

Web

Bruno BEAUFILS

2023/2024

1. Introduction

2. Documents numériques

3. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

4. DNS

Principes

Fonctionnement

5. World Wide Web

Introduction

HTTP

Site web

1. Introduction

2. Documents numériques

3. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

4. DNS

Principes

Fonctionnement

5. World Wide Web

Introduction

HTTP

Site web

Création documentaire

- ▶ 2 aspects importants
 - ▶ **fond** : informations, données
 - ▶ **forme** : présentation
- ▶ Exemples
 - ▶ nombre : quantité vs notation/représentation (indo-arabes, romains, etc.)
 - ▶ roman : auteur vs typographe
- ▶ Informatique a permis de **séparer fond et forme** facilement
 - ▶ Avantages
 - ▶ **accessibilité** (handicap, automatisation, etc.)
 - ▶ **universalité** = pas lié à un seul mode de représentation
 - ▶ seules les **données** sont importantes dans les documents
 - ▶ Problèmes
 - ▶ Outils mal conçus (Word) et éducation/formation déficiente
 - ▶ Utilisateurs plus souvent habitués au **WYSIWYG** qu'au **WYSIWYM**

Web conçu pour être un système **universel** d'échange de documents

- ▶ 2 outils pour décrire les documents
 - ▶ HTML : description des informations
 - ▶ CSS : description de la mise en forme
- ▶ 1 langage de programmation pour manipuler le tout
 - ▶ Javascript

fond
forme

Création documentaire

- ▶ 2 aspects importants
 - ▶ **fond** : informations, données
 - ▶ **forme** : présentation
- ▶ Exemples
 - ▶ nombre : quantité vs notation/représentation (indo-arabes, romains, etc.)
 - ▶ roman : auteur vs typographe
- ▶ Informatique a permis de **séparer fond et forme** facilement
 - ▶ Avantages
 - ▶ **accessibilité** (handicap, automatisation, etc.)
 - ▶ **universalité** = pas lié à un seul mode de représentation
 - ▶ seules les **données** sont importantes dans les documents
 - ▶ Problèmes
 - ▶ Outils mal conçus (Word) et éducation/formation déficiente
 - ▶ Utilisateurs plus souvent habitués au **WYSIWYG** qu'au **WYSIWYM**

Web conçu pour être un système **universel** d'échange de documents

- ▶ 2 outils pour décrire les documents
 - ▶ HTML : description des informations
 - ▶ CSS : description de la mise en forme
- ▶ 1 langage de programmation pour manipuler le tout
 - ▶ Javascript

fond
forme

- ▶ 2 aspects importants
 - ▶ **fond** : informations, données
 - ▶ **forme** : présentation
- ▶ Exemples
 - ▶ nombre : quantité vs notation/représentation (indo-arabes, romains, etc.)
 - ▶ roman : auteur vs typographe
- ▶ Informatique a permis de **séparer fond et forme** facilement
 - ▶ Avantages
 - ▶ **accessibilité** (handicap, automatisation, etc.)
 - ▶ **universalité** = pas lié à un seul mode de représentation
 - ▶ seules les **données** sont importantes dans les documents
 - ▶ Problèmes
 - ▶ Outils mal conçus (Word) et éducation/formation déficiente
 - ▶ Utilisateurs plus souvent habitués au **WYSIWYG** qu'au **WYSIWYM**

Web conçu pour être un système **universel** d'échange de documents

- ▶ 2 outils pour décrire les documents
 - ▶ HTML : description des informations
 - ▶ CSS : description de la mise en forme
- ▶ 1 langage de programmation pour manipuler le tout
 - ▶ Javascript

fond
forme

1. Introduction

2. Documents numériques

3. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

4. DNS

Principes

Fonctionnement

5. World Wide Web

Introduction

HTTP

Site web

1. Introduction

2. Documents numériques

3. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

4. DNS

Principes

Fonctionnement

5. World Wide Web

Introduction

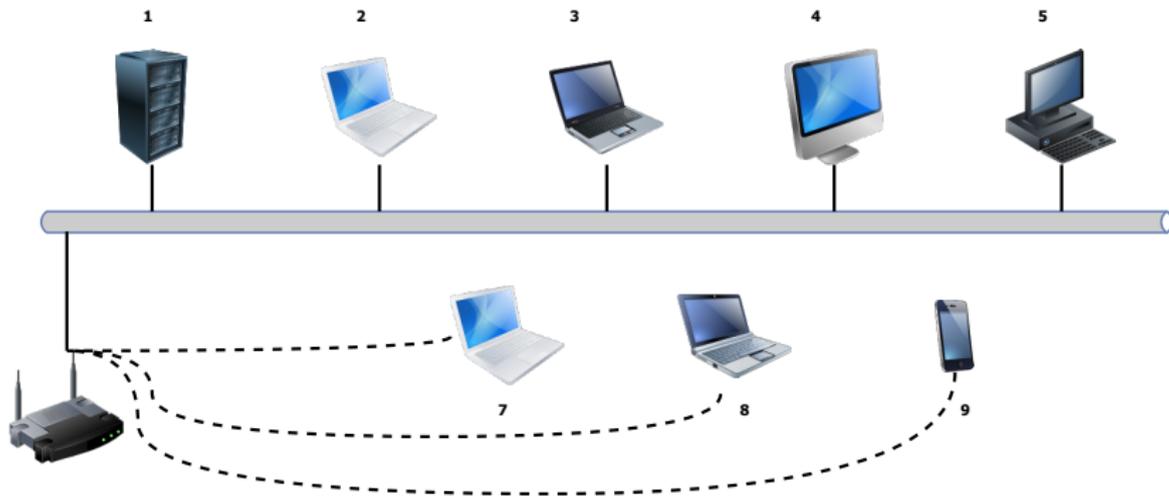
HTTP

Site web

Un réseau de machines

Toutes les machines

- ▶ échangent des nombres par un mécanisme physique (électricité, ondes radios)
- ▶ sont connectées les unes aux autres sur un *segment* physique
- ▶ ont accès à toutes les informations échangées sur le segment
- ▶ s'identifient par un numéro unique



