Applications réseau
Cours 7

Slides : Damien Imbs

Corentin Travers
corentin.travers@univ-amu.fr

AMU - L3 info

2021-2022

IPv4/IPv6

IPv4

Version historique

- Adresses sur 32bits
- 4×1 octet : `145.23.90.246`
- 4,3 milliards d'adresses/ 20,6 milliards d'appareils connectés (2020)

Historiquement, découpage en classes :

- Réseaux de Classe A : préfixe de 8bits, 2^{24} adresses par réseau
- Réseaux de Classe B : préfixe de 16bits, 2^{16} adresses par réseau
- Réseaux de Classe C : préfixe de 24bits 2^8 adresses par réseau
- Classe D : adresses multicast
- Classe E : adresses réservées

$\longrightarrow$ gaspillage de nombreuses adresses

IPv4

Division en classes trop restrictive :
$\Rightarrow$ Utilisation de CIDR (Classless Inter-Domain Routing)

Une adresse IPv4 est divisée en deux partie :

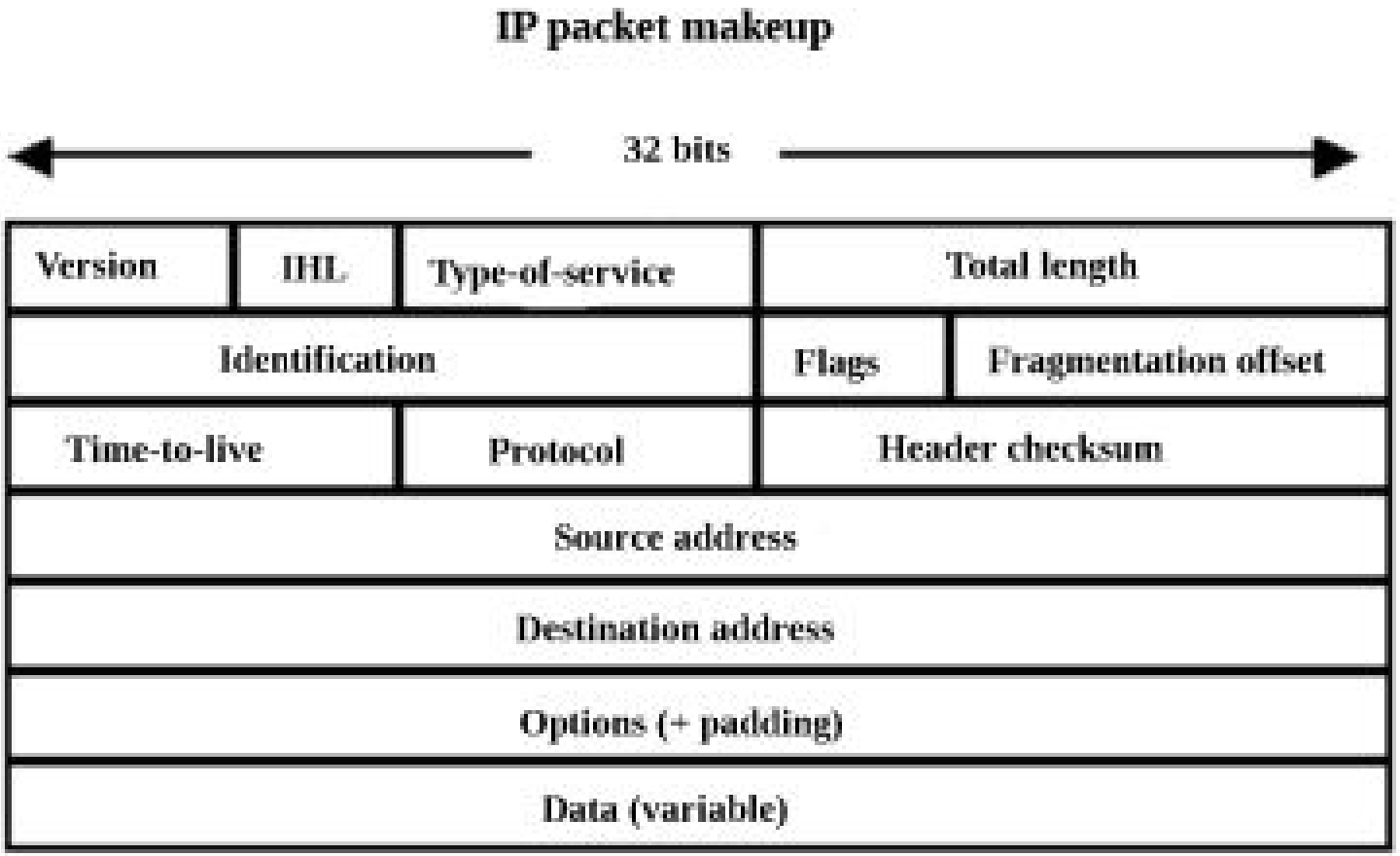
- Une adresse réseau, les premiers bits de l'adresse :
  00101101 11010001 00010011 10100111
- Une adresse machine, les derniers bits :
  00101101 11010001 00010011 10100111

On utilise un **masque réseau** pour distinguer les deux parties :
11111111 111 00000 00000000 00000000
00101101 110 10001 00010011 10100111

Notation CIDR : nombre de bits à 1 du masque
45.209.19.167/11

IPv4

- En-tête IPv4 (20 octets + options) :



