

Web

Bruno BEAUFILS

2021/2022

1. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

Résumé

Exercices

2. World Wide Web

Introduction

HTTP

Site web

3. DNS

Principes

Fonctionnement

Définitions

- ▶ Un ensemble de *pages web* (ressources) accessibles via une **adresse web**
 - ▶ hébergé par un **serveur** web
- ▶ Un serveur web est
 - ▶ un ordinateur connecté à l'Internet. . .
 - ▶ . . . accessible via le protocole HTTP
- ▶ Un site web nécessite
 - ▶ un **serveur** web
 - ▶ un ensemble de **pages** web

*une adresse sur le réseau
des fichiers sur un ordinateur*

Définitions

- ▶ Un ensemble de *pages web* (ressources) accessibles via une **adresse web**
 - ▶ hébergé par un **serveur** web
- ▶ Un serveur web est
 - ▶ un ordinateur connecté à l'Internet. . .
 - ▶ . . . accessible via le protocole HTTP
- ▶ Un site web nécessite
 - ▶ un **serveur** web
 - ▶ un ensemble de **pages** web

*une adresse sur le réseau
des fichiers sur un ordinateur*

Définitions

- ▶ Un ensemble de *pages web* (ressources) accessibles via une **adresse web**
 - ▶ hébergé par un **serveur** web
- ▶ Un serveur web est
 - ▶ un ordinateur connecté à l'Internet. . .
 - ▶ . . . accessible via le protocole HTTP
- ▶ Un site web nécessite
 - ▶ un **serveur** web
 - ▶ un ensemble de **pages** web

*une adresse sur le réseau
des fichiers sur un ordinateur*

- ▶ un ordinateur connecté à l'Internet
 - ▶ avec une adresse IP
 - ▶ un nom dans le DNS
- ▶ offrant un service de communication en HTTP
 - ▶ architecture client/serveur
 - ▶ logiciel spécifique
- ▶ peut répondre différemment en fonction du nom avec lequel on le questionne

▶ Utilisation **statique**

- ▶ un appel identique fait plusieurs fois donne toujours la même réponse
- ▶ projection d'un espace de nommage sur un système de fichier
 - ▶ renvoyer des fichiers
 - ▶ ressources de la requête = chemin relatif du fichier à renvoyer
- ▶ utilisation historique (échange de documents hypertextes)

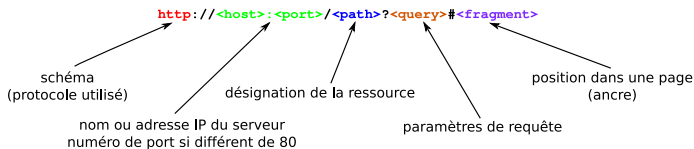
▶ Utilisation **dynamique**

- ▶ exécute une fonction/commande pour construire la réponse
- ▶ interface CGI ([RFC 3875](#)) : transmission requête/paramètres au programme
- ▶ langage adapté (perl, php, python, etc.)
- ▶ mode d'exécution différent

▶ Intermédiaire

- ▶ reçoit la requête et la transmet à un autre serveur
- ▶ **Proxy** : pour pouvoir accéder à l'Internet depuis un réseau privé
 - ▶ Utilisation : filtrage, cache
- ▶ **Reverse Proxy** : pour pouvoir accéder à un réseau privé depuis l'Internet
 - ▶ Utilisation : répartition de charge, protection, cache

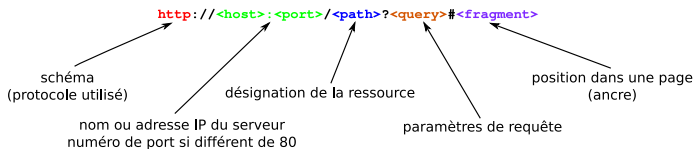
- ▶ permet d'identifier des ressources **web**
 - ▶ à n'importe quel endroit sur le réseau
 - ▶ de n'importe quel type (page web, images, fichiers, etc.)
- ▶ **identifier** les ressources :
 - ▶ **URI** (*Uniform Resource Identifier*) RFC-3986
 - ▶ une chaîne identifiant une ressource sur un réseau physique ou abstraite



- ▶ **localiser** les ressources :
 - ▶ **URL** (*Uniform Resource Locator*)
 - ▶ une chaîne spécifiant où aller chercher une ressource
 - ▶ identifier une ressource par son emplacement *adresse web = URL*