Un réseau de machines (représentation schématique)

Un réseau de réseaux

- ▶ principe de l'interconnexion des réseaux

1. Introduction

2. Documents numériques

3. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

4. DNS

Principes

Fonctionnement

5. World Wide Web

Introduction

HTTP

Site web

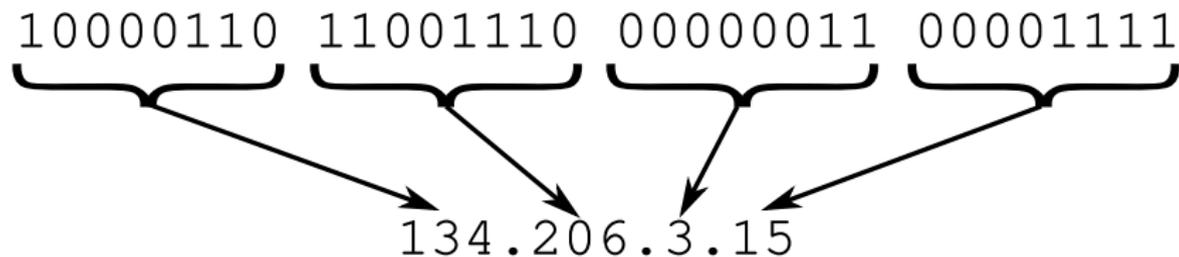
- ▶ une adresse IP est composé de 2 parties
 1. réseau
 2. adresse

numéro de réseau

numéro de machine

- ▶ elle combine 2 fonctions
 - ▶ **localisation** grâce au numéro de réseau
 - ▶ **identification**

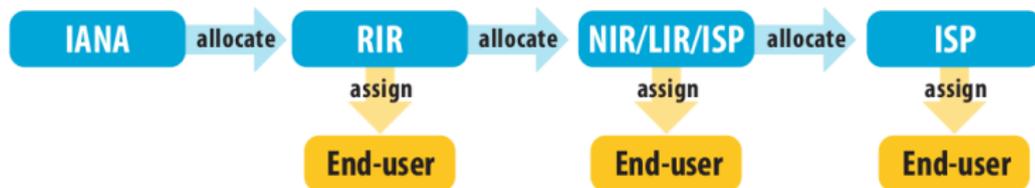
Adresse (IPv4)



- ▶ une adresse IPv4 est un identifiant sur 32 bits
- ▶ on regroupe ces bits en 4 octets pour faciliter la lecture en décimal

Gestion de l'adressage

- ▶ l'espace d'adressage est géré par l'**IANA**
- ▶ elle délègue des blocs aux **Registres Internet**