IPv6

- Beaucoup plus d'adresses (2^{128})
- Notation hexadécimale
  - Exemple :
    2001 :0db8 :0000 :85a3 :0000 :0000 :ac1f :8001
  - Possibilité de supprimer les 0 non significatifs par groupes de 1 à 3 :
    2001 :db8 :0 :85a3 :0 :0 :ac1f :8001
  - On peut supprimer une fois des blocs contigus de 0 :
    2001 :db8 :0 :85a3 : :ac1f :8001

IPv6

On conserve la notation CIDR :

- adresse/taille du préfixe
- Exemple : 2001 :db8 :1f89 : :/48

IPv6

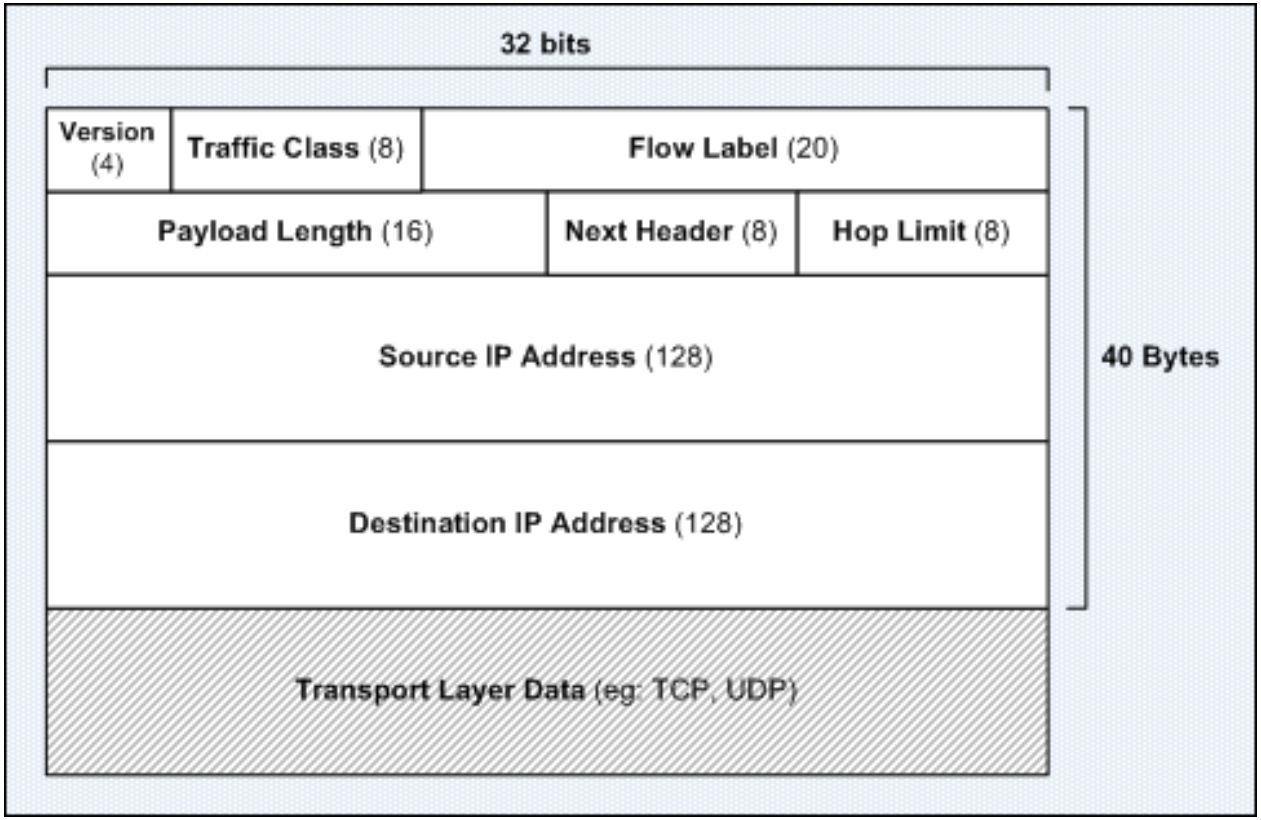
Types d'adresses :

Préfixe	Description
::/8	Adresses réservées
2000::/3	Adresses unicast routables sur Internet
fc00::/7	Adresses locales uniques
fe80::/10	Adresses liens locaux
ff00::/8	Adresses multicast

localhost, en IPv6, est à l'adresse : :1/128

IPv6

En-tête IPv6 (40 octets) :



