Architecture et langages Web
(NFA040)

Services Internet – HTTP[1]

Tristan Crolard

Laboratoire CEDRIC
Equipe « Systèmes Sûrs »

tristan.crolard@cnam.fr

cedric.cnam.fr/sys/crolard

1. Cours basé sur les supports de Sami Taktak

Protocoles Applicatifs

- Grand nombre de services disponibles
- Services les plus utilisés:
  - Consultation de page web
  - Utilisation de messagerie électronique
  - Transfert de fichiers
- Service fournis par un serveur à client
- Existe aussi des applications peer-to-peer
- Applications situées tout en haut du modèle OSI (couche applicative)

Protocoles Applicatifs

- Protocoles de niveau applicatif nécessaires pour qu'une application transmette ses données et que le service correspondant soit délivré
- Définissent des règles de dialogue entre le client et le serveur
  - Séquence de messages échangés entre client et serveur
  - Type de messages échangés (demande, réponse, confirmation, ...)
  - Syntaxe adoptée par les différents types de messages
  - Sémantique des différents champs (sens des informations qu'ils contiennent)
- Exemple : HTTP (Hyper Text Transfer Protocol)

Principaux Services Internet

- Domain Name System (DNS) ou système de noms de domaines
- Dynamic Host Configuration Protocol (DHCP)
- Service Web
- Messagerie électronique ou email
- Service de transfert de fichier
- Network Time Protocol (NTP) ou protocole d'heure réseau
- Voix et vidéo sur IP

Numéro de ports normalisés (IANA)[2]

https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

Port	Protocol	Use
20, 21	FTP	File transfer
22	SSH	Remote login, replacement for Telnet
25	SMTP	Email
80	HTTP	World Wide Web
110	POP-3	Remote email access
143	IMAP	Remote email access
443	HTTPS	Secure Web (HTTP over SSL/TLS)
543	RTSP	Media player control
631	IPP	Printer sharing

- **Ports 0–1023** : *system or well-known ports*
- **Ports 1024–49151** : *user or registered ports*
- **Ports >49151** : *dynamic / private ports*