- ▶ Les registres correspondent aux grandes régions géographiques



1. Introduction

2. Documents numériques

3. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

4. DNS

Principes

Fonctionnement

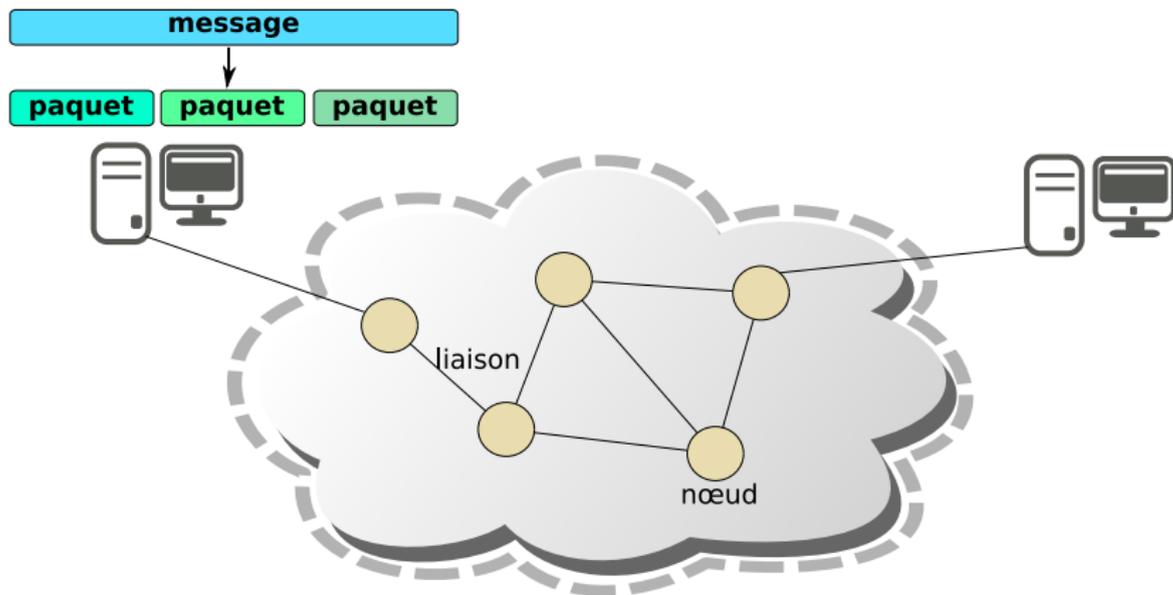
5. World Wide Web

Introduction

HTTP

Site web

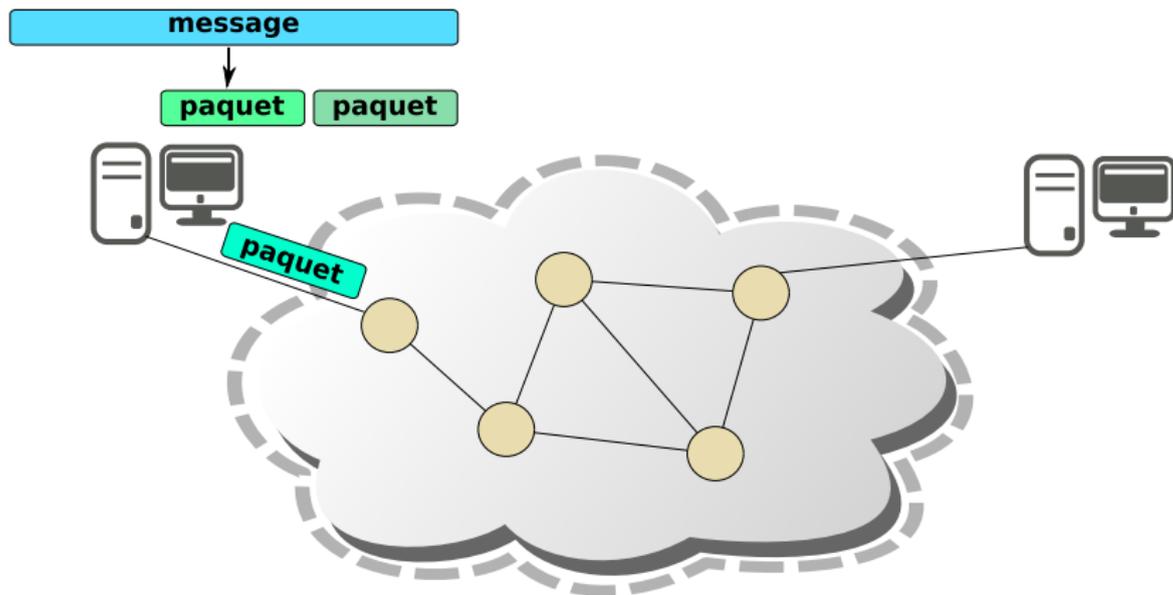
Découpage de l'information en paquets



Comment atteindre la destination ?

- ▶ par commutation des paquets
- ▶ et routage

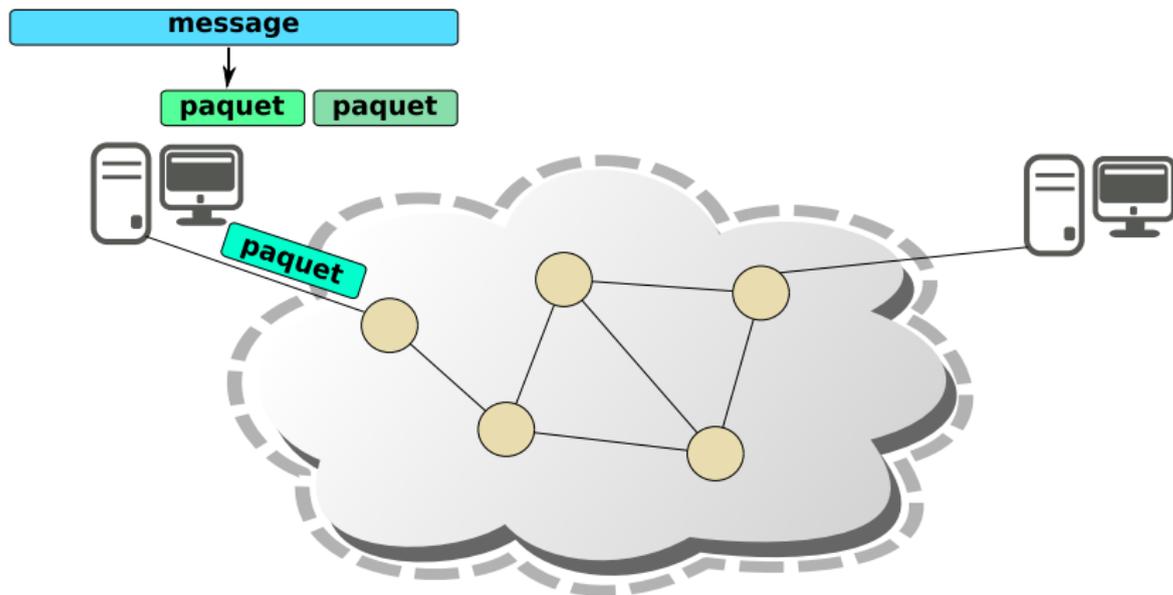
Découpage de l'information en paquets



Comment atteindre la destination ?

- ▶ par commutation des paquets
- ▶ et routage

Découpage de l'information en paquets

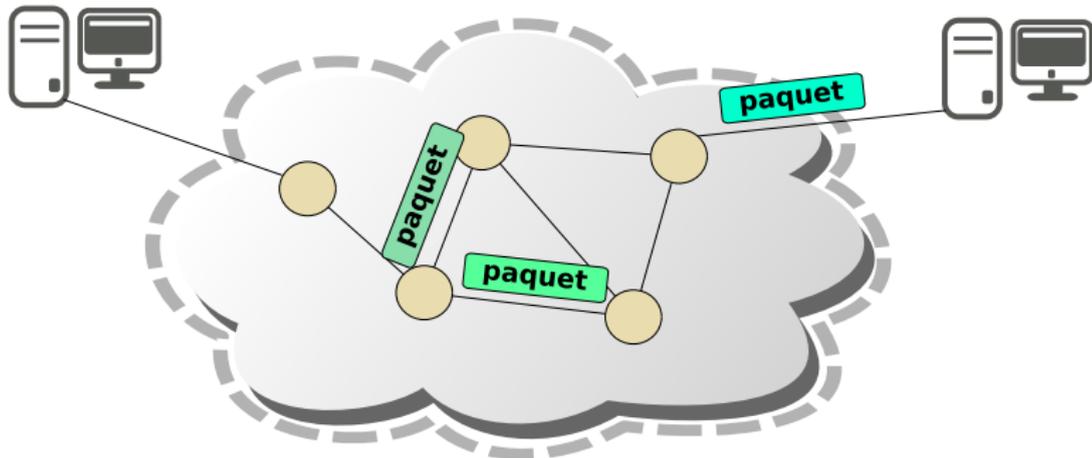


Comment atteindre la destination ?

