

Abdelahad SATOUR

# Web Historiquement

**Ma citation**

**Il n'y a pas une question banale!**



**Le web en résumé**



- **Internet**

- Internet est un réseau international de communication qui relie des réseaux d'ordinateurs entre eux.



## Le web en résumé



### • Histoire d'Internet

- 1945 : Internet, un projet de science-fiction.
- En 1945 Vannevar Bush universitaire américain anticipe Internet
- Article As We May Think (1945) magazine Atlantic Monthly.

Il décrit un dispositif hypermedia avant la lettre qu'il appelle Memex ("memory extender") et qui devrait permettre aux chercheurs d'accéder plus facilement à la documentation scientifique.



## Le web en résumé



### • La guerre froide

- En 1962, l'US Air Force demande la création d'un réseau de communication militaire capable de résister à une attaque nucléaire. En 1945 Vannevar Bush universitaire américain anticipe Internet
- Notions de routage
- Notions du chemin min avec coût max
- Packet switching : TCP



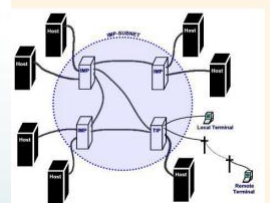


## Le web en résumé



### • L'ARPANET

- En août 1969, hors de tout objectif militaire, le réseau expérimental ARPANET est créé afin de relier quatre instituts universitaires.
- Ce réseau est aujourd'hui considéré comme le précurseur. La communication entre machines se faisait sans machine centralisée intermédiaire.
- Les protocoles utilisés étaient basiques.



## Le web en résumé



### • Le réseau, comment ça marche ?

- TCP/IP «Transmission Control Protocol/Internet Protocol» est un système gère l'ensemble des règles de communication sur internet et se base sur la notion adressage IP
- c'est-à-dire le fait de fournir une adresse à chaque machine du réseau afin de pouvoir acheminer des paquets de données
- elle est conçue pour répondre à un certain nombre de critères parmi lesquels :
  - le fractionnement des messages en paquets l'utilisation d'un système d'adresses ;
  - l'acheminement des données sur le réseau (routage)
  - le contrôle des erreurs de transmission de données

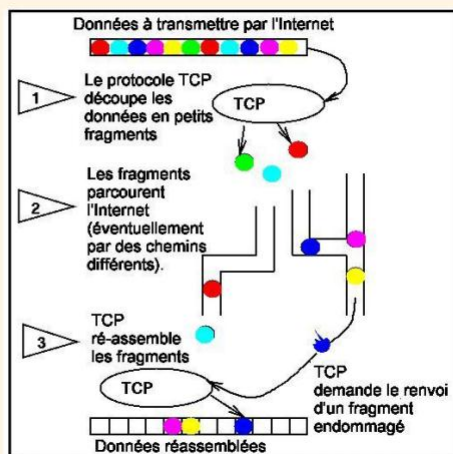


## Le web en résumé



- Le réseau, comment ça marche ?

### TCP



## Le web en résumé



- Le World Wide Web

- Dès 1980, Tim Berners-Lee, un chercheur au CERN de Genève, met au point un système de navigation hypertexte et développe, avec l'aide de Robert Cailliau, un logiciel baptisé Enquire permettant de naviguer selon ce principe.
- Fin 1990, Tim Berners-Lee met au point le protocole HTTP (Hyper Text Transfer Protocol), ainsi que le langage HTML (HyperText Markup Language) permettant de naviguer à l'aide de liens hypertextes, à travers les réseaux.
- Le World Wide Web est né.





## Le web en résumé



- La toile

Le World Wide Web, littéralement la « toile (d'araignée) mondiale », communément appelé le Web, est un système hypertexte public fonctionnant sur Internet et qui permet de consulter, avec un navigateur, des pages mises en ligne dans des sites.

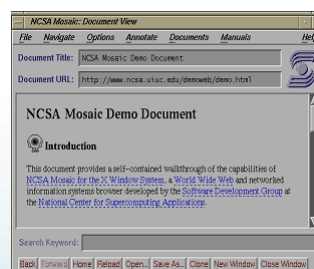


## Le web en résumé



- Les débuts

- 1993, aux États-Unis, le gouvernement Clinton donne le coup d'envoi officiel du programme National Information Infrastructure.



