requête

une entrée HIDDEN par formulaire

- ✓ Chaque réponse retournée par le serveur est un formulaire qui contient un identifiant caché dans une entrée HIDDEN
- ✓ Exemple
 - page de proposition

```
<FORM METHOD="POST" ACTION="/cgi-bin/command">  
<INPUT TYPE="checkbox" NAME="art12387">  
Chaussures  
...  
</FORM>
```

- réponse de /cgi-bin/command

```
<FORM METHOD="POST" ACTION="/cgi-bin/envoi">  
<INPUT TYPE="hidden" NAME="TransID" VALUE="54109848932">  
Nom: <INPUT TYPE="text" NAME="nom">  
Adresse: <INPUT TYPE="text" NAME="adresse">  
N° de Carte de Credit: <INPUT TYPE="text" NAME="numcarte">...  
<INPUT TYPE="hidden" NAME="Language" VALUE="French">  
</FORM>
```

une entrée HIDDEN par formulaire

✓ Inconvénients

- Dialogue uniquement par formulaire
 - car pas de persistance de l'identifiant côté client
- Ambiguïté dans le cas des retours-arrière de l'utilisateur
 - annulation d'une série d'actions
 - ou série d'actions supplémentaires

Suivi de Session

la ré-écriture des URLs

- ✓ L'identifiant de sessions est encodé dans les URLs des documents HTML retournés par le serveur.
 - Dans le PATH
 - `http://www.mycomp.com/cgi-bin/envoi?name=toto`
 - devient
 - `http://www.mycomp.com/182993954/cgi-bin/envoi?name=toto`
 - Dans l'EXTRA-PATH
 - `http://www.mycomp.com/cgi-bin/ envoi?name=toto`
- ✓ devient
 - `http://www.mycomp.com/cgi-bin/envoi/sid$182993954?name=toto`