---

2. Extrait tiré de [Tanebaum 11]

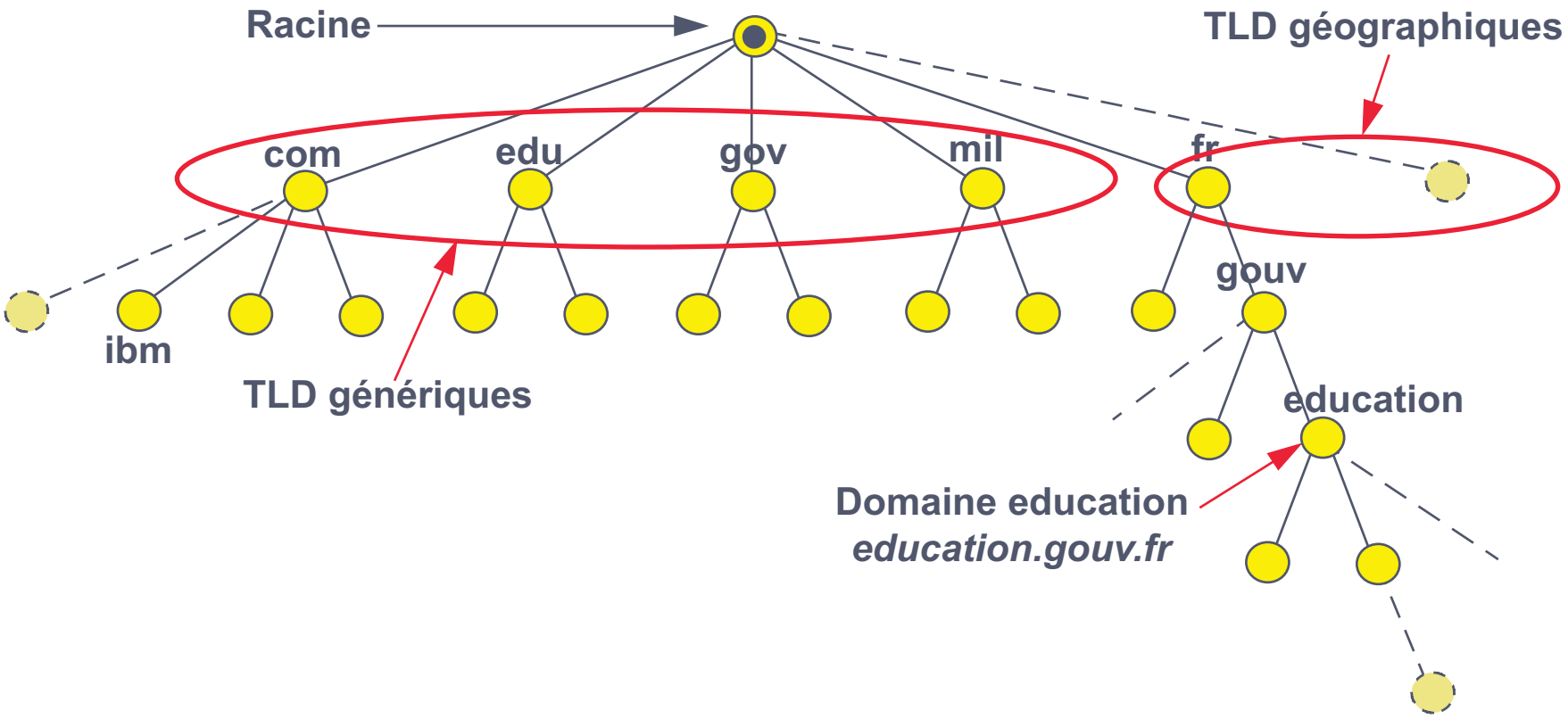
DNS – Domain Name System

- Machines localisées par leur adresse
- Peu pratique et difficile à mémoriser
- Utilisation d'un annuaire faisant correspondre un nom à une adresse IP
- Service Fournis par un serveur DNS
- Organisation arborescente : racine, TLD (Top Level Domain), ...
- TDL gérés par l'ICANN

Organisation Arborescente du Nommage DNS

- Autorité de nommage d'Internet : ICANN (Internet Corporation for Assigned Names and Numbers)
- Un responsable par zone de nomage
- En France, l'AFNIC (Association Française pour le Nommage Internet en Coopération)
  - responsable de la zone `.fr`
- Domaines de premier niveau : TLD, Top Level Domain
- 2 types de TLD :
  - gTLD (generic TLD) : domaines générique, `.com`, `.net`, `.org`, ...
  - ccTLD (country code TLD) : domaines correspondant à un pays, `.fr`, `.es`, `.us`, ...

Organisation Arborescente du Nommage DNS[3]