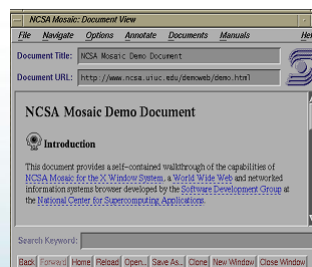


## Le web en résumé



### • Les débuts

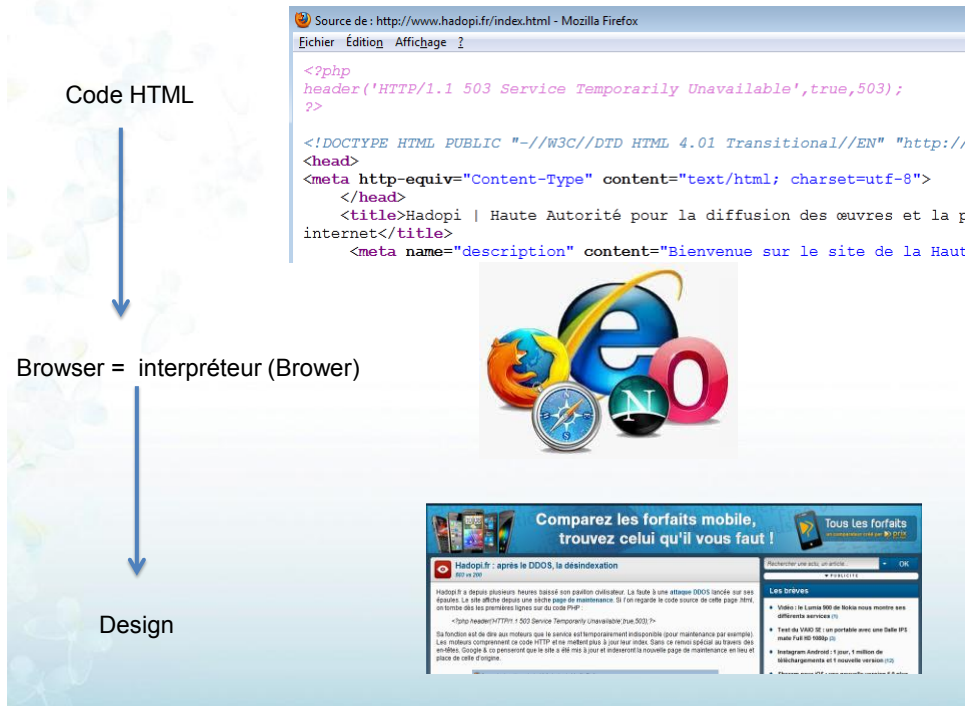
- Le navigateur Mosaic version 1.0 sort et en 1996, le terme « Internet » est utilisé par le grand public. Le Web était bien plus qu'une « autoroute de l'information », il avait la capacité de donner accès à des niveaux supérieurs d'information, classés de manière intelligente et enrichies



## Notions préliminaires

### • Internet et les pages web

- HTML : conception de pages destinées à être publiées sur Internet
- Page html : contient le texte à afficher et des instructions de mise en page
- HTML est un langage de description de page et non pas un langage de programmation
  - pas d'instructions de calcul ou pour faire des traitements suivant des conditions
- Des sites de plus en plus riches en informations
  - Nécessité croissante d'améliorer le contenu de sites
  - Mises à jour manuelles trop complexes
    - Pourquoi ne pas automatiser les mises à jour ?





## Notions préliminaires

- Pages web statiques : fonctionnement

- Leurs contenus ne changent ni en fonction du demandeur ni en fonction d'autres paramètres éventuellement inclus dans la requête adressée au serveur. Toujours le même résultat.
- Rôle du serveur : localiser le fichier correspondant au document demandé et répond au navigateur en lui envoyant le contenu de ce fichier

- Pages web statiques : limites

- Besoin de réponses spécifiques : passage de pages statiques à pages dynamiques



## Notions préliminaires



- Apache (C'est le serveur HTTP )  
Apache HTTP Server

### Index of /articles/apache

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">Parent Directory</a>	13-Jan-2006 15:10	-	
<a href="#">dossier1/</a>	13-Jan-2006 15:10	-	
<a href="#">fichier.avi</a>	13-Jan-2006 15:10	0k	
<a href="#">fichier.bat</a>	13-Jan-2006 15:10	0k	
<a href="#">fichier.jpg</a>	13-Jan-2006 15:10	0k	
<a href="#">fichier.pdf</a>	13-Jan-2006 15:10	0k	
<a href="#">fichier.txt</a>	13-Jan-2006 15:10	0k	
<a href="#">fichier.zip</a>	13-Jan-2006 15:10	0k	





## Notions préliminaires



- ALLEZ TP

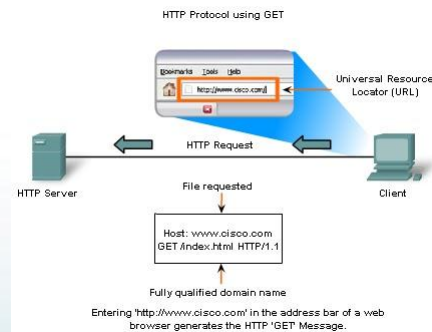


## Comment HttpD



### Comment?

- Browser -> lecture analyse du protocole.



Browser?  
MIME?  
Encoding?  
TimeZone?  
IP?

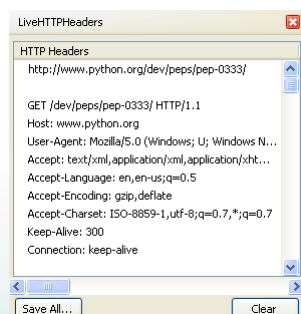


## Comment HttpD



### Comment?

- Création et l'envoi du packet HTTP à la carte réseau du système.