<https://m2-ape-cee.fr/index.html>

- ▶ permet d'identifier des ressources **web**
 - ▶ à n'importe quel endroit sur le réseau
 - ▶ de n'importe quel type (page web, images, fichiers, etc.)
- ▶ **identifier** les ressources :
 - ▶ **URI** (*Uniform Resource Identifier*) RFC-3986
 - ▶ une chaîne identifiant une ressource sur un réseau physique ou abstraite



- ▶ **localiser** les ressources :
 - ▶ **URL** (*Uniform Resource Locator*)
 - ▶ une chaîne spécifiant où aller chercher une ressource
 - ▶ identifier une ressource par son emplacement *adresse web = URL*

<https://m2-ape-cee.fr/index.html>

- ▶ identifier les types de ressources : **Type MIME**
 - ▶ 10 types définis
 - ▶ application,
 - ▶ audio,
 - ▶ example,
 - ▶ font,
 - ▶ image,
 - ▶ message,
 - ▶ model,
 - ▶ multipart,
 - ▶ text
 - ▶ video
 - ▶ sous-types précisables après une barre de fraction
- ▶ standard pour l'email et le web
- ▶ exemple : `text/html; charset=UTF-8`

- ▶ Une page web = un contenu et une mise en forme
 - ▶ Contenu
 - ▶ un fichier HTML
 - ▶ des médias (images, sons, etc.)
 - ▶ Mise en forme
 - ▶ des styles (un ou plusieurs fichiers CSS)
 - ▶ des programmes de modification dynamique (JavaScript)
- ▶ Liaison des fichiers via des attributs d'éléments HTML
 - ▶ href dans <a> ou <link/>
 - ▶ src dans ou <script>
- ▶ Attributs spécifient comment accéder à la ressource (au fichier)
 - ▶ via une **URL**
 - ▶ ou via un **chemin** *fichiers sur le même serveur*
 - ▶ liste des dossiers à traverser séparés par une barre de fraction (*slash*) : /
 - ▶ .. remonte vers le dossier parent
- ▶ Organisation d'un site web en **dossiers**
 - ▶ permet d'éviter la redondance
 - ▶ permet d'identifier aisément les données publiées

- ▶ Une page web = un contenu et une mise en forme
 - ▶ Contenu
 - ▶ un fichier HTML
 - ▶ des médias (images, sons, etc.)
 - ▶ Mise en forme
 - ▶ des styles (un ou plusieurs fichiers CSS)
 - ▶ des programmes de modification dynamique (JavaScript)
- ▶ Liaison des fichiers via des attributs d'éléments HTML
 - ▶ href dans `<a>` ou `<link/>`
 - ▶ src dans `` ou `<script>`
- ▶ Attributs spécifient comment accéder à la ressource (au fichier)
 - ▶ via une URL
 - ▶ ou via un chemin *fichiers sur le même serveur*
 - ▶ liste des dossiers à traverser séparés par une barre de fraction (*slash*) : /
 - ▶ .. remonte vers le dossier parent
- ▶ Organisation d'un site web en dossiers
 - ▶ permet d'éviter la redondance
 - ▶ permet d'identifier aisément les données publiées

- ▶ Une page web = un contenu et une mise en forme
 - ▶ Contenu
 - ▶ un fichier HTML
 - ▶ des médias (images, sons, etc.)
 - ▶ Mise en forme
 - ▶ des styles (un ou plusieurs fichiers CSS)
 - ▶ des programmes de modification dynamique (JavaScript)
- ▶ Liaison des fichiers via des attributs d'éléments HTML
 - ▶ href dans <a> ou <link/>
 - ▶ src dans ou <script>
- ▶ Attributs spécifient comment accéder à la ressource (au fichier)
 - ▶ via une **URL**
 - ▶ ou via un **chemin** *fichiers sur le même serveur*
 - ▶ liste des dossiers à traverser séparés par une barre de fraction (*slash*) : /
 - ▶ .. remonte vers le dossier parent
- ▶ Organisation d'un site web en **dossiers**
 - ▶ permet d'éviter la redondance
 - ▶ permet d'identifier aisément les données publiées