IPv6 et Java

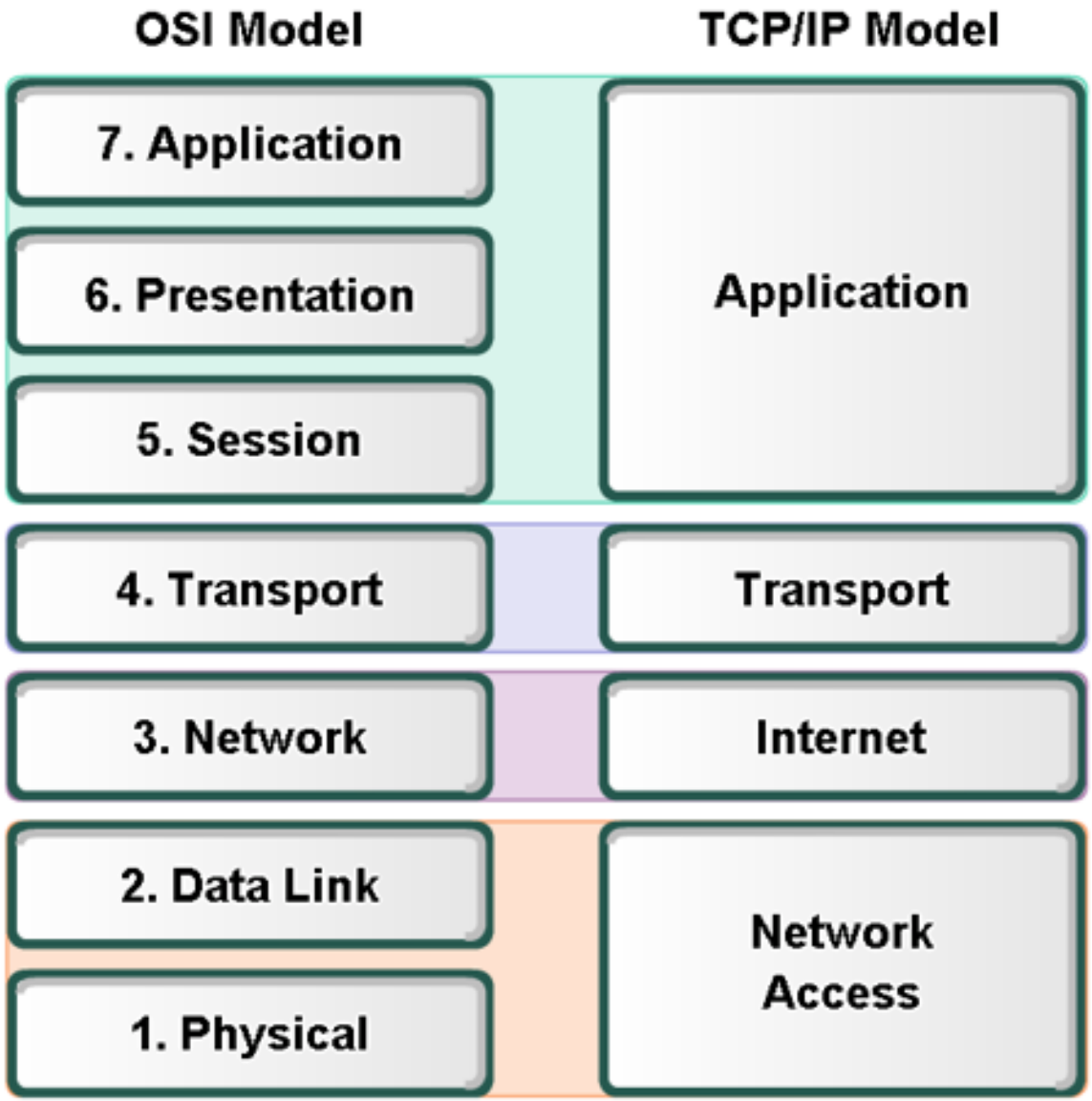
En Java, l'adaptation à IPv6 est automatique et transparente. Une application Java devrait automatiquement être compatible IPv6 si :

- pas d'adresses codées en dur
- Toute l'information sur les adresses et les sockets passe par l'API java.net

Attention :
ce n'est pas le cas pour tous les langages (par exemple C)

Quelques protocoles de la couche application

Rappel : réseaux en couches



Couche Application

LA COUCHE APPLICATION EST LA COUCHE FINALE.

Elle offre de très, très nombreux protocoles. Les protocoles que nous allons voir ici sont à la fois les plus importants historiquement, dans l'utilisation actuelle, ou les plus représentatifs.

Couche Application

Couche finale pour la communication de l'information => de très nombreux protocoles :

- HTTP, FTP,
- SMTP, POP, IMAP,
- DNS
- Jabber, MSN, ICQ, H323,
- SMB, NFS, ...

- ...
- intergiciels
- RPC, RMI, CORBA, DCOM, ...

La plupart des protocoles sont construits au dessus de TCP qui apporte la fiabilité nécessaire.

Protocoles Etudiés

- HTTP : le protocole du web
  TCP port 80
- SMTP : envoi de courriel
  TCP port 25
- DNS : protocole et base de données distribuée pour la résolution de nom
  UDP/TCP port 53

Protocole HTTP

Hyper Text Transfer Protocol (HTTP

- Né au CERN à la fin des années 80
- Requêtes/reponse simples, Sans état
- Port classique: 80 /443 (HTTPS)
- RFCs 1945 et 2048
- Versions :
  - HTTP 1.0 : très simple
  - HTTP 1.1 : amélioré (connexions persistantes)
  - HTTP 2.0 : multiplexage, latence moindre
  - Internet Draft HTTP 3.0