Open Firebug



## Comment HttpD

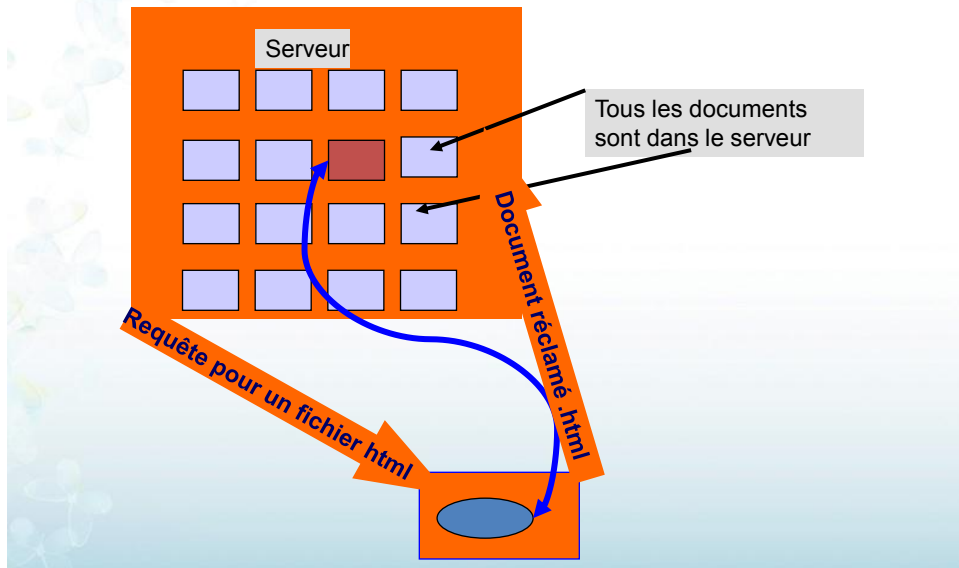


### Comment?

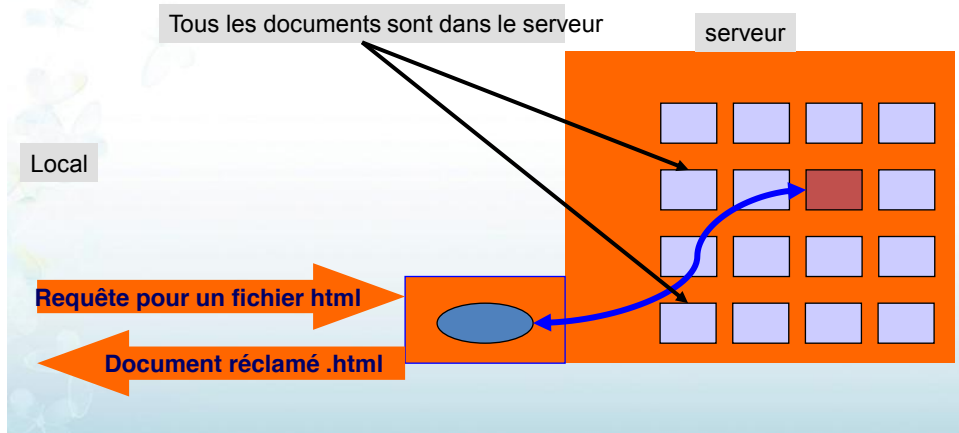
- System32 Analyse l'adresse pour la construction du packet TCP/IP. (Il analyse la table interne d'intranet et localhost etc/host)
- Il s'avère que l'adresse du site est une adresse interne(comment?)
- La carte réseau construit une trame tcp ip de la couche HTTP à l'adresse Localhost (dans le cas externe adresse externe)
- Un LoopUp
- La carte réseau lit la trame TCP/IP, il se trouve que le protocole en question est le protocole HTTP (comment? Le port 80).
- Elle sépare les données utilisées (http)-> System 32 -> Apache



## Comment HttpD

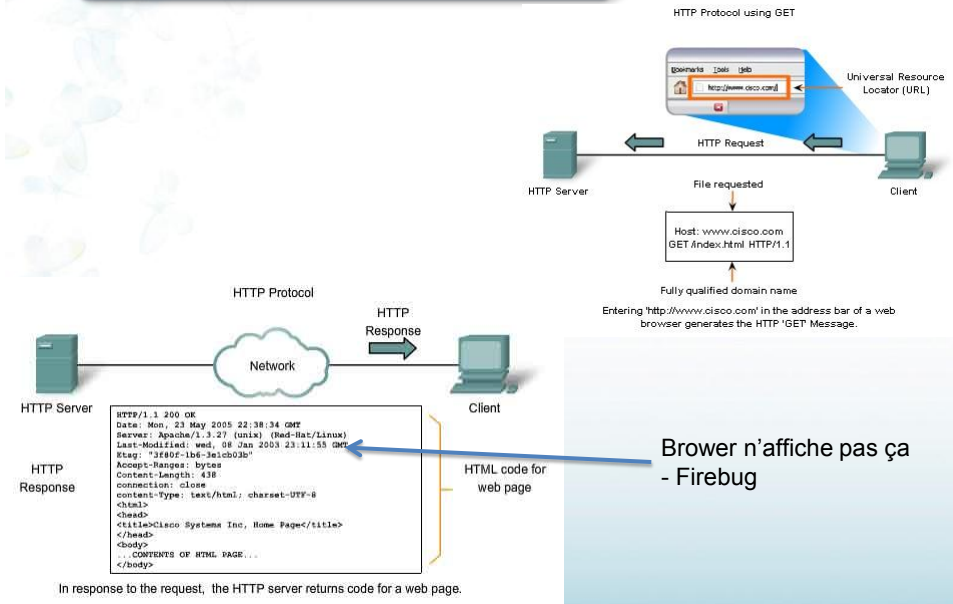


## Notions préliminaires





## Notions préliminaires



## Comment HttpD



### Comment?

- Apache -> System -> carte réseau
- > Local machine->carte réseau->Browser->Browser



