

# Introduction aux Réseaux

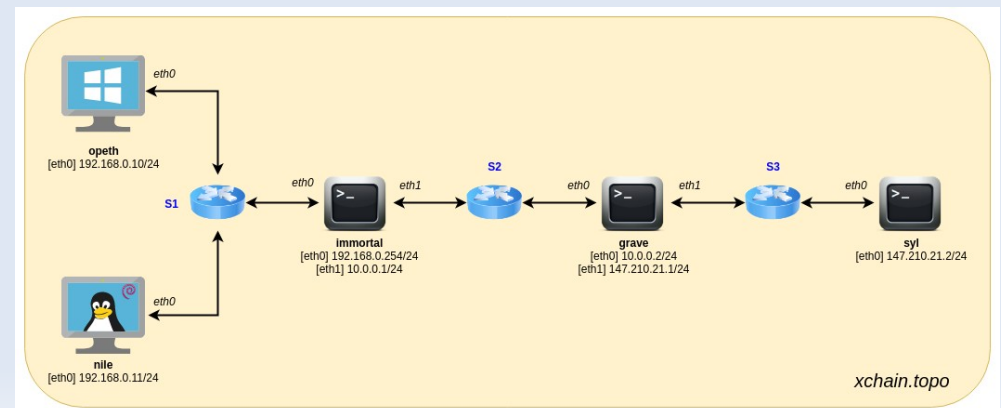
**DIU « Enseigner l'Informatique au Lycée »**

Université de Bordeaux

[aurelien.esnard@u-bordeaux.fr](mailto:aurelien.esnard@u-bordeaux.fr)  
[abdou.guermouche@u-bordeaux.fr](mailto:abdou.guermouche@u-bordeaux.fr)

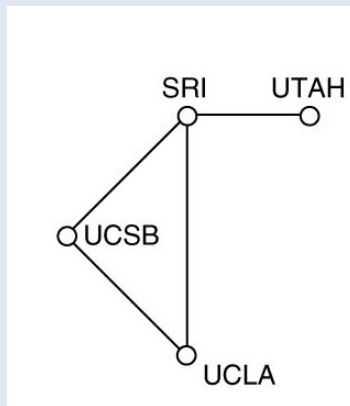
# Plan

- Introduction
- Modèle client/serveur et modèle OSI
- Protocoles de communication d'Internet et du Web
  - Routage, DNS, Web (HTTP), mail (SMTP & IMAP)
- Sécurité des communication
- Les outils réseaux
  - Les commandes en ligne : ping, ifconfig, route, ...
  - Socket en python3, Scapy, Wireshark, ...
- Utilisation de réseaux virtuels

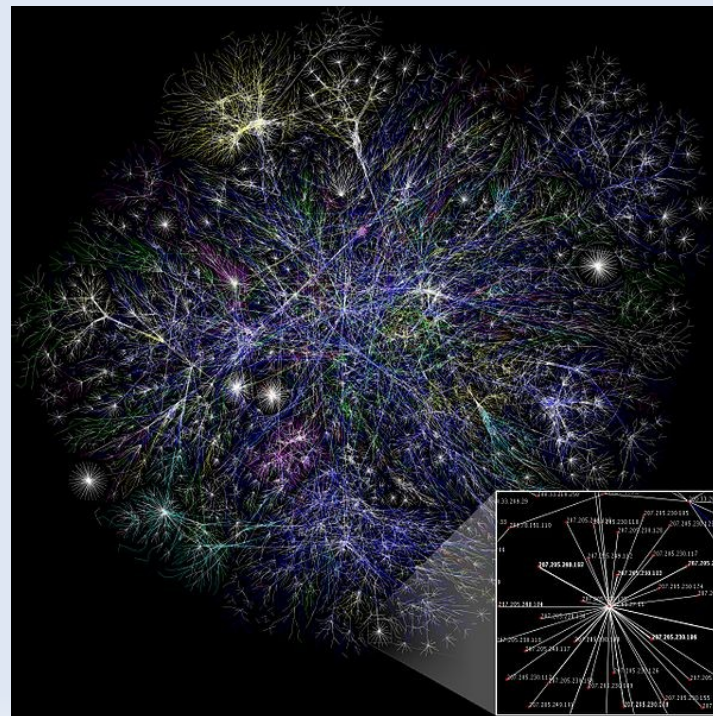


# Internet

- Internet : réseau informatique mondial, résultant de l'interconnexion d'une multitude de réseaux informatiques à travers la planète, unifiées grâce au protocole IP. [1983]
- Protocole réseau : un protocole définit de manière formelle et interopérable l'échange des informations entre ordinateurs.