URLs et URIs

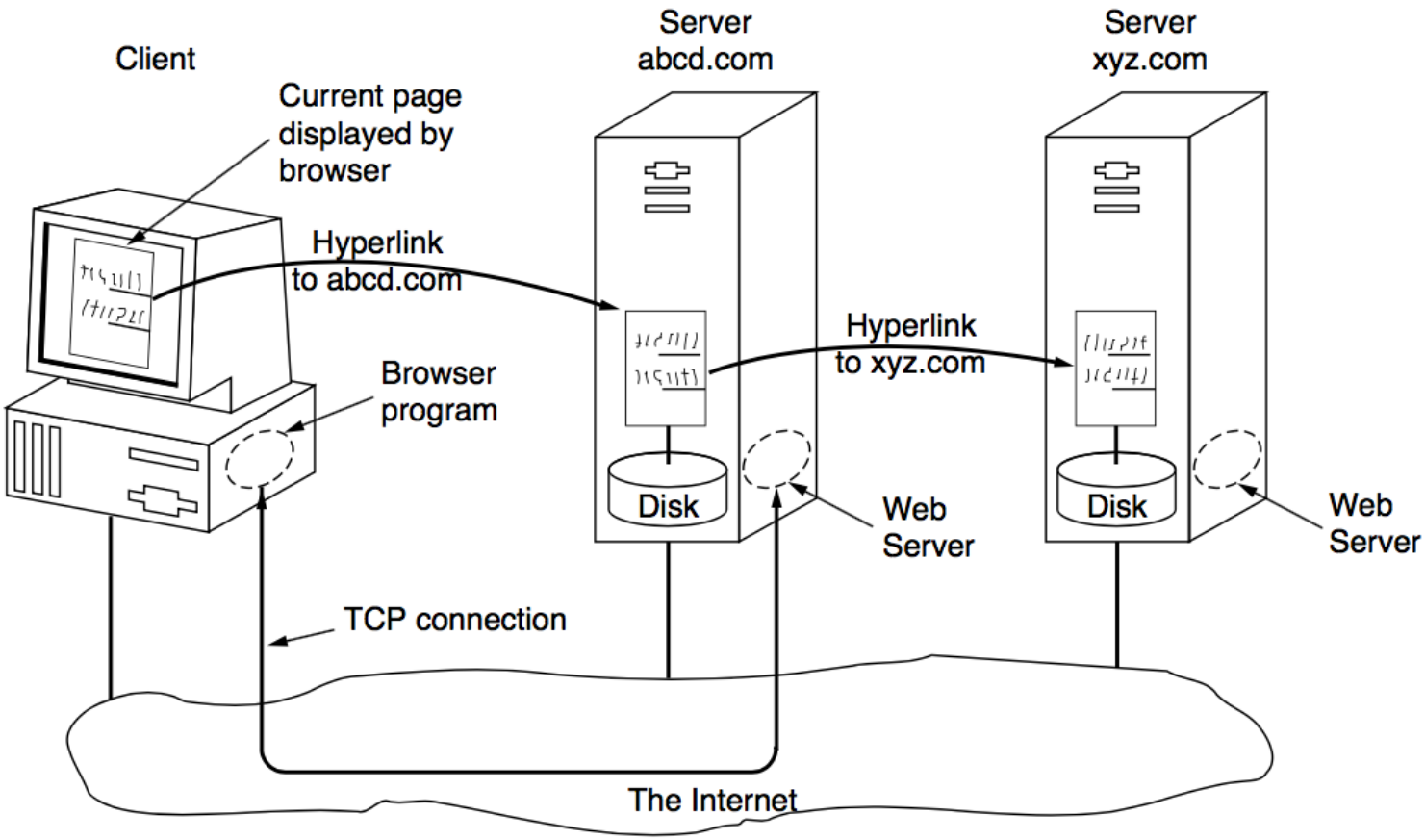
- URI : Uniform Resource Identifier
  - Chaîne de caractère unique qui identifie une ressource
  - ressource : fichier, livre, email, machine
    - `mailto:damien.imbs@univ-amu.fr`
    - `urn:isbn:096139210x`
    - `ftp://dimbs@mimosa.univ-mrs.fr/`
    - `file:/home/bob/my_file`
    - `http://www.example.com`
- URL : Uniform Ressource Locator
  - Identification d'une ressource et où en trouver une représentation
  - Ie. Adresse réseau + protocole pour récupérer une représentation de la ressource
  - `http://www.example.com/networking/ip.html`
    Fichier nommé `ip.html` dans un répertoire `networking` sur le serveur `http://www.example.com`, et que ce fichier est accessible par le protocole HTTP.

URLs

`protocol://userInfohost:port/path?query#fragment`

- `protocol` HTTP, FTP, HTTPS, etc.
- `host` www.example.com, 214.10.53.25
- `userInfo`, optionnel : login information
- `port` optionnel
- `path` chemin vers la ressource sur le serveur
- `?query` arguments pour le serveur (eg données de formulaire)
- `fragment` section particulière de la ressource

La Toile


Client
Current page
displayed by
browser
Hyperlink
to abcd.com
Browser
program
Server
abcd.com
Disk
Web
Server
Hyperlink
to xyz.com
Server
xyz.com
Disk
Web
Server
TCP connection
The Internet

HTTP 1.0

- Protocole en ligne de caractères (diagnostic plus facile)
- Requête :

```
GET /~egodard/ens/reseaux/index.html HTTP/1.0
Accept: text/html
Accept: image/png
User-Agent: telnet
```

Réponse

```
HTTP/1.0 200 OK
Date: Sun, 13 Nov 2005 13:20:04 GMT
Server: Apache/1.3.33 (Unix) mod_ssl/2.8.22 OpenSSL
Last-Modified: Thu, 10 Nov 2005 14:47:14 GMT
ETag: "b16038-1e4c-43735d72"
Accept-Ranges: bytes
Content-Length: 7756
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-W3CDTD HTML 4.01//EN">
<html>
<head>
  <META http-equiv="Content-Type" content="text/htm
  <title>E. Godard -- Réseaux</title>
  <style type="text/css">
  ...
```

Elément A l'Intérieur d'Une Page

Un élément interne (comme une image) est redemandé par le navigateur après analyse du code html.