Suivi de Session

la ré-écriture des URLs

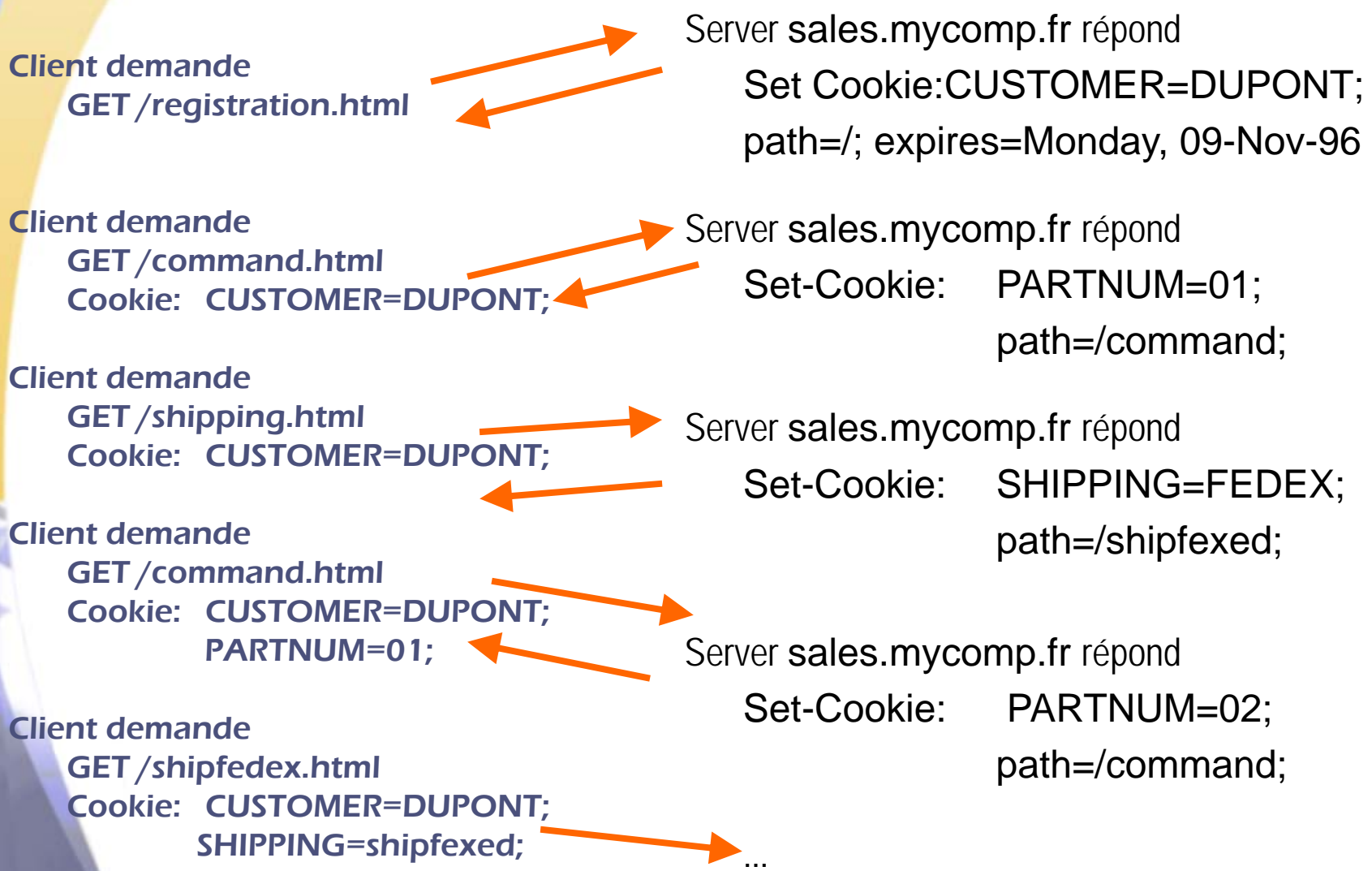
✓ Limites

- URL générée par un script
 - (=> programmation)
- ou parsing des documents HTML retournés
 - mais disfonctionnement en présence de scripts JavaScript ou VBScript générant eux aussi des URL !

les Cookies [Netscape puis RFC2109]

- ✓ Chaîne décrivant l'état d'une session
 - NAME=VALUE;
 - expires=DATE;
 - path=PATH_HEAD; /<</foo<</foobar ou /foo/bar.html
 - domain=DOMAIN_TAIL; fr<<mycomp.fr << sales.mycomp.fr
- ✓ Stocké sur le client
 - Limite :
 - 300 cookies simultanées par client, 20 cookies par serveur ou domaine, 4Ko par cookie (limite la taille des VALUES)
- ✓ Communiqué dans les entêtes de requêtes et dans les entêtes des réponses HTTP
- ✓ Accessible par les scripts JavaScript dans une page HTML

Positionnement des Cookies



L'évolution des Cookies (1)

- ✓ Les cookies menacent la vie privée (privacy) des cybernauts bien qu'ils soient très utiles
- ✓ Les navigateurs peuvent désactiver les cookies
- ✓ Un remplaçant : P3P (Platform for Privacy Preferences) en vue d'un accord juridique entre le client et le site sur
 - la définition du champs des divulgation
 - ex : nom, prénom, adresse mais pas l'âge ou le nombre d'enfants
 - la définition de l'utilisation de ces données par le propriétaire du site
 - ex : cession des informations à des tiers
 - la définition de la procédure de modification des données ultérieurement
 - ex: je me suis marié

L'évolution des Cookies (2)

- TUID/PUID Temporary et Pairwise Unique ID
 - identifiants de session (et multi-sessions) sans information attachée
- P3P exprimé en RDF/XML, Certificats / Signatures



Authentification avec HTTP

L'authentification dans HTTP

- ✓ Indiqué dans les ACL
- ✓ Modes d'authentification
 - BASIC
 - nom d'utilisateur et mot de passe échangé en clair (base64) !
 - base des mots de passe dans un fichier htpasswd utilitaires de gestion du fichier
 - DIGEST
 - sécurisation de BASIC
 - hachage sécurisé MD5 du (nom,password,URI, méthode,nombre aléatoire fourni par le serveur)
 - SSL
 - Secure Socket Layer (TLS : Transport Layer Security)
 - authentification avec CA du serveur (2.0) et du client (3.0)
 - confidentialité avec DES
 - puis dialogue HTTP sur la connexion SSL