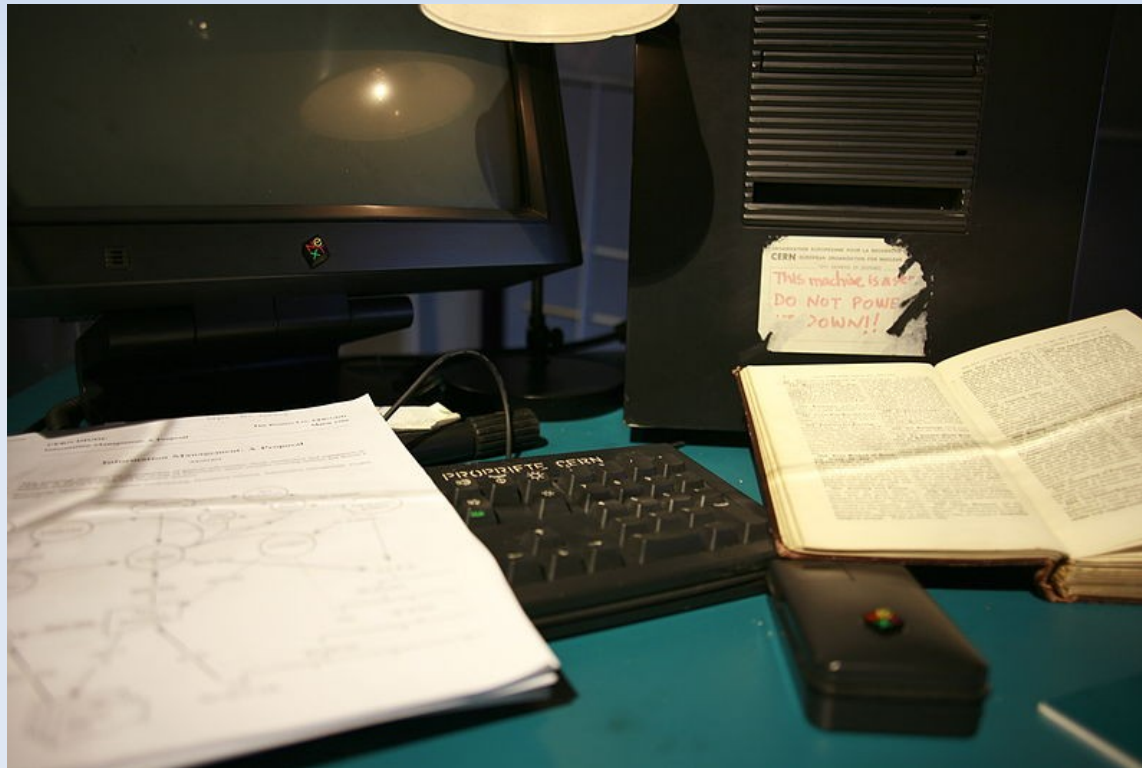
ARPANET. Source : Tannenbaum



Source : Wikipedia

# Web

- Web (ou la toile) : l'ensemble des hyperliens (ou liens hypertextes) qui relient les pages web entre elles. [1990]

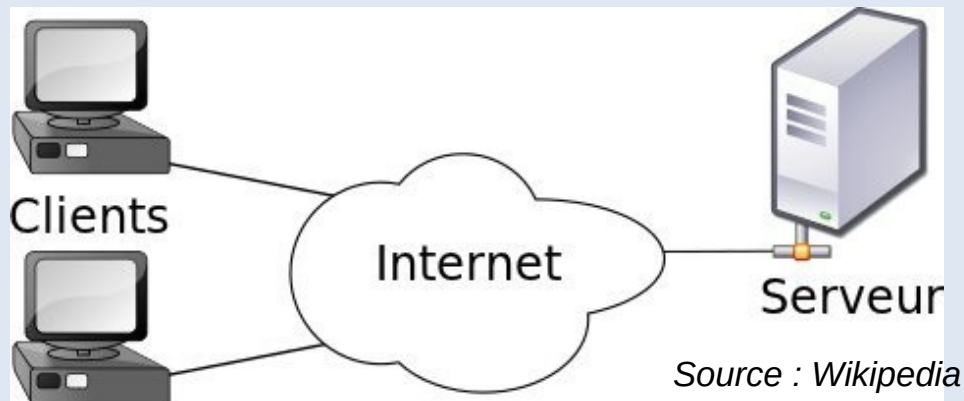


*L'ordinateur utilisé au CERN par Tim Berners-Lee pour inventer le Web. Source : Wikipedia.*

→ Ne pas confondre Internet et le Web, qui est un des nombreux services Internet !