Différence entre

- protocole : HTTP
- encodage : HTML

Types MIME

Type	Subtype	Description
Text	Plain	Unformatted text
	Enriched	Text including simple formatting commands
Image	Gif	Still picture in GIF format
	Jpeg	Still picture in JPEG format
Audio	Basic	Audible sound
Video	Mpeg	Movie in MPEG format
Application	Octet-stream	An uninterpreted byte sequence
	Postscript	A printable document in PostScript
Message	Rfc822	A MIME RFC 822 message
	Partial	Message has been split for transmission
	External-body	Message itself must be fetched over the net
Multipart	Mixed	Independent parts in the specified order
	Alternative	Same message in different formats
	Parallel	Parts must be viewed simultaneously
	Digest	Each part is a complete RFC 822 message

Requêtes

- `GET` : informations et données de l'URI
- `HEAD` : seulement les informations d'entêtes
- `POST` : envoi de données de formulaire
- `PUT` : enregistrement ("upload") de données
- `DELETE` : effacement
- `OPTIONS` : demande des options de communication
- ...

Méthode POST

```
POST /~egodard/ens/reseaux/index.html HTTP/1.0
Accept:text/html
Accept:image/png
User-Agent: firefox/linux ppc

champs1=valeur1&
champs2=valeur2
```

Statuts de Retour

- 100-199 : informationnel
- 200 : OK
- 201 : créé
  200-299 : succès
- 301 : redirection
- 304 : non modifié
  300-399 : redirections
- 400 : mauvaise requête
- 403 : interdit
- 404 : non trouvé
  400-499 : requête incorrecte
- 500 : erreur interne du serveur
- 503 : service indisponible pour l'instant
  500-599 : erreurs côté serveur

Gestion de Session

Le protocole est sans mémoire : comment associer une session à plusieurs requêtes consécutives ?
Utilisation de cookie

- `Set-Cookie:nom=valeurunique` côté serveur
- `Cookie:nom=valeurunique` dans les entêtes des requêtes suivantes
- Pratique pour les clients (achats en ligne, forums, ...)
- Très pratique pour la veille marketing
  => problème vis-à-vis vie privée
  - désactivation / contrôle des cookies
  - code de bonne conduite
  - ... ?

Mandataire (proxy) HTTP

On utilise des mandataires pour

- partager un cache (vers des requêtes fréquentes)
- authentifier les connexions sortantes
- contrôle de contenu
- réécrire les pages à la volée (publicité, requête espionne, cookies ...)

Des informations sont contenus dans les entêtes à destination des mandataires :

- `Last-Modified` : date (pour comparer à la dernière copie)
- `Expires` : date de péremption
- `Pragma` : contenu volatil ou non
- `Via` : mandataires intermédiaires

La requête devient :

```
GET http://hote/chemin
```

HTTP 1.1

- RFC 2616 (1999)
- nouvelle version pour connaître exactement les propriétés du correspondant
  permet d'héberger plusieurs sites sur le même serveur :

```
GET http://www.example1.com/home.html HTTP/1.1
```

vs

```
GET http://www.example2.com/site.php HTTP/1.1
```

- connexion persistante :
  - 100 : continuer

HTTP/2

Nouvelle version

- Basé sur SPDY
  - réduction de la latence par réutilisation de la même socket TCP
  - binaire / encapsulation de messages
  - multiplexage
- chiffrement (TLS 1.2) obligatoire (ou pas ...)
- RFC 7540 (mai 2015)
- support logiciel : IE 11, Firefox 34, Chrome

WebDAV

Web Distributed Authoring and Versioning
Surcouche de HTTP pour gérer la publication avancée :

- révisions et commentaires
- contrôle d'accès
- information d'indexation

=> logiciels collaboratifs et nombreux clients.