## Notions préliminaires

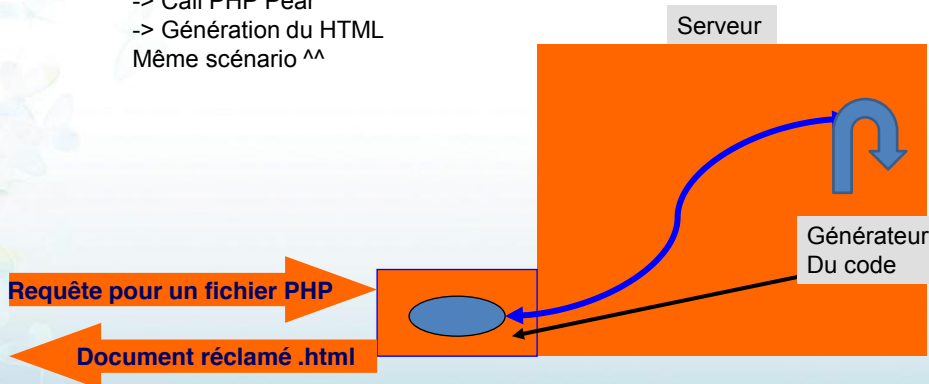
### • Les langages de script-serveur : Définition

- Un langage de script **-serveur** est :
  - un programme stocké sur un serveur et exécuté par celui-ci,
  - qui passe en revue les lignes d'un fichier source pour en modifier une partie du contenu,
  - avant de renvoyer à l'appelant ( un navigateur par exemple) le résultat du traitement.
- La tâche d'interprétation des ordres à exécuter est déléguée à un composant, souvent appelé moteur,
  - installé sur le serveur,
  - qui est doté d'une API et d'un fonctionnement identique quel que soit la plate-forme utilisée pour gérer le serveur



## Notions préliminaires

Apache identifie qu'il le fichier d'entrée est un fichier PHP  
 -> Call PHP Pear  
 -> Génération du HTML  
 Même scénario ^^





## Notions préliminaires

- Les langages de création de pages web dynamiques côté serveur
  - Les CGI
    - Sont des composants exécutables (fichiers .exe ou .dll) qui produisent sur le serveur des contenus html à envoyer aux clients.
    - Les CGI sont compilés. Ils sont rapides mais fortement liés à la plate-forme sur laquelle ils tournent.
  - PERL
    - Surcharge rapide du serveur par la création de plusieurs processus
    - Employé sur de nombreux serveurs. Il tourne sur de nombreuses plateformes : Unix, Linux, Windows, Mac
    - Prévu à l'origine pour la manipulation de chaînes de caractères, il est rapidement devenu un véritable langage orienté objet.
    - Abord difficile et faible lisibilité.



## Notions préliminaires

- Les langages de création de page web dynamiques côté serveur
  - PHP
    - Connaît un succès toujours croissant sur le Web et se positionne comme un rival important pour ASP
    - L'environnement Linux est sa plateforme de prédilection
    - Combiné avec le serveur Web Apache et la base de données MySQL, PHP offre une solution particulièrement robuste, stable et efficace
    - Gratuité : Tous les logiciels sont issus du monde des logiciels libres (Open Source).



## A l'attaque(BANZAI!!!!)

Extension Html -> PHP

lecture pas de balise PHP-> retourner le même code HTML

*RQ: PowerBy Firebug*



## Les standards

- Les standards un Pourquoi

- Les standards du Web est une expression désignant différentes technologies et protocoles utilisés sur le Web et en particulier ceux définis par le W3C sous forme de recommandations.
- Cette expression a été popularisée à partir de la fin des années 1990, à travers des ouvrages de designers tels que Designing With Web Standards de Jeffrey Zeldman<sup>1</sup>, ainsi que l'action d'associations de professionnels du Web comme le Web Standards Project (WaSP), ou encore en France le collectif Openweb<sup>2</sup>.
- Elle se réfère principalement aux technologies formant le socle principal d'un document web : le HTML et le XHTML, les feuilles de style en cascade (CSS) et le DOM (Document Object Model). Cependant, elle peut également, selon les contextes, s'étendre à un ensemble plus vaste, dont :



## Les standards

- Les standards un Pourquoi

- [HTTP](#) (*Hypertext Transfer Protocol*)
- [PNG](#) (*Portable Network Graphics*)
- [RDF](#) (*Resource Description Framework*)
- [SVG](#) (*Scalable Vector Graphics*)
- [SMIL](#) (*Synchronized Multimedia Integration Language*)
- [URI](#) (*Uniform Resource Identifier*)
- [XML](#) (*Extensible Markup Language*)
- [WCAG](#) (*Web Content Accessibility Guidelines*)



## Les standards

- Les Organismes

Il existe de nombreux organismes de standardisation, d'importances diverses. Parmi les plus notables se trouvent :

- l'IETF (protocoles...)
- le W3C (langages de balisage, langages de présentation, accessibilité, respect de la vie privée...)
- l'ISO et le Consortium Unicode (codages de caractères...)
- l'ECMA (JavaScript...)
- l'IANA (noms de domaine, types MIME...)...
- ...