# Web

- Serveur Web : ordinateur qui contient les ressources du Web (pages, media, ...) et les met à disposition sur Internet.
  - Ex. : [www.google.com](http://www.google.com), [fr.wikipedia.org](http://fr.wikipedia.org), ...
- Navigateur Web : logiciel (client du serveur Web) permettant de consulter les ressources du Web.
  - Ex. : Internet Explorer, Firefox, Chromium, ...



# Web

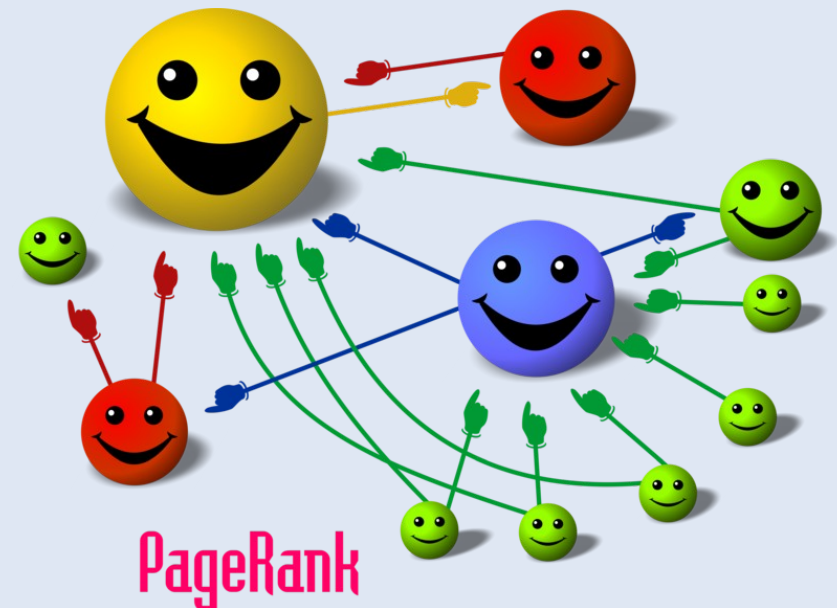
- HTTP (HyperText Transfert Protocol) : protocole de transfert des pages HTML permettant de naviguer sur le Web (HTTPS pour la version sécurisée).
- HTML (Hypertext Markup Language) : langage à balise pour représenter les pages Web (mise en forme, liens hypertextes, ressources multimédias, ...).

```
<!DOCTYPE html PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
  <head>
    <title>
      Exemple de HTML
    </title>
  </head>
  <body>
    Ceci est une phrase avec un <a href="cible.html">hyperlien</a>.
    <p>
      Ceci est un paragraphe où il n'y a pas d'hyperlien.
    </p>
  </body>
</html>
```

*Source : Wikipedia*

# Moteur de Recherche

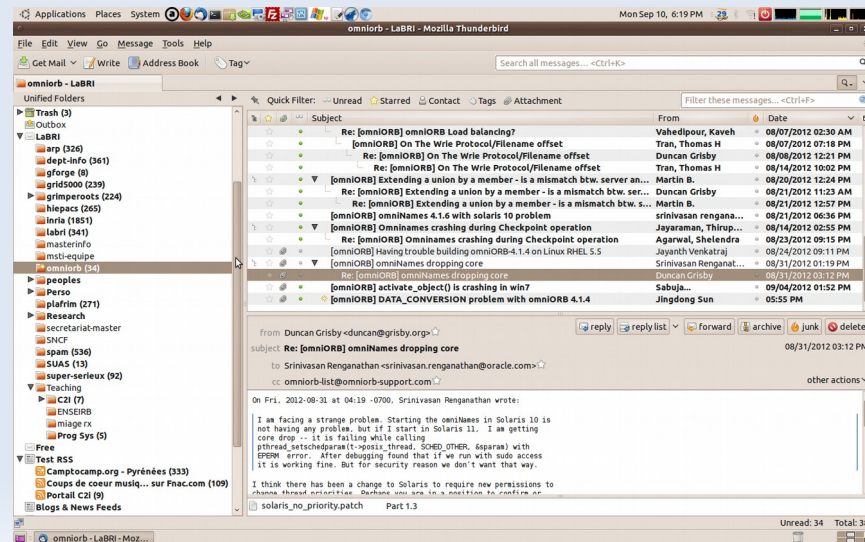
- Moteur de recherche : outil permettant de retrouver des pages Web à partir d'une requête
  - Ex. : Google, Yahoo Search, Bing, ...
- Indexation automatique : les pages du Web sont parcourues automatiquement par un « robot » et analysées pour en extraire des mots-clés significatifs.
- Ordre des réponses : il dépend de l'adéquation des mot-clefs et de la « popularité » de la page web
  - nombre de liens vers la page (PageRank de Google)
  - les clics des utilisateurs sur la page de réponse