- ▶ Une page web = un contenu et une mise en forme
 - ▶ Contenu
 - ▶ un fichier HTML
 - ▶ des médias (images, sons, etc.)
 - ▶ Mise en forme
 - ▶ des styles (un ou plusieurs fichiers CSS)
 - ▶ des programmes de modification dynamique (JavaScript)
- ▶ Liaison des fichiers via des attributs d'éléments HTML
 - ▶ href dans `<a>` ou `<link/>`
 - ▶ src dans `` ou `<script>`
- ▶ Attributs spécifient comment accéder à la ressource (au fichier)
 - ▶ via une **URL**
 - ▶ ou via un **chemin** *fichiers sur le même serveur*
 - ▶ liste des dossiers à traverser séparés par une barre de fraction (*slash*) : /
 - ▶ .. remonte vers le dossier parent
- ▶ Organisation d'un site web en **dossiers**
 - ▶ permet d'éviter la redondance
 - ▶ permet d'identifier aisément les données publiées

- ▶ Copier les pages web sur le serveur web
 - ▶ copie **distante** via un outil adéquat
 - ▶ Linux : Navigateur de fichiers (via ssh)
 - ▶ Windows : [FileZilla](#), [WinSCP](#)
 - ▶ MacOS : [CyberDuck](#)
- ▶ Vérification de la syntaxe des fichiers publiés
 - ▶ HTML/CSS
 - ▶ <https://validator.w3.org/unicorn/>
 - ▶ <http://w3c.github.io/developers/tools/#tools>

▶ Rôle de base

1. interpréter/préparer une URL pour **faire une requête** à un serveur
2. interpréter et **présenter la réponse** à l'utilisateur
 - ▶ généralement afficher une page HTML
 - ▶ faire un rendu (graphique/sonore/braille)
 - ▶ séparation fond/forme (HTML/CSS)

▶ Rôles courants

- ▶ préparer la requête
 - ▶ saisie de formulaire, raccourci (moteur de recherche), authentification HTTP
- ▶ aider l'utilisateur
 - ▶ historique, bookmarks, paramétrage de la connexion (proxy), stockage local d'informations (mot de passe, etc.)

▶ Travail caché

- ▶ Lors de son rendu d'une page HTML le client peut faire beaucoup de requêtes sans en informer **explicitement** l'utilisateur
- ▶ Récupération nécessaire au rendu ou demandé par la présentation de la réponse
 - ▶ image
 - ▶ script
 - ▶ styles
- ▶ Chaque requête est en HTTP et peut-être accompagné d'un envoi de cookie
 - ▶ Cookies tierce-partie

▶ Rôle de base

1. interpréter/préparer une URL pour **faire une requête** à un serveur
2. interpréter et **présenter la réponse** à l'utilisateur
 - ▶ généralement afficher une page HTML
 - ▶ faire un rendu (graphique/sonore/braille)
 - ▶ séparation fond/forme (HTML/CSS)

▶ Rôles courants

- ▶ préparer la requête
 - ▶ saisie de formulaire, raccourci (moteur de recherche), authentification HTTP
- ▶ aider l'utilisateur
 - ▶ historique, bookmarks, paramétrage de la connexion (proxy), stockage local d'informations (mot de passe, etc.)

▶ Travail caché

- ▶ Lors de son rendu d'une page HTML le client peut faire beaucoup de requêtes sans en informer **explicitement** l'utilisateur
- ▶ Récupération nécessaire au rendu ou demandé par la présentation de la réponse
 - ▶ image
 - ▶ script
 - ▶ styles
- ▶ Chaque requête est en HTTP et peut-être accompagné d'un envoi de cookie
 - ▶ Cookies tierce-partie

▶ Rôle de base

1. interpréter/préparer une URL pour **faire une requête** à un serveur
2. interpréter et **présenter la réponse** à l'utilisateur
 - ▶ généralement afficher une page HTML
 - ▶ faire un rendu (graphique/sonore/braille)
 - ▶ séparation fond/forme (HTML/CSS)

▶ Rôles courants

- ▶ préparer la requête
 - ▶ saisie de formulaire, raccourci (moteur de recherche), authentification HTTP
- ▶ aider l'utilisateur
 - ▶ historique, bookmarks, paramétrage de la connexion (proxy), stockage local d'informations (mot de passe, etc.)