## Evolution du web

- Navigateur et technologies

<http://www.evolutionoftheweb.com/?hl=fr#/evolution/day>



## Les Protocoles

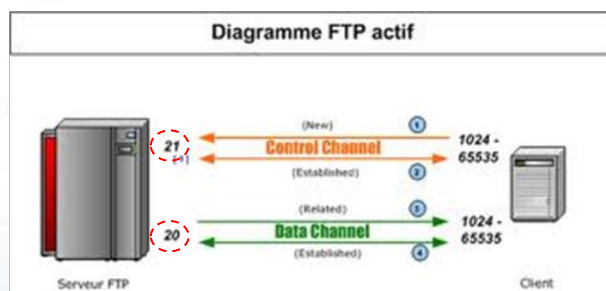
## 1. FTP

- FTP est l'acronyme de (*File Transfer Protocol*), soit une méthode de transfert de fichiers.
- La mise en place du protocole FTP date de 1971.
- Tous les fournisseurs d'accès d'Internet proposent un tel service.
- Lorsque on veut transmettre de gros fichiers, le plus simple est d'utiliser un logiciel de FTP.

## 1. FTP

- Le protocole FTP a pour objectifs de :
  - permettre un partage de fichiers entre machines distantes
  - permettre une indépendance aux systèmes de fichiers des machines clientes et serveur
  - permettre de transférer des données de manière efficace

## 1. FTP



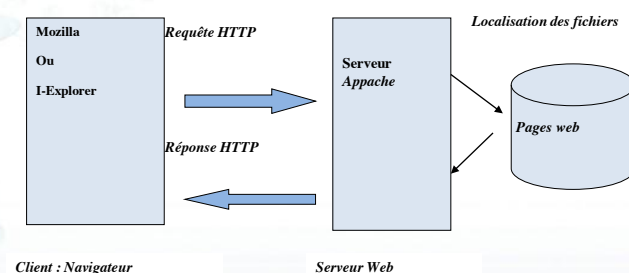
## 1.FTP

- Lorsque on est connecté en mode, on peut agir sur le disque dur **éloigné** comme sur le local et donc renommer ou supprimer des fichiers.
- **Utilité:** pour placer les fichiers d'un site Web sur le disque dur d'un fournisseur d'accès ou pour échanger de gros fichiers avec un autre utilisateur, l'usage d'un FTP est extrêmement pratique.
- Plusieurs logiciels FTP gratuits peuvent être télécharger de l'internet tel que **FileZilla** .

## 2. HTTP

- Le protocole HTTP (HyperText Transfer Protocol) est le protocole le plus utilisé sur Internet depuis 1990.
- Destiné à transférer des données sur Internet (en particulier des pages Web écrites en HTML).
- Quelle est la différence avec FTP?
- Les données sont localisées grâce à une chaîne de caractères appelée **URL** (Uniform Ressource Locator) entre un navigateur (le client) et un serveur Web.

## 2. HTTP



La communication entre le navigateur et le serveur se fait en deux temps :

- Le navigateur effectue une **requête HTTP**
- Le serveur traite la requête puis envoie une **réponse HTTP**

## 2.HTTP

- Une requête HTTP est destinée à une adresse de la forme (URL)  

www. Machine. Domaine
- Exemple: [www.google.com](http://www.google.com)
- Cette adresse doit être spécifiée dans la barre adresse du navigateur

Commandes

Commande	Description
GET	Requête de la ressource située à l'URL spécifiée
HEAD	Requête de l'en-tête de la ressource située à l'URL spécifiée
POST	Envoi de données au programme situé à l'URL spécifiée
PUT	Envoi de données à l'URL spécifiée
DELETE	Suppression de la ressource située à l'URL spécifiée

## En-têtes

Nom de l'en-tête	Description
Accept	Type de contenu accepté par le browser (par exemple <i>text/html</i> ). Voir <a href="#">types MIME</a>
Accept-Charset	Jeu de caractères attendu par le browser
Accept-Encoding	Codage de données accepté par le browser
Accept-Language	Langage attendu par le browser (anglais par défaut)
Authorization	Identification du browser auprès du serveur
Content-Encoding	Type de codage du corps de la requête
Content-Language	Type de langage du corps de la requête
Content-Length	Longueur du corps de la requête
Content-Type	Type de contenu du corps de la requête (par exemple <i>text/html</i> ). Voir <a href="#">types MIME</a>
Date	Date de début de transfert des données
Forwarded	Utilisé par les machines intermédiaires entre le browser et le serveur
From	Permet de spécifier l'adresse e-mail du client
From	Permet de spécifier que le document doit être envoyé s'il a été modifié depuis une certaine date
Link	Relation entre deux URL
Orig-URL	URL d'origine de la requête
Referer	URL du lien à partir duquel la requête a été effectuée
User-Agent	Chaîne donnant des informations sur le client, comme le nom et la version du navigateur, du système d'exploitation

## En-têtes de réponse

Nom de l'en-tête< souligné>	Description< souligné>
Content-Encoding	Type de codage du corps de la réponse
Content-Language	Type de langage du corps de la réponse
Content-Length	Longueur du corps de la réponse
Content-Type	Type de contenu du corps de la réponse (par exemple <i>text/html</i> ). Voir <a href="#">types MIME</a>
Date	Date de début de transfert des données
Expires	Date limite de consommation des données
Forwarded	Utilisé par les machines intermédiaires entre le browser et le serveur
Location	Redirection vers une nouvelle URL associée au document
Server	Caractéristiques du serveur ayant envoyé la réponse

### Les codes de réponse

Ce sont les codes que vous voyez lorsque le navigateur n'arrive pas à vous fournir la page demandée. Le code de réponse est constitué de trois chiffres : le premier indique la classe de statut et les suivants la nature exacte de l'erreur.