L'authentification applicative

- ✓ **Motivations**
 - interface de login
 - identification externe
 - BD, Annuaire LDAP, ...
 - authentification plus forte
- ✓ **L'application gère l'authentification de l'utilisateur**
 - formulaire d'accueil HTML (nom, password)
 - attention le mot de passe peut-être est en clair
 - gestion des tables d'utilisateur
 - une session est ensuite ouverte associé à un utilisateur authentifié (ou non : par exemple rejet au bout de 3 tentatives)

Contrôle d'Accès dans HTTP

- ✓ **ACL (Access Control List)**
 - spécifie les autorisations (**ALLOW**) ou les interdictions (**DENY**) d'accès à une arborescence virtuelle du serveur
 - en fonction :
 - de l'authentification
 - de la localisation du client sous domaine DNS, réseau ou adresse IP
- ✓ **ACF (Access Control File)**
 - fichier regroupant les ACL
 - global : `access.conf` dans Apache
 - par arborescence : `.htaccess`
 - combinaison des **ALLOW** et des **DENY**

Audit des Requêtes

- ✓ **Journaux des requêtes**
 - les accès (access.log, refferee.log), et les erreurs (error.log),... sont journalisés
- ✓ **Exploitation des Journaux**
 - erreur dans les liens, ...
 - clientèle, analyse d'activité, ...
- ✓ **Reporting (Présentation Synthétique)**
 - Pour Apache
 - AccessWatch, Wusage, Analog, wwwstat
 - IIS, NS
 - intégré et visualisé par un script
 - Généraux
 - Net Analysis (Net Genesis), Enterprise Suite (Web Trends)