# Messagerie Electronique

- Messagerie électronique : outil permettant d'échanger des messages (courriel ou *mail*) de manière asynchrone par l'intermédiaire d'une boîte à lettres électronique identifiée par une adresse électronique.
- Adresse électronique : prenom.nom@etu.u-bordeaux.fr
- Client de messagerie local ou application webmail
  - Ex. : Thunderbird, Outlook, ... vs Gmail, Yahoo!, ...

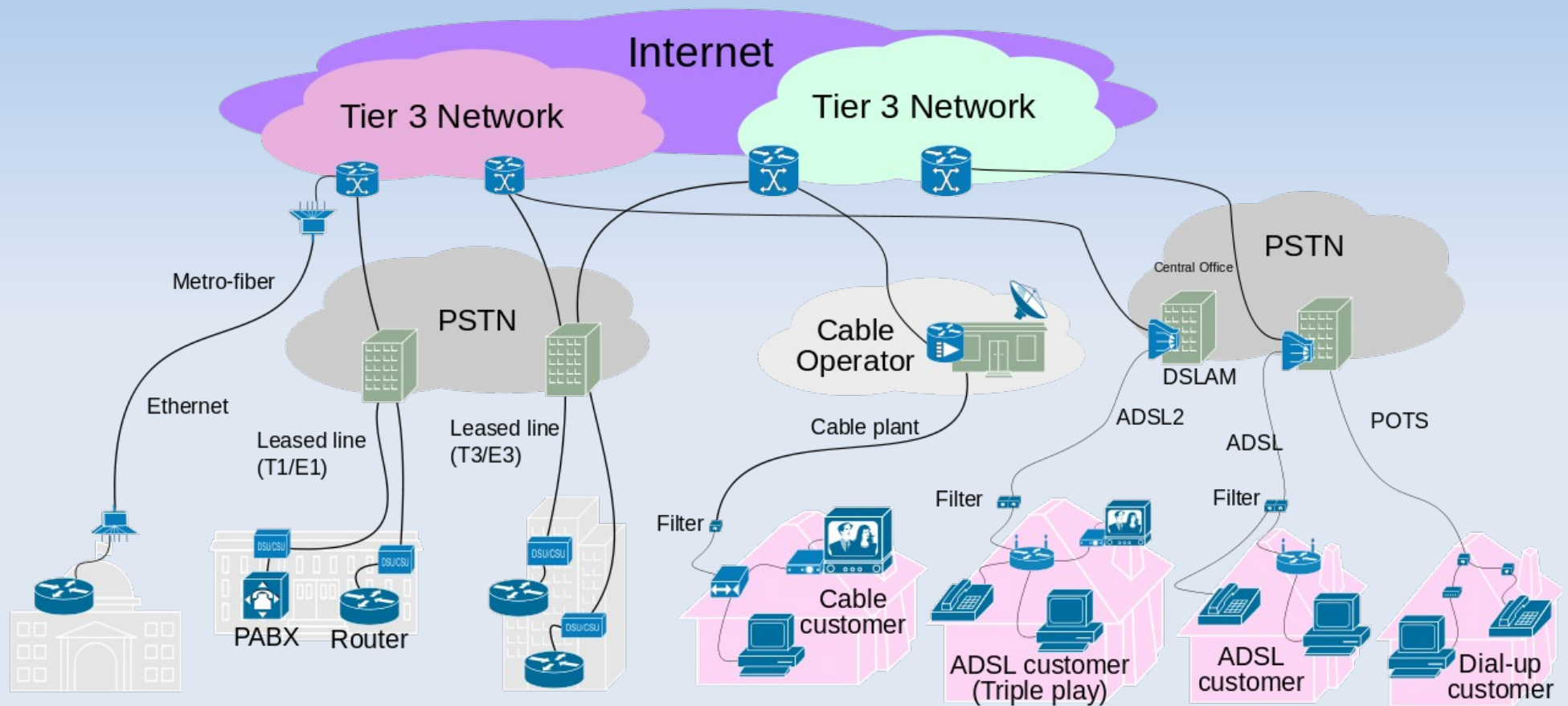




# Latence & Débit

- Débit
  - nombre de bits que le réseau peut transporter par seconde
- Latence
  - nombre de secondes que met le premier bit pour aller de la source à la destination
- Quelques exemples de débits (en bit/s)
  - modem RTC 56K, ADSL (1M à 8M), FTTH (1G)
  - Ethernet (10M, 100M, 1G, 10G), ATM (155M), FDDI (100M), ...
  - sans-fil : IEEE 802.11 (11M à 54M)
  - GSM : 3G (144K-1,9M), EDGE (64k-384k), 3G+ (3,6M, 14,4M), 4G (100M-1G), 5G (10G) ...