Code	Message	Description
10x	Message d'information	Ces codes ne sont pas utilisés dans la version 1.0 du protocole
20x	Réussite	Ces codes indiquent le bon déroulement de la transaction
200	OK	La requête a été accomplie correctement
201	CREATED	Elle suit une commande <a href="#">POST</a> , elle indique la réussite, le corps du reste du document est sensé indiquer l' <a href="#">URL</a> à laquelle le document nouvellement créé devrait se trouver.
202	ACCEPTED	La requête a été acceptée, mais la procédure qui suit n'a pas été accomplie
203	PARTIAL INFORMATION	Lorsque ce code est reçu en réponse à une commande <a href="#">GET</a> , cela indique que la réponse n'est pas complète.
204	NO RESPONSE	Le serveur a reçu la requête mais il n'y a pas d'information à renvoyer
205	RESET CONTENT	Le serveur indique au navigateur de supprimer le contenu des champs d'un formulaire

206	PARTIAL CONTENT	Il s'agit d'une réponse à une requête comportant l'en-tête <i>range</i> . Le serveur doit indiquer l'en-tête <i>content-Range</i>
30x	Redirection	Ces codes indiquent que la ressource n'est plus à l'emplacement indiqué
301	MOVED	Les données demandées ont été transférées à une nouvelle adresse
302	FOUND	Les données demandées sont à une nouvelle URL, mais ont cependant peut-être été déplacées depuis...
303	METHOD	Cela implique que le client doit essayer une nouvelle adresse, en essayant de préférence une autre méthode que <a href="#">GET</a>
304	NOT MODIFIED	Si le client a effectué une commande <a href="#">GET</a> conditionnelle (en demandant si le document a été modifié depuis la dernière fois) et que le document n'a pas été modifié il renvoie ce code.
40x	Erreur due au client	Ces codes indiquent que la requête est incorrecte
400	BAD REQUEST	La syntaxe de la requête est mal formulée ou est impossible à satisfaire
401	UNAUTHORIZED	Le paramètre du message donne les spécifications des formes d'autorisation acceptables. Le client doit reformuler sa requête avec les bonnes données d'autorisation
402	PAYMENT REQUIRED	Le client doit reformuler sa demande avec les bonnes données de paiement
403	FORBIDDEN	L'accès à la ressource est tout simplement interdit
404	NOT FOUND	Classique! Le serveur n'a rien trouvé à l'adresse spécifiée. Parti sans laisser d'adresse... :)

400	BAD REQUEST	La syntaxe de la requête est mal formulée ou est impossible à satisfaire
401	UNAUTHORIZED	Le paramètre du message donne les spécifications des formes d'autorisation acceptables. Le client doit reformuler sa requête avec les bonnes données d'autorisation
402	PAYMENT REQUIRED	Le client doit reformuler sa demande avec les bonnes données de paiement
403	FORBIDDEN	L'accès à la ressource est tout simplement interdit
404	NOT FOUND	Classique! Le serveur n'a rien trouvé à l'adresse spécifiée. Parti sans laisser d'adresse....)
50x	<b>Erreur due au serveur</b>	<b>Ces codes indiquent qu'il y a eu une erreur interne du serveur</b>
500	INTERNAL ERROR	Le serveur a rencontré une condition inattendue qui l'a empêché de donner suite à la demande (comme quoi il leur en arrive des trucs aux serveurs...)
501	NOT IMPLEMENTED	Le serveur ne supporte pas le service demandé (on ne peut pas tout savoir faire...)
502	BAD GATEWAY	Le serveur a reçu une réponse invalide de la part du serveur auquel il essayait d'accéder en agissant comme une passerelle ou un proxy
503	SERVICE UNAVAILABLE	Le serveur ne peut pas vous répondre à l'instant présent, car le trafic est trop dense (toutes les lignes de votre correspondant sont occupées veuillez rappeler ultérieurement)
504	GATEWAY TIMEOUT	La réponse du serveur a été trop longue vis-à-vis du temps pendant lequel la passerelle était préparée à l'attendre (le temps qui vous était imparti est maintenant écoulé...)