- ▶ par commutation des paquets
- ▶ et routage

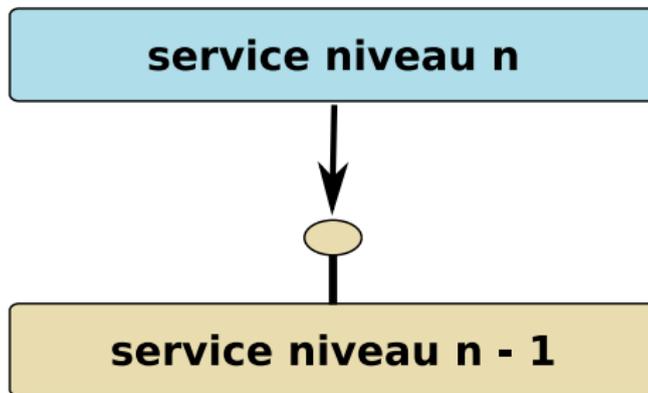
Routage de paquets

message



Routage des paquets

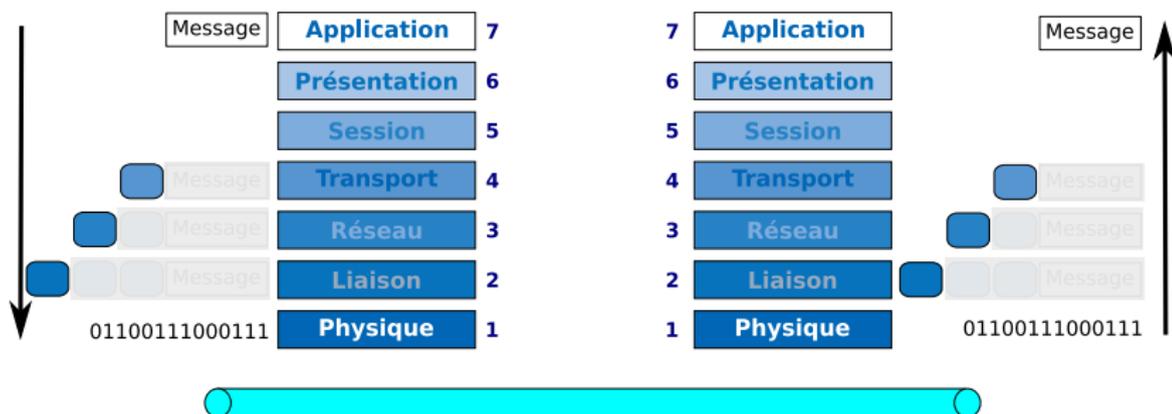
- ▶ chaque noeud choisi la prochaine liaison en fonction de la destination
- ▶ les paquets peuvent suivre des chemins différents



- ▶ permet d'identifier un ensemble de fonctionnalités ou **service**
- ▶ l'indépendance entre les couches est assurée par l'**interface**
- ▶ dans une couche, un **protocole** permet la communication entre entités

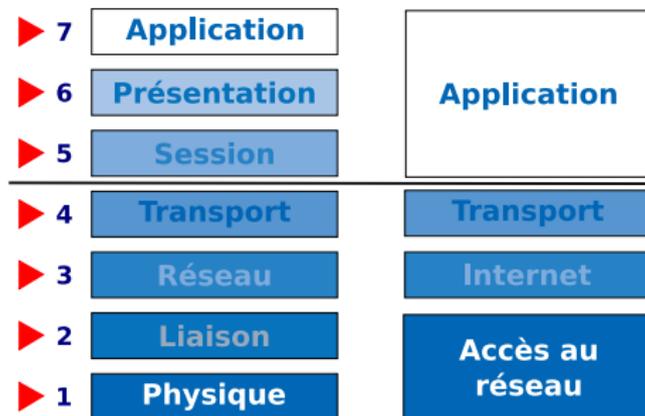
Modèle OSI : encapsulation

- ▶ analogie : lettre dans une enveloppe



- ▶ chaque couche ajoute une en-tête propre à son protocole
- ▶ les informations transmises par une couche sont opaques pour la couche inférieure