SMTP

Protocole SMTP

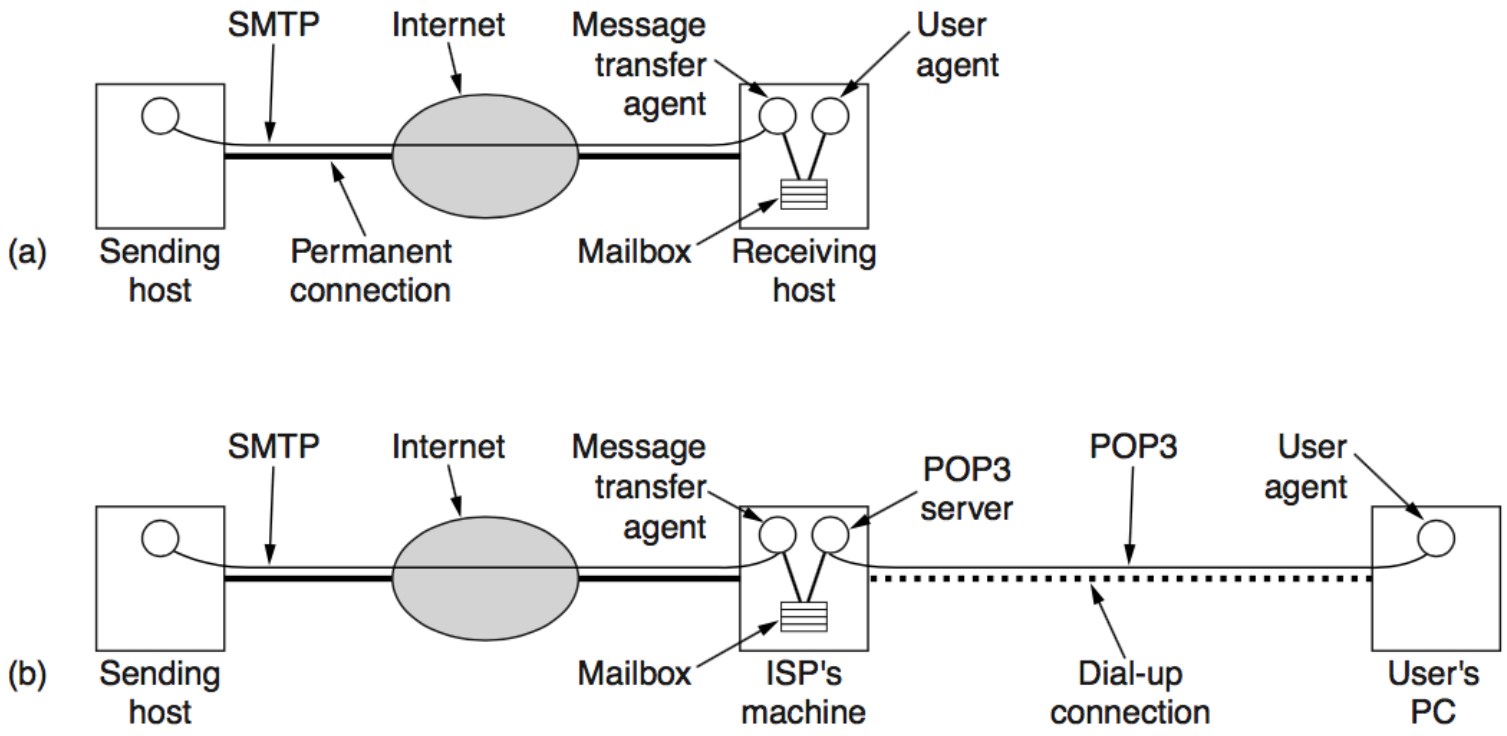
Simple Mail Transfer Protocol

- Envoi/transport de courrier électronique
- RFC 821 (1982)
- RFC 2821 (2001)

Agents du Courrier Electronique

- MUA (Mail USer Agent) : client de messagerie
- MTA (Mail Transfer Agent) : transport des courriers (via des serveurs SMTP en général)
- MDA (Mail Delivery Agent) : remise du courrier au destinataire final

Architecture du Courrier Electronique



Exemple de Transport de Courrier

- L'utilisateur clique sur "Envoyer"
- Le logiciel de messagerie(MUA) contacte le serveur SMTP local(MTA)
- Ce serveur contacte le serveur SMTP distant (MTA) associé à l'adresse du destinataire
- Celui-ci transmet le courrier au serveur Pop (MDA)
- Le destinataire clique sur le bouton "Recevoir" de son logiciel de messagerie (MUA).

MDA : POP3 vs IMAP

Feature	POP3	IMAP
Where is protocol defined	RFC 1939	RFC 2060
TCP port used	110	143
Where is e-mail stored	User's PC	Server
Where is e-mail read	Off-line	On-line
Connect time required	Little	Much
Use of server resources	Minimal	Extensive
Multiple mailboxes	No	Yes
Who backs up mailboxes	User	ISP
Good for mobile users	No	Yes
User control over downloading	Little	Great
Partial message downloads	No	Yes
Are disk quotas a problem	No	Could be in time
Simple to implement	Yes	No
Widespread support	Yes	Growing

Courrier Electronique et Espace de Noms

Une adresse électronique `identifiant@domaine.tld` est constituée de

- `identifiant` local du destinataire
- `domaine.tld`
  - nom de la machine MTA du destinataire
  - ou domaine DNS dont le champ `MX` indique le MTA

Exemples (identiques côté utilisateur mais pas côté serveur)

```
aconnu@cmi.univ-mrs.fr

Alain.Connu@cmi.univ-mrs.fr

Alain.Connu@lif.univ-mrs.fr

Alain.Connu@up.univ-mrs.fr

Alain.Connu@univ-provence.fr
```

Envoi par Poignée de Main

```
serveur: 220 smtp.exemple.net SMTP Ready
client : EHLO station1.exemple.net
serveur: 250 smtp.exemple.net
client : MAIL FROM:<src@exemple.net>
serveur: 250 OK
client : RCPT TO:<dst@security.net>
serveur: 250 OK
client : DATA
serveur: 354 Start mail input; end with <CRLF>.<CRLF>
client : Subject: Bonjour
client : Salut,
client : comment ca va?
client :
client : A bientot !
client : <CRLF>.<CRLF>
serveur: 250 OK
client : QUIT
serveur: 221 smtp.exemple.net closing transmission
```

Encodage et Extension

- les caractères transmis doivent être codés sur 7bits
- format MIME (Multipurpose Internet Mail Extension)
- encodage base64 ou quoted-printable

Entête d'un Courrier Electronique

- `From:` adresse de l'émetteur
- `Return-Path:` adresse pour la réponse (peut être différent du précédent)
- `To:` adresses(s) du(des) destinataire(s) principal(aux)
- `Cc:` adresses(s) du(des) destinataire(s) secondaire(s)
- `Subject:` sujet
- `Received:` ajouté par chaque MTA
- `MessageId:` identifiant unique
- `Date:` date d'envoi du message (par le MUA)
- `X-...:` champs officieux

(In)Sécurité

Le pragmatisme l'a emporté (et de loin) sur la sécurité.
Il n' y a absolument aucune garantie sur un courrier électronique

- Pas d'encryptage (par défaut)
- Pas de vérification de l'expéditeur
- D'autres failles...

Pourriel

Le courrier électronique non sollicité (spam, pourriel) est un important problème.