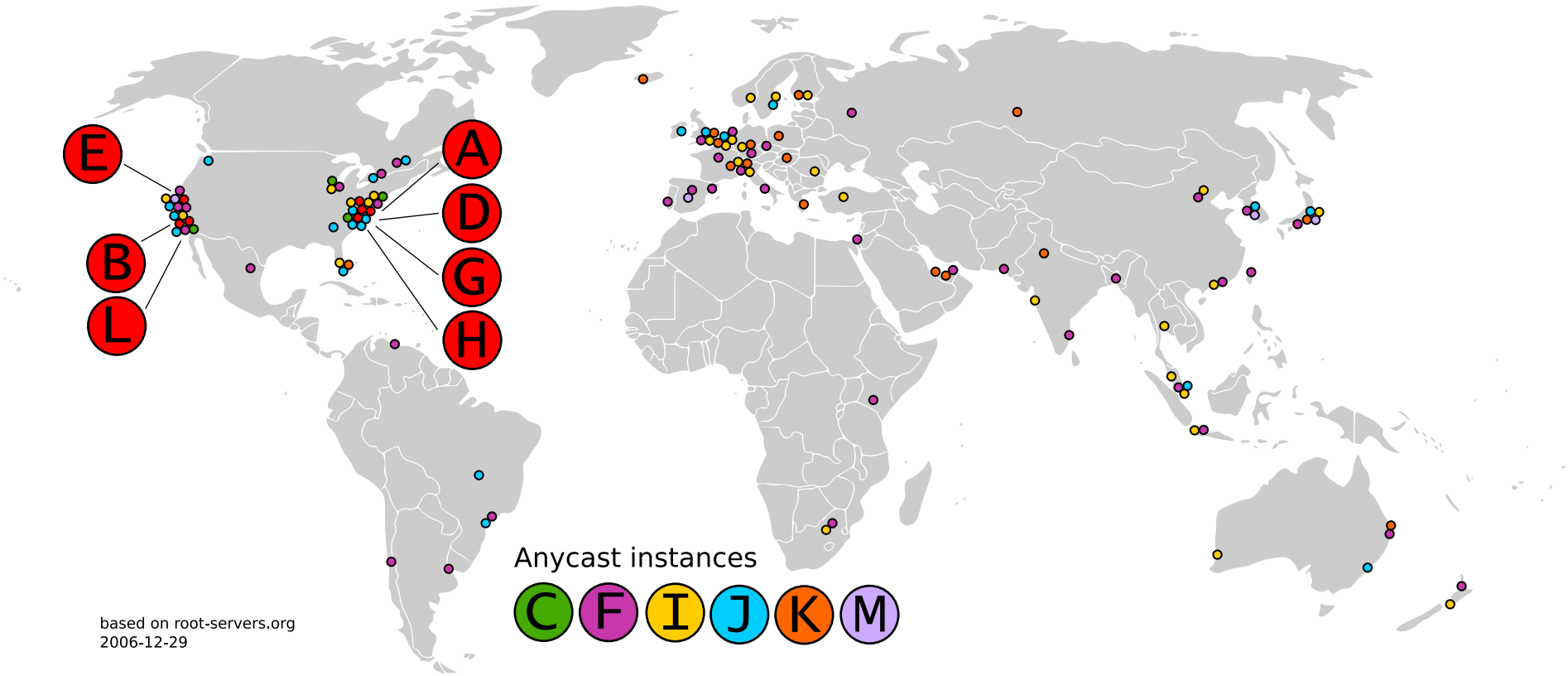


3. Les figures sont extraites de [Lohier 10] (sauf indication contraire)

Serveurs Racines

- Racine gérée par l'ICANN
- 13 serveurs de nom racine
- Répartis à travers le monde (mais forte proportion aux USA)
- Nommé `a.root-servers.net` à `m.root-servers.net`
- FQDN (Fully Qualified Domaine Name) : nom complet d'un hôte, composé du nom de l'hôte suivit du domaine auquel il appartient
- Exemple : `www.exemple.com`

Carte des Serveurs Racines



Résolution de Nom de Domaine

- Chaque client possède au moins une adresse d'un serveur DNS
- Serveur DNS interrogé par le client lorsqu'un nom doit être résolue
- Résolution récursive des noms par les serveurs DNS
- Un serveur DNS peut résoudre un nom s'il a autorité sur un domaine ou si il possède déjà en cache l'information
- Les serveurs DNS utilisent les serveurs racine pour trouver les serveurs DNS ayant autorité sur un domaine

Enregistrement DNS

- Contenus dans une base de données
- Structurée en « enregistrement de ressources » ou « Ressource Records »(RR)
- À chaque nom de domaine correspond un enregistrement RR