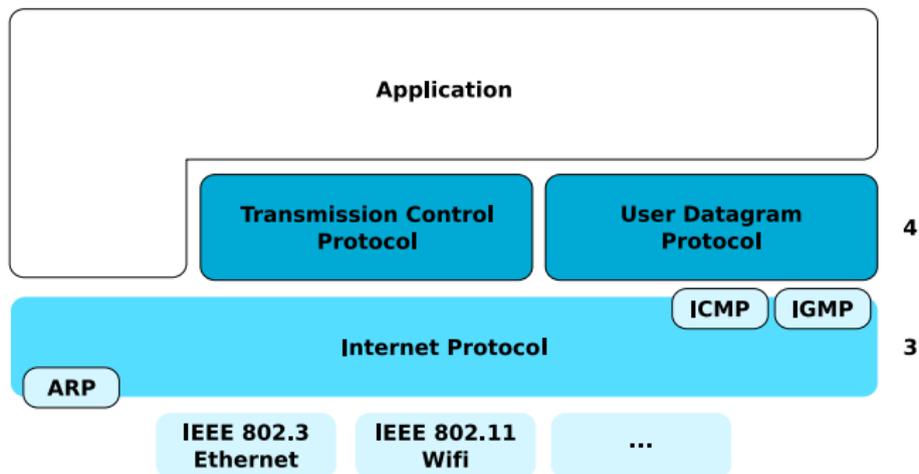
Modèle Internet (TCP/IP) : protocoles



- ▶ la couche 3 est l'**Internet Protocol**
- ▶ la couche 4 offre 2 protocoles :
 - ▶ **TCP** (*Transmission Control Protocol*)
 - ▶ **UDP** (*User Datagram Protocol*)

connecté, fiable
déconnecté, non fiable

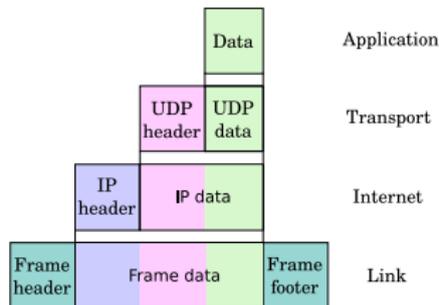
Modèle Internet : couches



- ▶ couches 1, 2 normalisées par [Institute of Electrical and Electronics Engineers \(IEEE\)](#)
- ▶ autres couches gérées par [The Internet Engineering Task Force \(IETF\)](#)

Modèle Internet : encapsulation

- ▶ même principe que pour OSI
- ▶ exemple avec UDP



1. Introduction

2. Documents numériques

3. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

4. DNS

Principes

Fonctionnement

5. World Wide Web

Introduction

HTTP

Site web

Protocoles de communication

- ▶ Chiffrement du transport : SSL/TLS
- ▶ Transformation nom en adresse : DNS
- ▶ Courrier électronique : SMTP et POP/IMAP
- ▶ Échange de document lié : HTTP et HTTPS

Analogie avec les restaurants

- ▶ clients et serveur dans un restaurant
 1. serveur attend
 2. client demande
 3. échange entre le client et le serveur
 - ▶ souvent le serveur fait ce que le client lui a demandé
 4. connexion terminée
- ▶ plusieurs serveurs dans un restaurant
 - ▶ maître d'hôtel
 - ▶ chef de rang
 - ▶ serveurs
 - ▶ sommelier
 - ▶ responsable de zones différents d'une salle
- ▶ plusieurs logiciels sur une même machine
 - ▶ rendent des **services** différents
 - ▶ **écoutent** sur des *ports* différents

1. Introduction

2. Documents numériques

3. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

4. DNS

Principes

Fonctionnement

5. World Wide Web

Introduction

HTTP

Site web

1. Introduction

2. Documents numériques

3. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

4. DNS

Principes

Fonctionnement

5. World Wide Web

Introduction

HTTP

Site web

- ▶ 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.)

1. un **protocole de communication**

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

1. 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

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.)

1. un **protocole de communication**

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

1. 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

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.)