▶ Travail caché

- ▶ Lors de son rendu d'une page HTML le client peut faire beaucoup de requêtes sans en informer **explicitement** l'utilisateur
- ▶ Récupération nécessaire au rendu ou demandé par la présentation de la réponse
 - ▶ image
 - ▶ script
 - ▶ styles
- ▶ Chaque requête est en HTTP et peut-être accompagné d'un envoi de cookie
 - ▶ Cookies tierce-partie

- ▶ Utilisation de robots logiciels
 - ▶ *crawler*
 - ▶ lire le plus de page web possibles
 - ▶ stocker des copies
- ▶ Utilisation du **graphe** des pages
 - ▶ créée via les ancrs et références
- ▶ **Indexation des pages**
 - ▶ par mots
 - ▶ par nombre de liens

1. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

Résumé

Exercices

2. World Wide Web

Introduction

HTTP

Site web

3. DNS

Principes

Fonctionnement

1. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

Résumé

Exercices

2. World Wide Web

Introduction

HTTP

Site web

3. DNS

Principes

Fonctionnement

- ▶ pour les humains moyens :

retenir des mots est plus simple que de retenir des numéros

- ▶ plein de conversion noms \longleftrightarrow numéro

- ▶ **noms de machines** \rightarrow **adresse IP** *(www.fdn.fr au lieu de 80.67.169.52)*
- ▶ noms de services \rightarrow port *(www plutôt que 80)*

- ▶ historiquement

- ▶ juste des fichiers transmis à toutes les machines de l'Internet
 - ▶ /etc/services
 - ▶ /etc/hosts
- ▶ gérés par des individus pour le compte d'Université
- ▶ passage à l'échelle difficile

Une infrastructure importante de l'Internet

- ▶ quasiment **tous** les services réseaux utilisent le DNS
 - ▶ web
 - ▶ email
- ▶ service **très** sensible de l'accès à l'Internet
 - ▶ disponibilité
 - ▶ fiabilité (censure)
- ▶ géré par des *associations*
 - ▶ **IANA** (Internet Assigned Numbers Authority)
 - ▶ qui dépend de l'**ICANN** (Internet Corporation for Assigned Names and Numbers)
 - ▶ associations de droits américains
 - ▶ débat sur le statut de ces associations et le poids des gouvernements
 - ▶ important pour les règles de délégations de gestion
 - ▶ autres associations importantes de l'Internet
 - ▶ **ISOC** (Internet Society)
 - ▶ **IAB** (Internet Architecture Board)
 - ▶ **IETF** (Internet Engineering Task Force)
- ▶ délégation de gestion
 - ▶ registres
 - ▶ bureaux d'enregistrements
 - ▶ Gandi.net
 - ▶ OVH

DNS = Domain Name System

1. une **base de données** distribuée

- ▶ base de données hiérarchique
- ▶ délégation de gestion (administrative et technique)
- ▶ peut stocker *presque* n'importe quoi
 - ▶ conversion noms vers adresse
- ▶ utilisé pour beaucoup de choses
 - ▶ fédération de services (email, etc.)
 - ▶ identification de serveurs (certification, etc.)