## Client en C et Java



## 2-4 Interpréteurs Web

Rendu graphique

Scripts

Plugins

Protocoles (Présentation + Application)

## 2-4 Interpréteurs Web

Rendu graphique

Gecko	Mozilla Firefox, Kompozer, Nvu	FZ
KHTML, WebKit	Konqueror, GtkHTML	KDE
WebCore	Safari, Google Chrome, Chromium	Apple
Trident	IE, Netscape	Microsoft
Elektra -> Presto	Opera	Opera Software

## 2-4 Interpréteurs Web

### Scripts

**Recherche**

**Scriptol**

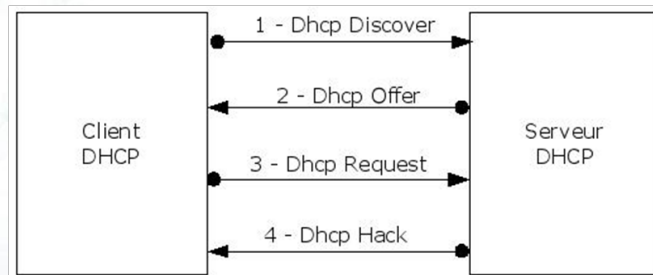
### 3.DHCP

- **DHCP**(Dynamic Host Configuration Protocol): est un **protocole** qui permet à un serveur DHCP d'affecter des adresses IP temporaires (et d'autres paramètres) à des terminaux ou des PCs lors de leur mise sous tension.
- Ce protocole simplifie grandement l'administration d'un réseau.

### 3.DHCP

- Lorsque le client DHCP démarre, Il envoie une trame "DHCP-DISCOVER", destinée à trouver un serveur DHCP. Cette trame est un "broadcast". N'ayant pas encore d'adresse IP, il adopte provisoirement l'adresse 0.0.0.0. IL fournit aussi sa "MAC Address".
- Le serveur DHCP du réseau qui va recevoir cette trame va se sentir concerné et répondre par un "DHCP-OFFER". Cette trame, elle aussi en "broadcast" car il n'est pas encore possible d'atteindre le client (il n'a pas encore d'adresse IP valide), la "MAC Address" du client, avec également l'adresse IP du serveur.
- Le client répond alors par un DHCP-REQUEST au serveur (donc toujours en "Broadcast") pour indiquer qu'il accepte.
- Le serveur DHCP Concerné répond définitivement par un DHCP-ACK qui constitue une confirmation. L'adresse du client est alors marquée comme utilisée et ne sera plus proposée à un autre client pour toute une durée.

### 3.DHCP



### 4. DNS

- DNS (Domain Name Service) est un serveur qui associe des noms aux adresses IP des terminaux ou des PCs.
- L'utilisation d'un serveur DNS simplifie la gestion du réseau car les utilisateurs ont simplement besoin de connaître le nom des machines sans se préoccuper des adresses IP.

#### 4. DNS

- Sur Internet une machine est identifiée de manière unique par son adresse IP → Annuaire Adresse IP / Nom
- Au début (1970-1984) : annuaire complet dans un fichier texte
- Géré par Network Information Center (<http://www.nic.com>)

#### 4.DNS

Exemples de correspondances:

Ccp.poste.dz. IN A 80.249.65.13

www.umc.edu.dz. IN A 80.246.8.22

localhost.stage.org. IN A 127.0.0.1

## Structuration des noms DNS

Hiérarchique par domaine:

Exemple: www.umc.edu.dz

- machine www dans le domaine umc, lui-même dans le domaine edu lui-même dans le domaine dz
- Analogie nom de fichier/répertoire à l'envers avec le . à la place de /
- Les majuscules ne sont pas significatives

## ***SERVEURS DNS***

- Racine : environ 15 serveurs de nom répartis dans le monde.
- Serveur origine (ou primaire, ou maitre) géré par ICANN → ***A.ROOT-SERVERS.NET***
- Serveurs miroirs (ou secondaire, ou esclave) → de ***B.ROOT-SERVERS.NET*** à ***M.ROOT-SERVERS.NET***
- Modification manuel faite sur le serveur primaire
- Echange des bases de données automatique vers les serveurs secondaires.

## Serveurs DNS

- Gère la base de données contenant
- nom/@IP des machines du domaine
- nom/@IP des serveurs d'un sous-domaine
- Système robuste par redondance: plusieurs serveurs possèdent la base de données d'un domaine

### 1. Pourquoi Apache est-il devenu un standard ?

- Coût nul
- Code source disponible et modifiable permet un développement rapide du serveur.
- Très grande flexibilité du serveur grâce à sa structure modulaire l'ajout d'un nouveau module permet d'ajouter de nouvelles fonctionnalités.

## **2. Le protocole HTTP**

HTTP ou HyperText Transfer Protocol est un protocole de requêtes et de réponses.

Le dialogue entre un client web (un navigateur tel que Netscape) et un serveur (Apache) se traduit par une requête du client à laquelle le serveur répond en effectuant le traitement intermédiaire adéquat.

## **3. INSTALLATION ET EXÉCUTION D'APACHE**

### **3.1. Installer Apache à partir d'un paquetage préconstruit**

Se référer au site <http://www.funix.org> pour les détails de l'installation.

Le .tar.gz du serveur Apache peut être téléchargé gratuitement.

**Décompressez l'archive, compilez la puis l'installer .**

```
[root@host] /# cd /usr/src
[root@host] /# tar -xvzf Apache-*.tar.gz
[root@host] /# cd Apache -*
[root@host] /# ./configure
[root@host] /# make
[root@host] /# make install
```

**Sur RedHat ou Fedora, vous pouvez récupérer le rpm et l'installer.**

```
[root@host] /# rpm -ivh Apache -*.rpm
```



### **3.2. Tester le serveur Apache**

L'approche la plus simple et la plus naturelle pour tester le bon fonctionnement d'Apache consiste à lancer un navigateur web et d'utiliser `http://localhost` comme adresse.