- achat de liste de millions d'adresse
- coût nul de l'envoi, même massif
- un très faible retour sur un envoi massif reste profitable => secteur économique
- relais ouverts
- extension aux messageries instantanées et VoIP...

Lutte anti-pourriels

Le MUA peut filtrer le courrier (une fois qu'il est arrivé).
Au niveau des serveurs :

- certains serveurs refusent systématiquement les connexions en provenance de certains hôtes
  - listes noires/listes blanches
  - listes dynamiques (relais ouverts, ...)
- cela n'est pas prévu dans le protocole

Enjeu : Universalité du courrier électronique
Cf messagerie instantanée pour un contre exemple.

Liste Grise

Méthode respectant rigoureusement la RFC, en jouant sur le fait que les délais prévus sont larges (5 jours au pire) :

- le serveur SMTP refuse pendant un certain temps un nouvel expéditeur donné,
- un spammeur ne réessaie pas sur une erreur
- un serveur “normal” réessaiera (avec succès) ultérieurement
- paramétrables :
  - listes blanches/grises
  - temps de refus variable et aléatoire

DNS

DNS : Protocole et BD Distribuée

DNS : Domain Name System
base de données hiérarchique et distribuée permettant la résolution de noms vers les adresses IP (v4 et v6).

1. Hiérarchie : les noms sont structurés et forment une arborescence
2. Zones : ce sont des ensembles disjoints de noms formant des sous-arbres
3. Struct. de données distribuée : Pour chaque zone, un ou plusieurs serveurs DNS sont chargés de stocker les données ou une indirection vers un autre serveur.

Structure Hiérarchique du DNS

La structure de données distribuée est héritée de la structure hiérarchique du DNS : il y a (au moins) un serveur pour chaque zone

- `.fr`
- `univ-amu.fr`
- `dii.univ-amu.fr`

NB le découpage en zone ne suit pas nécessairement le découpage par "."

Base de données

La structure est << clef champ valeur >>

enregistrement A fait correspondre un nom d'hôte à une adresse IPv4 (32 bits)

enregistrement AAAA idem pour IPv6
=>double résolution possible

enregistrement CNAME alias vers un autre nom. Permet de nommer des "services" (`www`, etc ...)

enregistrement NS indique un serveur de nom du domaine. Il en faut au moins 2 pour tolérer les défaillances

enregistrement PTR inverse de A : associe une adresse IP à un enregistrement de nom de domaine,

Base de Données (suite)

enregistrement SOA indique serveur principal et administrateur de la zone

enregistrement MX indique le MTA de courrier du domaine (statut particulier du courriel !)
permet de transmettre un courrier à `untel@lif.univ-mrs.fr` même si aucune machine ne s'appelle `lif.univ-mrs.fr`.

enregistrement SRV généralise le enregistrement MX pour des services quelconques + gestion de la répartition de charge (a priori).

Exemples

Réponse à `dig a www.lif.univ-mrs.fr`

```
; <<>> DiG 9.8.1-P1 <<>> a www.lif.univ-mrs.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28077
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 0

;; QUESTION SECTION:
;www.lif.univ-mrs.fr.          IN  A

;; ANSWER SECTION:
www.lif.univ-mrs.fr.    172800  IN  CNAME   serveur.cmi.up.univ-mrs.fr
serveur.cmi.up.univ-mrs.fr. 86400 IN    A   147.94.64.6

;; AUTHORITY SECTION:
up.univ-mrs.fr.     86400   IN  NS  lsh.up.univ-aix.fr.
up.univ-mrs.fr.     86400   IN  NS  ns1.univmed.fr.
up.univ-mrs.fr.     86400   IN  NS  dns.univ-provence.fr.

;; Query time: 86 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Mon Nov 26 23:38:21 2012
;; MSG SIZE  rcvd: 170
```