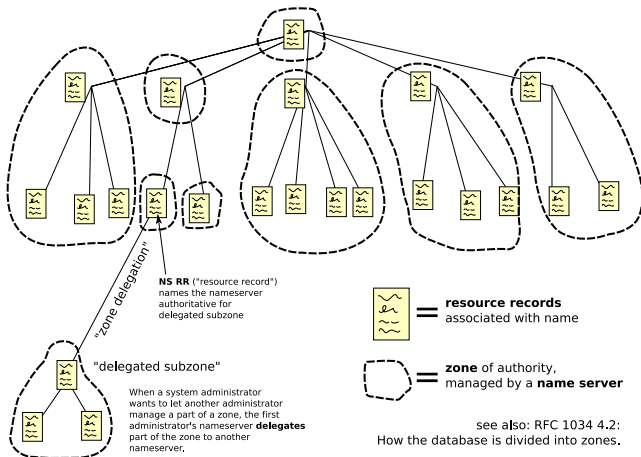
2. un **protocole de communication**

- ▶ interrogation de la base
- ▶ assurer la cohérence et la fiabilité de la base

3. des **serveurs**

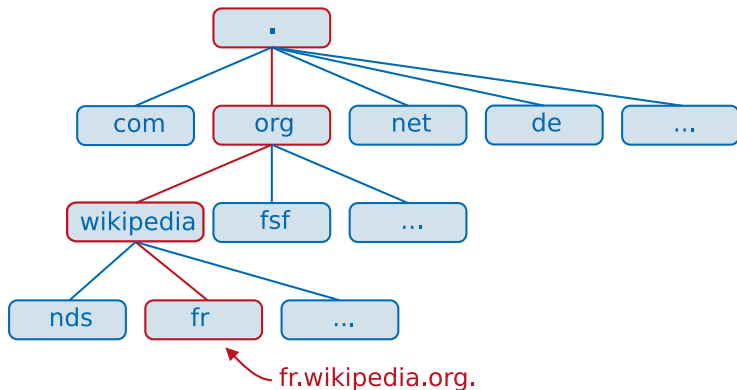
- ▶ répondre à une demande de résolution de noms
- ▶ 2 modes :
 - ▶ **autorité** sur des parties (sous-ensemble) de la base
 - ▶ **récuratif** sur d'autres
- ▶ UDP sur le port 53

Domain Name Space



- ▶ zones, domaines, sous-domaines
- ▶ registres, bureau d'enregistrement
- ▶ TLD : Top-Level Domain
 - ▶ gTLD : Generic TLD (.com, .net, .org, etc.)
 - ▶ ccTLD : Country Code TLD (.fr, .uk, .de, .tv, etc.)
 - ▶ IDN : Internationalized TLD

Exemple



Crédit : Wikimedia

1. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

Résumé

Exercices

2. World Wide Web

Introduction

HTTP

Site web

3. DNS

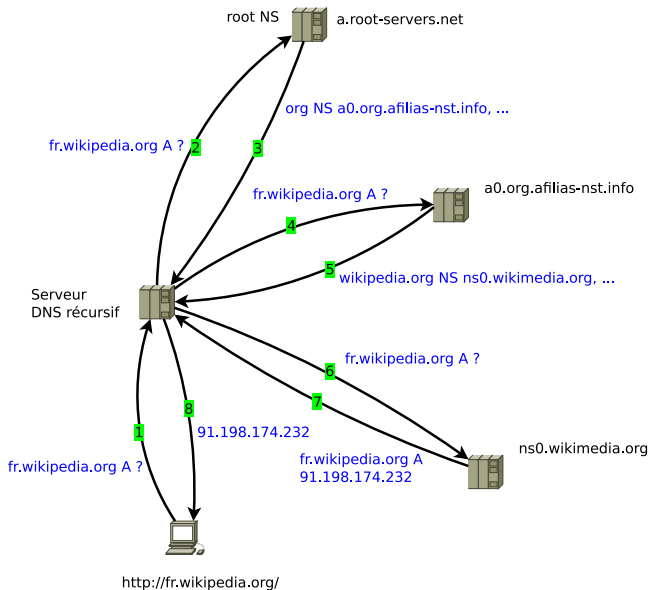
Principes

Fonctionnement

Types d'enregistrements dans le DNS

- ▶ **A** utilisation d'une adresse IPv4
- ▶ **AAAA** utilisation d'une adresse IPv6
- ▶ **CNAME** définition d'un alias
- ▶ **MX** définition d'un serveur d'email pour le domaine
 - ▶ reçoit les connexions SMTP pour les emails du domaine
 - ▶ plusieurs serveurs avec des priorités
- ▶ **PTR** correspondance adresse IP vers nom
 - ▶ zone dédiée à la résolution **inverse** (-in-addr.arpa et ip6.arpa)
 - ▶ ex : 1.168.192.in-addr.arpa pour les machines de 192.168.1.0/24
 - ▶ ex : 0.a.1.3.8.c.b.0.1.0.0.2.ip6.arpa. pour les machines 2001 :0bc8 :31a0 : :/48
- ▶ **NS** définition du serveur de nom
 - ▶ au moins un enregistrement par domaine
 - ▶ peut en contenir plusieurs (primaire, secondaire, etc.)
- ▶ **TXT** information quelconque

Résolution



- ▶ Types de serveurs
 - ▶ **autoritaire**
 - ▶ **récuratif**
- ▶ Fournisseur de services DNS récuratifs
 - ▶ FAI
 - ▶ limités aux abonnés (4 grands)
 - ▶ ouverts à tous (FAI de FFDN dont **FDN** par exemple)
 - ▶ Grande entreprise/organisations/projets (ouverts)
 - ▶ Cloudfare / Google / IBM
 - ▶ OpenDNS
 - ▶ vous même
- ▶ Faire attention à ce que font les serveurs des informations collectés
 - ▶ même principe le serveur sait (et stocke) beaucoup de choses
 - ▶ modèle économique : profilage marketing