Si aucune page n'apparaît vérifier tout d'abord si le serveur est bien lancé en affichant les processus :

```
$ ps - aux | grep httpd
```

### **3.3. Lancer, arrêter et redémarrer le serveur**

Pour lancer Apache, il suffit d'exécuter `httpd` :

```
/etc/init.d/httpd start
```

Une fois lancé, Apache s'exécute en tâche de fond. Pour arrêter Apache :

```
/etc/init.d/httpd stop
```

Si le process apache ne réponds plus aux commandes il faut rechercher les processus `httpd` :

```
$ ps - aux | grep httpd
```

puis les détruire :

```
<blockquote>
```

```
$ kill -9 <pid>
```

Dans le cas où on a simplement changé un fichier de configuration on peut

exécuter la commande :

```
/etc/init.d/httpd graceful
```

qui demande au process apache courant de relire ses fichiers de configuration.

### **3.5.Lancer automatiquement le serveur au démarrage de l'ordinateur**

Il est important de s'assurer que les sites hébergés soient accessibles à tout moment ou presque. Dans le cas d'une coupure de courant prolongée dépassant la durée d'efficacité de l'onduleur, le serveur s'éteint. Lorsque le courant est rétabli, la machine redémarre. Il est crucial comme tout autre service réseau que le serveur Apache soit redémarré automatiquement afin que le temps d'indisponibilité des sites hébergés soit le plus court possible.

Si apache a été installé par un paquetage de la distribution le démarrage au boot doit être automatique.

Si apache a été compilé et installé depuis le code source il faut recopier le script `apachectl` dans le répertoire `/etc/init.d/` et créer un lien dans le répertoire `/etc/rcX.d` correspondant au run-level désiré.

#### **4.CONFIGURATION DE BASE DU SERVEUR HTTP**

La configuration du serveur Apache peut s'effectuer soit en modifiant manuellement son fichier de configuration avec un éditeur de texte soit en utilisant une interface graphique dédiée (par exemple l'excellent WebMin). Même si la configuration d'Apache à partir d'un fichier texte peut paraître fastidieux, cette méthode présente l'avantage de contrôler exactement et de façon exhaustive le fonctionnement du serveur.

Le fichier de configuration d'Apache se nomme `httpd.conf` et est placé dans le répertoire `conf/` du répertoire d'Apache dans le cas d'une installation manuelle (par exemple `/usr/local/apache/conf/` ). Dans le cas d'une installation à partir d'un paquetage, le fichier `httpd.conf` se trouve généralement dans le répertoire `/etc/httpd/conf` A noter que dans les versions anciennes d'Apache, le contenu du fichier `httpd.conf` était réparti dans 3 fichiers différents : `httpd.conf`, `access.conf` et `srm.conf`.

A l'issu de l'installation du serveur Apache, le fichier `httpd.conf` est configuré par défaut.

Avant d'envisager de le modifier, assurer au préalable que le serveur fonctionne et n'oublier pas de dupliquer le fichier de configuration par défaut.

En effectuant par exemple :

```
$ cp httpd.conf httpd.conf.default
```

Comment Apache structure-t-il sa configuration ?

- les directives de conteneur Apache ont une portée limitée ;
- les directives peuvent être utilisées à portée globale ou locale ;
- il est possible d'outrepasser une directive à l'aide d'une configuration par répertoire.

**Nom de serveur** : ServerName www.ouaga.bf Il ne s'agit pas du nom du serveur pour lequel Apache répond mais du nom avec lequel Apache envoie sa réponse.

**Adresse IP du serveur** : BindAddress 192.168.13.11

**Port(s) à écouter** : Port 80 Il est possible de remplacer les configurations Port et Bindaddress par la directive Listen : Listen 192.168.13.11 :80 (L'avantage est qu'il est possible de spécifier plusieurs fois la directive Listen, ce qui n'est pas le cas pour BindAddress)

**Serveur autonome ou non** : ServerType standalone : apache est autonome  
ServerType inetd : apache n'est lancé que lorsque inetd reçoit une requête sur les ports pour lequel il est configuré.

**Utilisateur et groupe** : dans le but de rendre Apache moins vulnérable aux éventuelles attaques, il est possible de le configurer pour qu'il s'exécute sous un utilisateur et un groupe ayant des droits restreints.

User nobody

Group nobody

**Adresse électronique de l'administrateur** : ServerAdmin [webmaster@ouaga.bf](mailto:webmaster@ouaga.bf)

**Racine du serveur** : ServerRoot /etc/httpd

**Journal d'erreur par défaut** : ErrorLog logs/error\_log

**Emplacement par défaut des pages html** : DocumentRoot /home/httpd/html

**Pages par défaut** : DirectoryIndex index.htm index.html index.php

## **5. ANALYSER LES LOGS APACHE**

Le premier Webalizer est sûrement le plus connu, le deuxième Awstats est le nouveau venu dans le domaine, il gagne à être connu car il fournit une information plus riche que Webalizer. Les deux outils possèdent l'avantage de présenter les résultats dans une page web.

## **Conclusion**

Nous avons vu dans cet exposé qu'est ce qu'un serveur Apache et comment le configurer dans les détails en éditant les fichiers de configuration via les lignes de commandes.