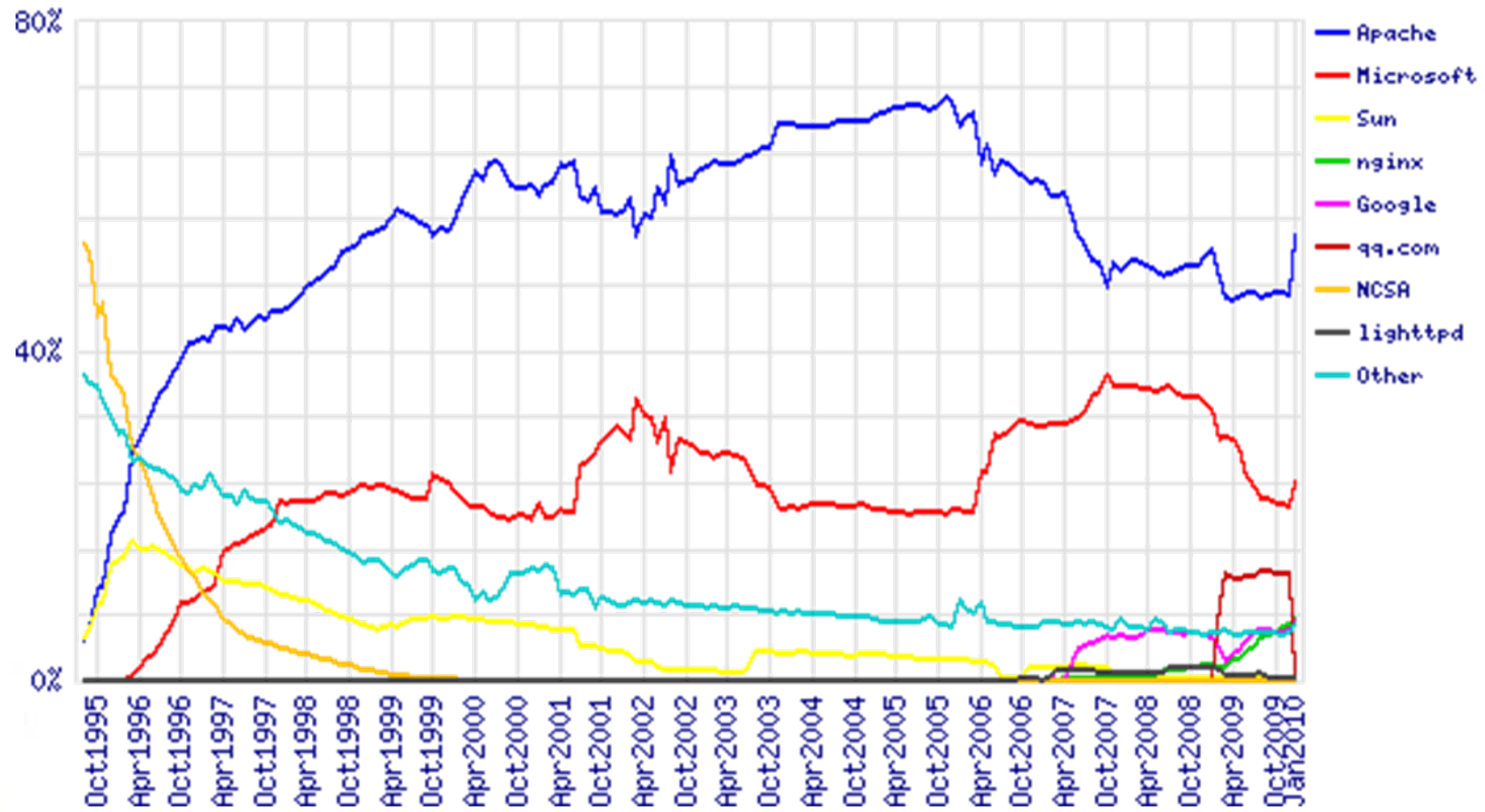


Infrastructure HTTP

Les Serveurs du Marché

- ✓ **Offre très large**
 - Apache HTTPD
 - Netscape Enterprise Server
 - Microsoft Internet Information Server
 - W3C Jigsaw
 - Sun JavaServer
 - Oracle Web Server
 - IBM Web Sphere
 - ...etc...
- ✓ **Fonctionnalités supplémentaires**
 - gestion des sessions, des transactions, ...
 - accès aux serveurs d'applications, ...

Répartition part de Marché Serveurs Web



Source: <http://news.netcraft.com/>

✓ A patch of NCSA HTTPD

- serveur le plus répandu (« toujours » plus de 50% de part de marché)
- gratuit, issu du serveur NCSA HTTPD
- très nombreuses plates-formes Unix et Windows NT
- extensible par des modules tiers

✓ Nombreux Modules Tiers

- possibilité d'étendre Apache avec des modules externe
http://www.zyzzzyva.com/server/module_registry
 - `mod_auth_cookies_file`, `mod_auth_cookies_mysql`, `mod_cgi_sugid`,
`mod_perl`, `mod_perl_fast`, `mod_auth_kerb`, `mod_auth_dbi`,
`mod_rewrite`, `mod_jserv(servlet)`, `mod_java` (CGI écrit en Java),
`php3`
- nombreux sous-projets autour de Java (Jakarta) et XML
(Xerces, Xalan, XSP, Cocoon, ...)

Configuration Apache

- ✓ **Fichiers de configuration**
 - **httpd.conf**
 - comportement de base port TCP/IP, journaux, keepalive, UID, virtualhost, proxy, ...
 - **srm.conf**
 - traitement des ressources locales lors des requêtes index, script, répertoire, AddType, AddIcon, Alias, DocumentRoot
 - **access.conf**
 - contrôle d'accès global (ACF : access config file)
 - **mime.types**
 - table de correspondance suffixe fichier -> type MIME document
- ✓ **Outil**
 - **GUI : Vision (focus-array.com)**
 - ...

2. Quelques Manipulations

✓ 1. Utilisation de Telnet pour contacter un serveur Web :

```
telnet www.unice.fr 80
```

Ouvre une connexion sur le port 80 (port par défaut) de www.unice.fr

Tout ce qui est tapé est maintenant transmis au serveur sur le port 80

2. Envoi d'une requête GET

```
GET /index.html HTTP/1.0
```

En tapant ceci, vous envoyez cette requête GET, minimale mais complète au serveur http (suivi de 2 « retour chariot »).

3. Récupération de la réponse du serveur Web