Enregistrement DNS

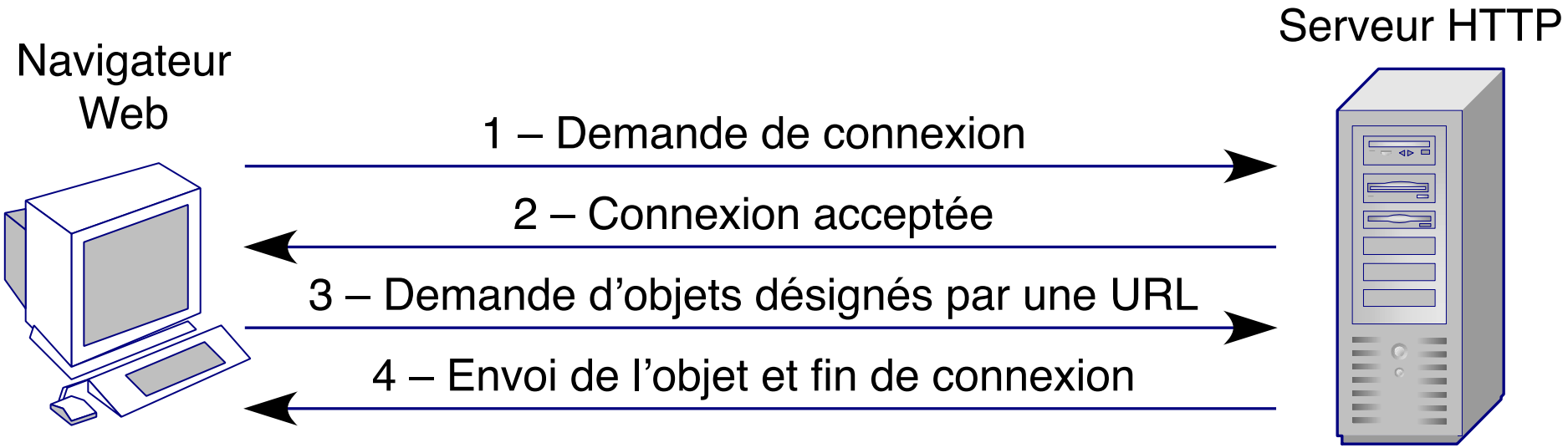
Plusieurs types de RR :

- **A** : Address, correspondance entre nom et IP d'une machine
- **NS** : Name Server, serveur de nom pour ce domaine
- **CNAME** : Canonical NAME, nom d'origine (des alias peuvent exister)
- **SOA** : Start Of Authority, serveur(s) faisant autorité sur la zone
- **PTR** : PoinTeR, correspondance entre l'adresse et le nom (résolution inverse)
- **MX** : Mail eXchange, indique le serveur de messagerie pour ce domaine

Service Web

- Fournit par un serveur Web, Apache, IIS, ...
- HyperText Transfer Protocol (HTTP) ou protocole de transfert hypertexte
- Protocole de la couche application
- Pages Web écrites en HTML (Hypertext Markup Language)
- Client : navigateur web, mais aussi aspirateurs de site Web ou robots d'indexation
- Inventé en 1990 par Tim Berners-Lee (alors informaticien au CERN)

Hyperliens vers des ressources web



Hyperliens vers des Ressources web

- Service web : permet d'accéder à des documents au format HTML
- HTML : Hyper Text Markup Language
- Utilise le protocole HTTP (Hyper Text Transfert Protocol)
- Application coté client : navigateur web (firefox, chrome, safari, ...)
- Application coté serveur : serveur web (Apache, IIS, ...)
- Liens hypertextes mis en évidence par le navigateur : correspondent à la localisation d'une autre ressource
- Initialement prévus que pour le texte

URL – Uniform Ressource Locator

- URL : Uniform Ressource Locator
- Nom donnée à la localisation de la ressource web correspondant à l'hyperlien
- Comporte au minimum le nom de la machine (serveur HTTP, FTP, ...)
- Éventuellement un chemin d'accès à la ressource sur la machine et le nom de la ressource

URL – Uniform Ressource Locator

Syntaxe générale :

protocol://[user:password@]host[:port][/path][/document]

- `user`, `password` : nom de l'utilisateur et son mot de passe
- `host` : nom complet de l'hôte FQDN
- `port` : identifie la connexion sur le serveur suivant le protocole (80 pour HTTP)
- `path` : chemin d'accès à la ressource (/répertoire/sous-répertoire)
- `document` : nom du fichier de la ressource (index.html, video.mpeg, ...)

Protocol HTTP

- méthode : commande spécifiant un type de requête
- `GET` : méthode la plus courante pour demander une ressource
- `HEAD` : demande que des informations sur la ressource, sans demander la ressource elle-même
- `POST` : méthode utilisée pour soumettre des données typiquement depuis un formulaire HTML
- `OPTION` : méthode permettant d'obtenir les options de communication d'une ressource ou du serveur en général
- `CONNECT` : méthode permettant d'utiliser un proxy comme un tunnel de communication

Protocol HTTP

- `TRACE` : méthode demande au serveur de retourner ce qu'il a reçu, dans le but de tester et effectuer un diagnostic sur la connexion
- `PUT` : méthode permet de remplacer ou d'ajouter une ressource sur le serveur
- `DELETE` : méthode permettant de supprimer une ressource du serveur
- Ces 2 dernières méthodes nécessitent généralement un accès privilégié
- Certains serveurs autorisent d'autres méthodes de gestion de leurs ressources (par exemple WebDAV)