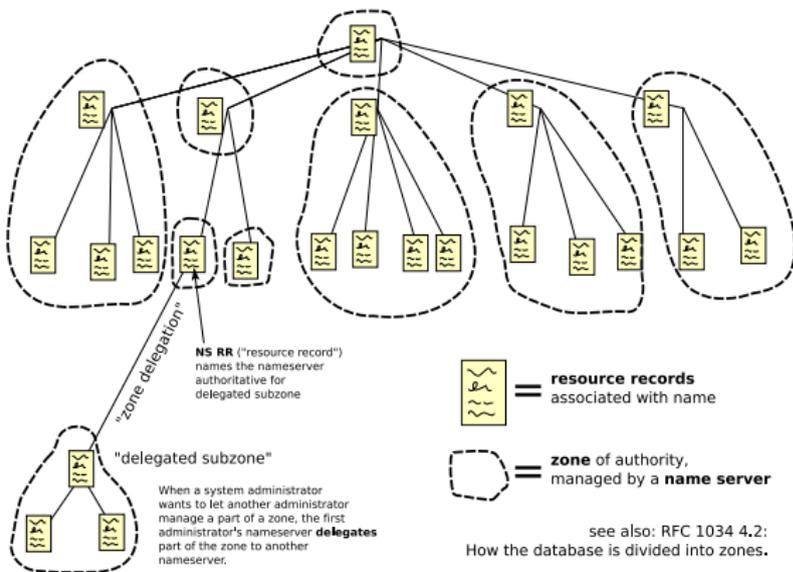
1. un **protocole de communication**

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

1. 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

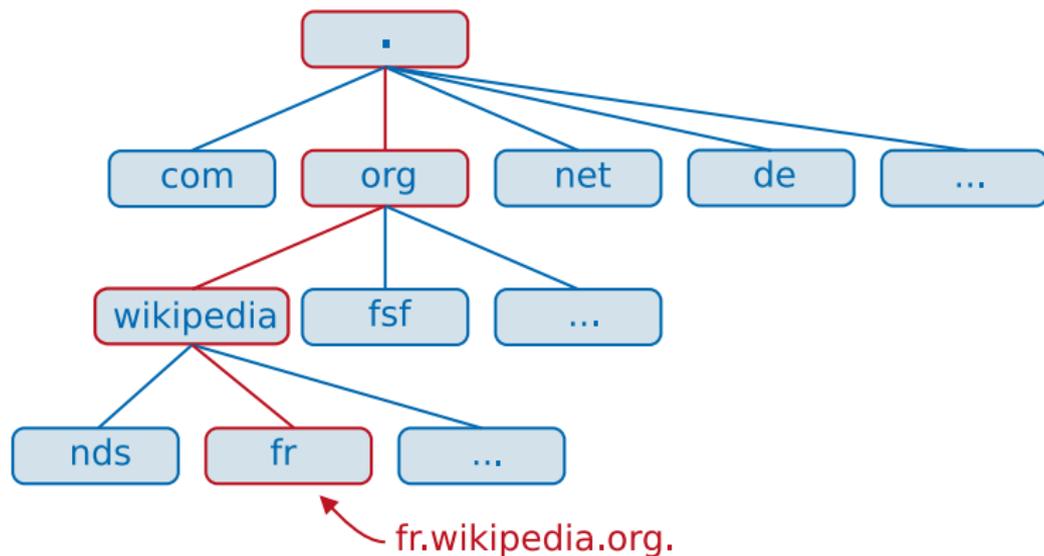
Domain Name Space



Crédit : Wikimedia

- ▶ 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. Introduction

2. Documents numériques

3. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

4. DNS

Principes

Fonctionnement

5. World Wide Web

Introduction

HTTP

Site web

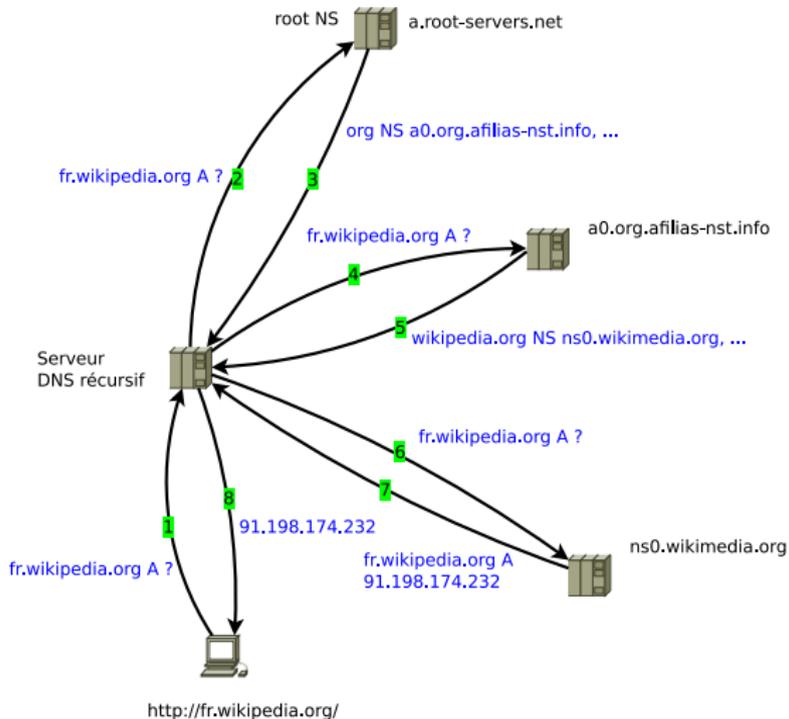
Types d'enregistrements dans le DNS

- ▶ **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

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



Crédit : Wikimedia

- ▶ 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

1. Introduction

2. Documents numériques

3. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

4. DNS

Principes

Fonctionnement

5. World Wide Web

Introduction

HTTP

Site web

1. Introduction

2. Documents numériques

3. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

4. DNS

Principes

Fonctionnement

5. World Wide Web

Introduction

HTTP

Site web

Qu'est-ce que le web ?

- ▶ Un système permettant à des **logiciels** d'échanger des **documents**
 - ▶ service de la couche application
 - ▶ architecture client/serveur
 - ▶ le client demande un document (**une page web**) *requête*
 - ▶ le serveur la lui envoie *réponse*
 - ▶ construit pour être **universel** = pas lié à un seul mode de représentation
 - ▶ **accessibilité** (handicap, automatisation, etc.)
 - ▶ seul les **données** sont importantes dans les documents
- ▶ Logiciels
 - ▶ Client web Navigateur
 - ▶ Mozilla **Firefox**, Google Chrome / **Chromium**, Apple Safari, Microsoft Edge, etc.
 - ▶ Aide l'utilisateur à faire les requêtes
 - ▶ Présente les réponses, documents reçus (mise en forme)
 - ▶ Serveur web
 - ▶ **Apache**, **nginx**, Microsoft IIS, etc.
 - ▶ Stocke ou prépare des documents
- ▶ Standards et formats
 - ▶ protocole de communication entre clients et serveur *HTTP*
 - ▶ localisation des documents/données *URI, URL*
 - ▶ description des documents/données *HTML*
 - ▶ description de la mise en forme des données *CSS*

Qu'est-ce que le web ?

- ▶ Un système permettant à des **logiciels** d'échanger des **documents**
 - ▶ service de la couche application
 - ▶ architecture client/serveur
 - ▶ le client demande un document (**une page web**) *requête*
 - ▶ le serveur la lui envoie *réponse*
 - ▶ construit pour être **universel** = pas lié à un seul mode de représentation
 - ▶ **accessibilité** (handicap, automatisation, etc.)
 - ▶ seul les **données** sont importantes dans les documents
- ▶ Logiciels
 - ▶ Client web **Navigateur**
 - ▶ Mozilla **Firefox**, Google Chrome / **Chromium**, Apple Safari, Microsoft Edge, etc.
 - ▶ Aide l'utilisateur à faire les requêtes
 - ▶ Présente les réponses, documents reçus (mise en forme)
 - ▶ Serveur web
 - ▶ **Apache**, **nginx**, Microsoft IIS, etc.
 - ▶ Stocke ou prépare des documents
- ▶ Standards et formats
 - ▶ protocole de communication entre clients et serveur *HTTP*
 - ▶ localisation des documents/données *URI, URL*
 - ▶ description des documents/données *HTML*
 - ▶ description de la mise en forme des données *CSS*

Qu'est-ce que le web ?