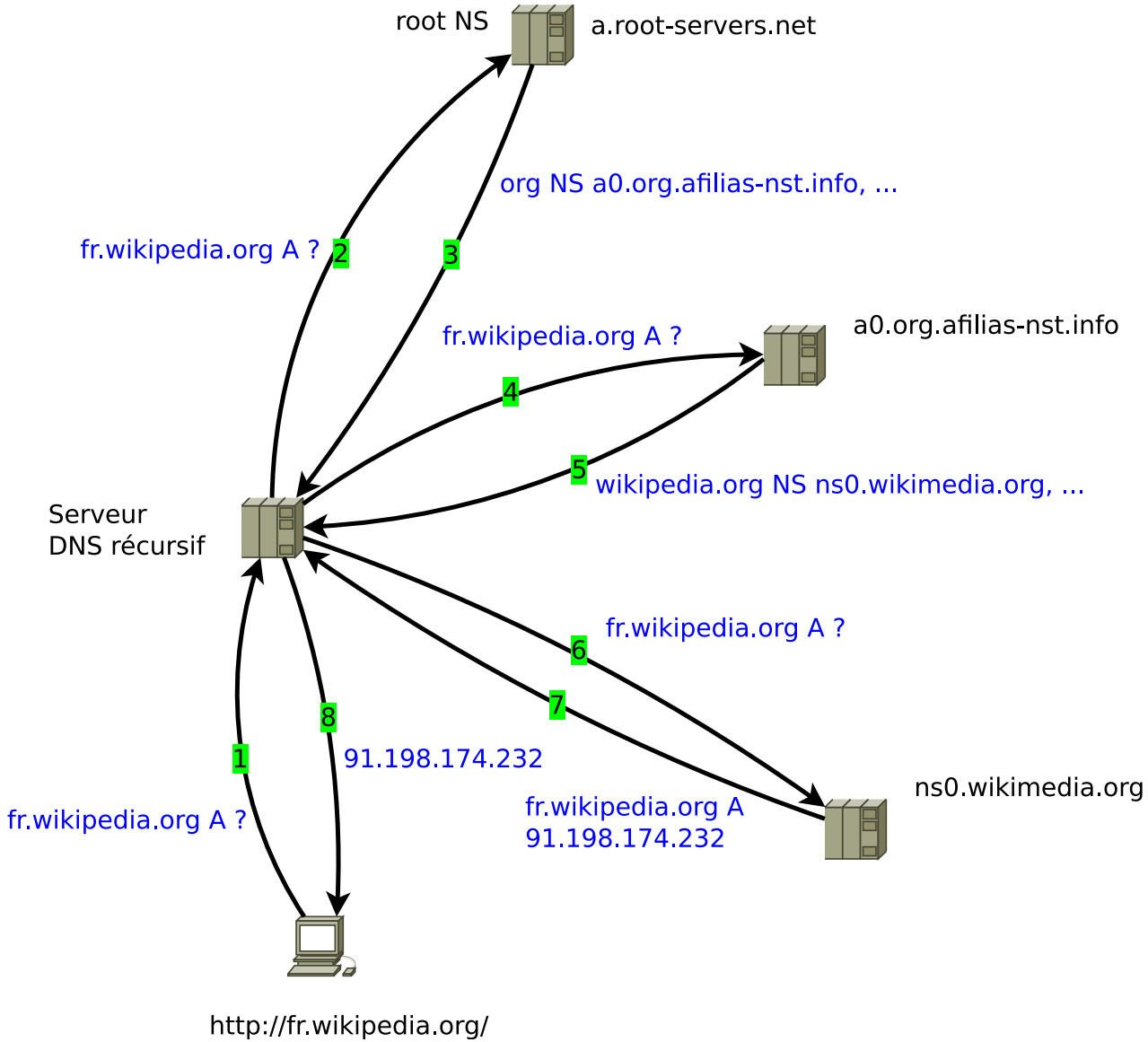
Exemple de Répartition de charge avec SRV

```
_sip._tcp.example.com. 86400 IN SRV 10 60 5060 gros-serveur.example.co
_sip._tcp.example.com. 86400 IN SRV 10 20 5060 petit-serveur1.example.
_sip._tcp.example.com. 86400 IN SRV 10 20 5060 petit-serveur2.example.
_sip._tcp.example.com. 86400 IN SRV 20 0 5060 serveur-de-secours.examp
```

Requête DNS

Une requête (question) porte sur un champ particulier concernant un domaine passé en paramètre. Le protocole UDP est utilisée, sauf si le message dépasse 512 octets.
Dans ce cas, TCP est utilisé.

Interrogations Récursives et Itératives



Mise en cache

Un champs TTL indique combien de temps la valeur d'un enregistrement peut être gardé en cache (ie, les services racines ne sont pas sollicités à chaque requête d'un internaute...)
C'est en général de l'ordre de la journée

Problème de Sécurité

Il est possible d'intercepter une réponse DNS.
=> protocole DNSSEC, où les réponse sont signées cryptographiquement.
Le déploiement est en cours...

RFCs

Request For Comments

- Spécifications techniques
- Documents officiels
- Malgré le nom, documents stables :
  Chaque RFC reçoit un numéro. Après avoir reçu un numéro, plus de modifications possibles.

Exemples :

- RFC 791 : Internet Protocol (IP)
- RFC 959 : FTP
- RFC 4251 SSH

La référence pour les protocoles réseaux

RFCs

Request For Comments

- Tous ne deviennent pas des standards :
  certains sont expérimentaux
  certains sont obsolètes (remplacés par d'autres)
- Présentés en format texte (historique, et lisible partout)
- Disponibles sur le site de l'IETF
  (Internet Engineering Task Force) :
  `https://tools.ietf.org/html/`

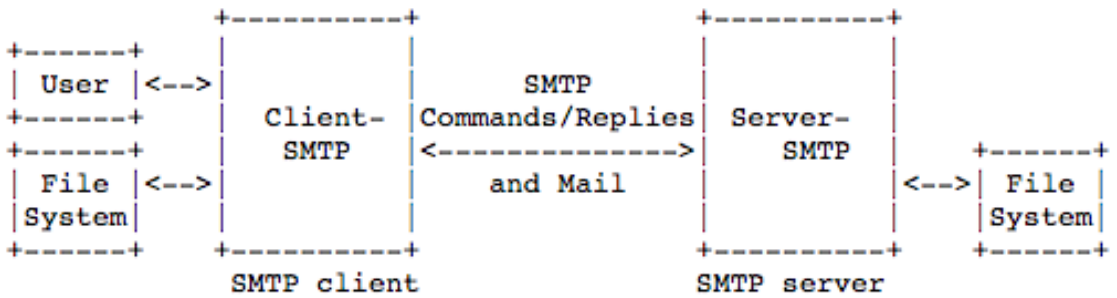
Extrait d'un RFC

illustrative examples that should not actually be used in either code or configuration files.

2. The SMTP Model

2.1. Basic Structure

The SMTP design can be pictured as:



When an SMTP client has a message to transmit, it establishes a two-way transmission channel to an SMTP server. The responsibility of an SMTP client is to transfer mail messages to one or more SMTP servers, or report its failure to do so.