Requêtes HTTP

```
Méthode PATH Version
En-Tête : valeur
En-Tête : valeur
...
Corps de la requête
```

- Première ligne : méthode (GET, HEAD, POST, ...), l'URL, version du protocole utilisé
- En-tête : champ facultatif donnant des informations sur la requête (navigateur utilisé, OS, ...)
- HTTP 1.1 : hôte placé dans le champ d'en-tête « Host » obligatoire.

Réponse HTTP

```
Version Code de réponse
  En-Tête : valeur
  En-Tête : valeur
  ...
  Corps de la réponse
```

- Première ligne : version puis du code réponse
  - Code sur 3 chiffres commençant par :
  - 2 : bon déroulement des opérations; `200 OK`, `204 NO REPONSE`
  - 3 : indique une redirection; `310 MOVED`
  - 4 : erreur due au client; `404 NOT FOUND`
  - 5 : erreur due au serveur; `503 SERVICE UNAVAILABLE`

Méthode GET

```
GET / HTTP/1.1
Host: www.cnam.fr

User-Agent: Mozilla/5.0 (X11; U; Linux x86_64; en-US;
  rv:1.9.2.13) Gecko/20101206 Ubuntu/10.10 (maverick)
  Firefox/3.6.13
Accept: text/html,application/xhtml+xml
Accept-Language: fr,fr-fr; en-us
Accept-Encoding: gzip,deflate
Accept-CharSet: ISO-8859-1, utf8
Keep-Alive: 300
Connection: keep-alive
```

Réponse GET

```
HTTP/1.1 200 OK
Date: Thu, 21 Feb 2013 14:21:23 GMT
Server: Apache/2.2.3 (Red Hat)
Set-Cookie: JSESSIONID=858621A395AE8A; Path=/
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html;charset=ISO-8859-1

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
 lang="fr" xml:lang="fr">
```

Méthode HEAD

```
HEAD / HTTP/1.1
Host: www.cnam.fr

HTTP/1.1 200 OK
Date: Thu, 21 Feb 2013 14:47:38 GMT
Server: Apache/2.2.3 (Red Hat)
Set-Cookie: JSESSIONID=8021F058339AABB084CED; Path=/
Connection: close
Content-Type: text/html;charset=ISO-8859-1
```

Méthode POST

- Permet au client d'envoyer des données au serveur
- Utilisé pour l'envoi de formulaire
- Variables (nom et valeur) dans le corps de la requête
- Format : `name1=value1&name2=value2&...`
- Utilisé par un script coté serveur
- Résultat renvoyé comme pour la méthode `GET`

Méthode POST

```
POST / HTTP/1.1
Host: jungle.net
User-Agent: Mozilla/5.0 (X11; U; Linux x86_64; ... )
Accept: text/html,application/xhtml+xml
Accept-Language: fr,fr-fr; en-us
Accept-Encoding: gzip,deflate
Accept-CharSet: ISO-8859-1, utf8
Keep-Alive: 300
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 65

username=tarzan%40jungle.net&password=jane
         &logon=Log+on
```

Cookies

Transactions à état

- HTTP $\Rightarrow$ dialogue client-serveur sans état
- Pas d'information stockée coté serveur concernant les transactions
- Système de cookies inventé par Netscape permet d'établir des transaction avec état
- Normalisé dans HTTP 1.1
- Cookie : information envoyée par le serveur et stockée coté client sous forme d'un fichier texte
- Renvoyée par le client lors d'une nouvelle transaction

Cookies

- Entête HTTP spécifique dans la réponse du serveur
  `Set-Cookie: Nom=Valeur; expires=Date; path=Chemin; domain=Domaine; secure`
- `Nom` : permet d'identifier le cookie
- `expires` : indique la date d'expiration du cookie
- `domain` : spécifie le nom de domaine d'application du cookie
- `path` : utilisé pour désigner un sous-ensemble de ressources auxquelles le cookie est accessible
- `secure` : utilisé pour que le cookie ne soit envoyé que si la transmission est sécurisée