- ▶ Un système permettant à des **logiciels** d'échanger des **documents**
 - ▶ service de la couche application
 - ▶ architecture client/serveur
 - ▶ le client demande un document (**une page web**) *requête*
 - ▶ le serveur la lui envoie *réponse*
 - ▶ construit pour être **universel** = pas lié à un seul mode de représentation
 - ▶ **accessibilité** (handicap, automatisation, etc.)
 - ▶ seul les **données** sont importantes dans les documents
- ▶ Logiciels
 - ▶ Client web **Navigateur**
 - ▶ Mozilla **Firefox**, Google Chrome / **Chromium**, Apple Safari, Microsoft Edge, etc.
 - ▶ Aide l'utilisateur à faire les requêtes
 - ▶ Présente les réponses, documents reçus (mise en forme)
 - ▶ Serveur web
 - ▶ **Apache**, **nginx**, Microsoft IIS, etc.
 - ▶ Stocke ou prépare des documents
- ▶ Standards et formats
 - ▶ protocole de communication entre clients et serveur *HTTP*
 - ▶ localisation des documents/données *URI, URL*
 - ▶ description des documents/données *HTML*
 - ▶ description de la mise en forme des données *CSS*

1. Introduction

2. Documents numériques

3. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

4. DNS

Principes

Fonctionnement

5. World Wide Web

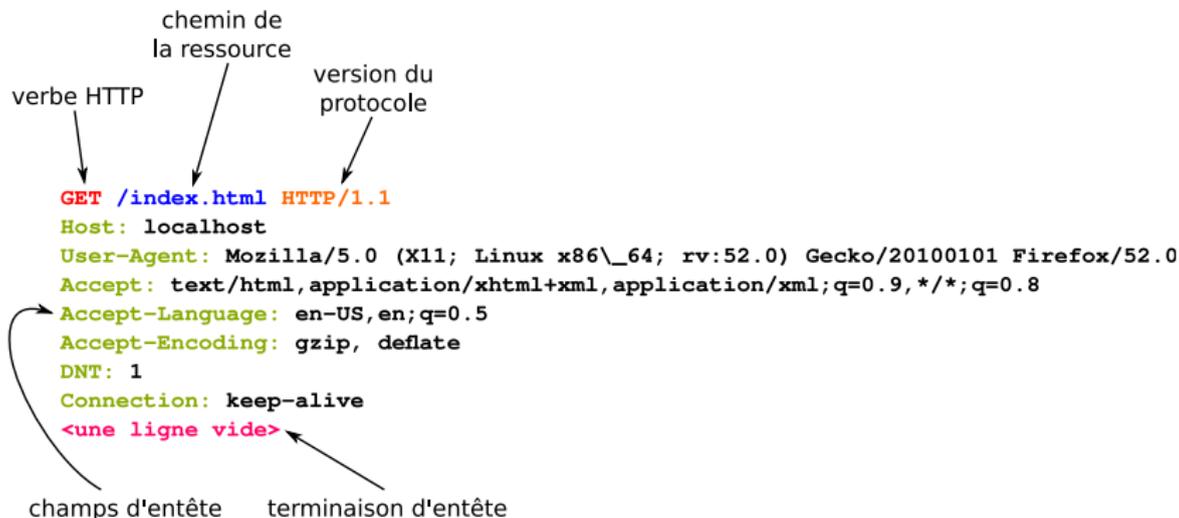
Introduction

HTTP

Site web

- ▶ HTTP est un protocole de manipulation de ressources web
 - ▶ *HyperText Transfer Protocol*
 - ▶ développé au début des années 1990
 - ▶ initialement les ressources sont des documents hypermédia (avec des liens vers d'autres documents)
 - ▶ le protocole permettait uniquement de récupérer une ressource
- ▶ Protocole de communication applicatif **déconnecté** *TCP, port 80*
- ▶ Architecture Client/Serveur
 - ▶ Analogie avec les clients et serveurs d'un restaurant

Format d'une requête HTTP



Format d'une réponse HTTP

version du
protocole

code résultat

HTTP/1.1 200 OK

Date: Tue, 26 Sep 2017 12:32:15 GMT

Server: Apache/2.4.10 (Debian)

Last-Modified: Mon, 19 Sep 2016 07:26:08 GMT

ETag: "29cd-53cd739c04ce7"

Accept-Ranges: bytes

Content-Length: 10701

Vary: Accept-Encoding

Connection: close

Content-Type: text/html

<une ligne vide>

<!DOCTYPE html ...>

<html>

<head>

...

</html>

terminaison d'entête

corps de
la réponse

champs d'entête

Méthodes HTTP

- ▶ **GET** Récupère une représentation de la ressource auprès du serveur
- ▶ **HEAD** Récupère uniquement les informations sur la ressource
- ▶ **POST** Soumettre une entité (corps du message) à la ressource visée
- ▶ **PUT** Créé ou met à jour (remplace) une nouvelle ressource
- ▶ **DELETE** Détruit la ressource associée
- ▶ **OPTIONS** Permet de connaître les options de communication
- ▶ **PATCH** Créer ou met à jour une nouvelle ressource

Le **code de statut** donne au client une indication sur le résultat du traitement de la requête.

- ▶ 1xx – information
 - ▶ Le traitement est en cours. Le client doit attendre un résultat définitif
- ▶ 2xx – succès
 - ▶ La requête a été traitée avec succès par le serveur
- ▶ 3xx – redirection
 - ▶ Une action supplémentaire est nécessaire pour compléter la requête
- ▶ 4xx – erreur du client
 - ▶ La requête est incorrecte (mal formée) ou non autorisée
- ▶ 5xx – erreur du serveur
 - ▶ La requête est valide mais le serveur n'a pas pu la réaliser

Entêtes de requête

Fournis par le client pour

- ▶ fournir des informations sur le contexte de la requête,
- ▶ obtenir un traitement conditionnel en fonction de l'état de la ressource,
- ▶ définir les formats préférés, etc.

Entêtes de réponse

Permet au serveur de fournir des informations complémentaires au statut de la réponse :

- ▶ à propos du serveur,
- ▶ **des prochains accès à la ressource**
- ▶ à propos de ressources associées.

Serveur HTTP

- ▶ respecte HTTP
 - ▶ protocole de communication applicatif **déconnecté sans état**
- ▶ reçoit une **requête** :
 1. méthode, ressource, protocoles
 2. entêtes

TCP, port 80

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

- ▶ produit une **réponse** :
 1. protocole, code, message
 2. entêtes
 3. fichiers

```
HTTP/1.1 200 OK
Content-type: text/html

<html>
<head><title>Mon titre</title></head>
<body><p>Un texte</p></body>
</html>
```

Serveur HTTP

- ▶ respecte HTTP
 - ▶ protocole de communication applicatif **déconnecté sans état**
- ▶ reçoit une **requête** :
 1. méthode, ressource, protocoles
 2. entêtes

TCP, port 80

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

- ▶ produit une **réponse** :
 1. protocole, code, message
 2. entêtes
 3. fichiers

```
HTTP/1.1 200 OK
Content-type: text/html

<html>
<head><title>Mon titre</title></head>
<body><p>Un texte</p></body>
</html>
```

- ▶ équivalent d'un fichier texte stocké par le client pour le serveur
- ▶ solution pour échapper à l'aspect **déconnecté** et **sans-état** de HTTP
 1. le client envoie des informations au serveur
 2. le serveur les synthétise (sérialisation/hachage/etc.)
 3. le serveur les met dans un cookie chez le client
 4. lors de la connexion suivante le serveur demande le cookie
- ▶ permet : authentification, session, identification du client, panier, pistage
- ▶ transfert
 - ▶ données envoyées par le serveur au client dans les entêtes de réponse

```
Set-Cookie: nom=nouvelle_valeur; expires=date; path=/; domain=.exemple.org
```

- ▶ données retournées par le client au serveur à sa demande

Cookies (exemple)

1. Requête du navigateur vers le serveur

```
GET /index.html HTTP/1.1  
Host: www.facebook.com
```

2. Réponse du serveur au navigateur

```
HTTP/1.1 200 OK  
Content-type: text/html  
Set-Cookie: name=value
```

3. **Toutes les requêtes suivantes** faites au même serveur seront de la forme

```
GET /toto.html HTTP/1.1  
Host: www.facebook.com  
Cookie: name=value  
Accept: */*
```

Généralement le Set-Cookie ajouté par un gestionnaire dynamique

Exercice

1. Déterminer des informations sur la page web <http://www.lemonde.fr>
 - ▶ le nombre de requêtes nécessaires à l'affichage de la page
 - ▶ le nombre de cookies envoyés au site lui même
 - ▶ le nombre de cookies envoyés à des sites tiers
2. Faites le même travail pour la page web <http://www.univ-lyon2.fr>
3. Installer les extensions suivantes dans votre navigateur
 - ▶ [Privacy Badger](#)
 - ▶ [uBlock Origin](#)
4. Pour les deux pages précédentes, déterminer
 - ▶ le nombre de pisteurs
 - ▶ le nombre de requêtes nécessaires à l'affichage

1. Introduction

2. Documents numériques

3. Internet (aspect technique)

Principe

Adressage

Transmission de l'information

Applications

4. DNS

Principes

Fonctionnement

5. World Wide Web

Introduction

HTTP

Site web

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

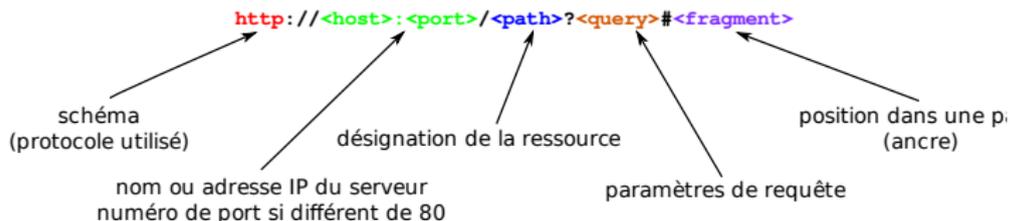
▶ Intermédiaire

- ▶ 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

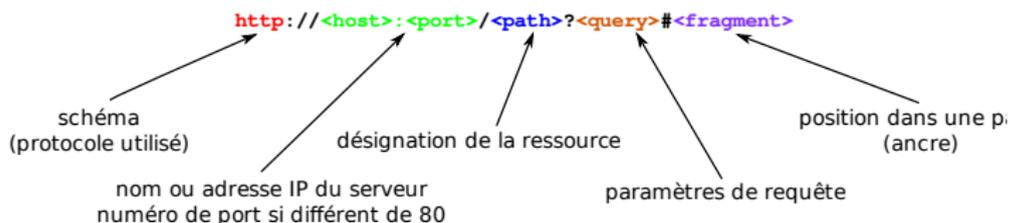
<https://m2-ape-cee.fr/index.html>

Adresse web

- ▶ 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