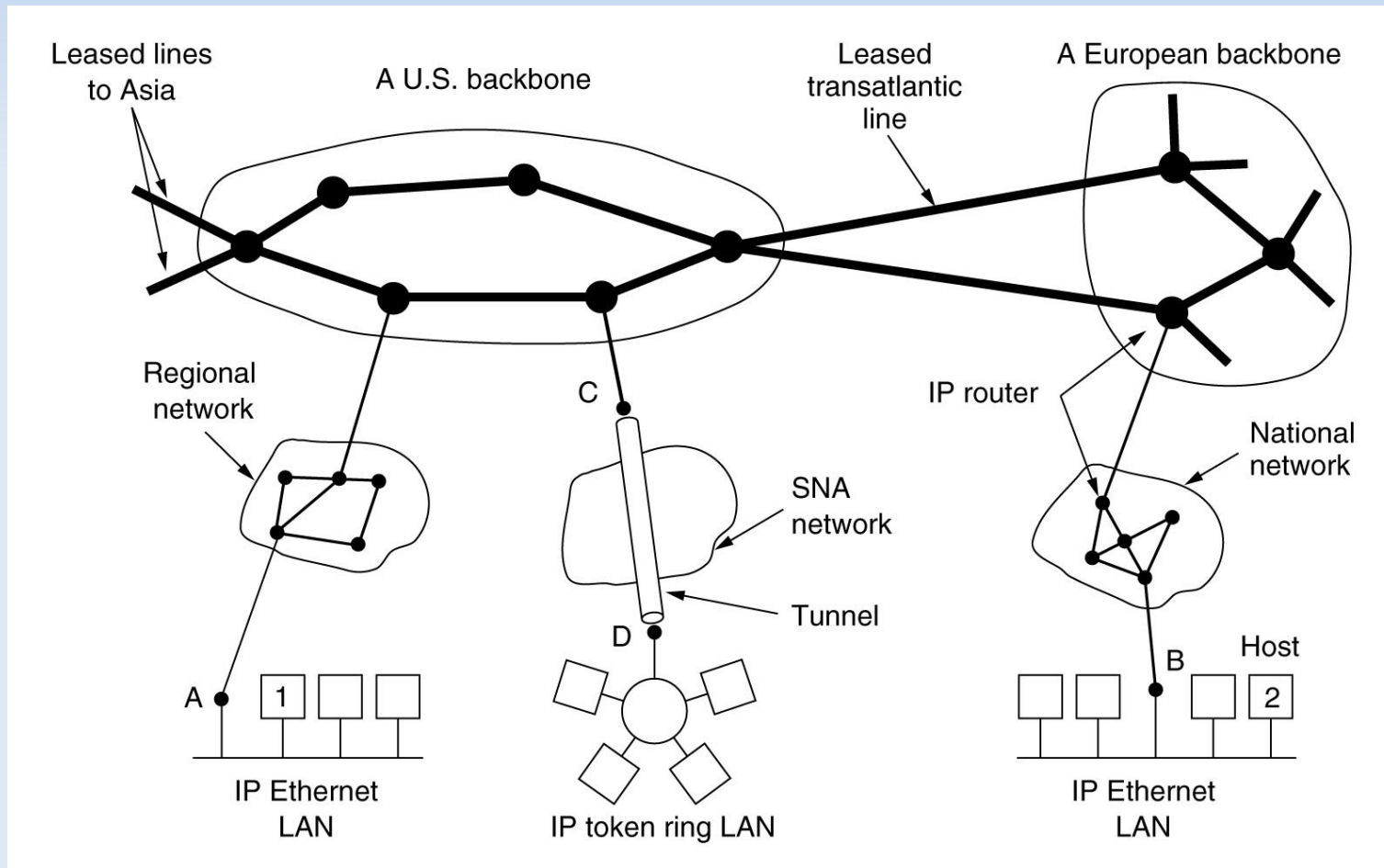
# Structure d'Internet



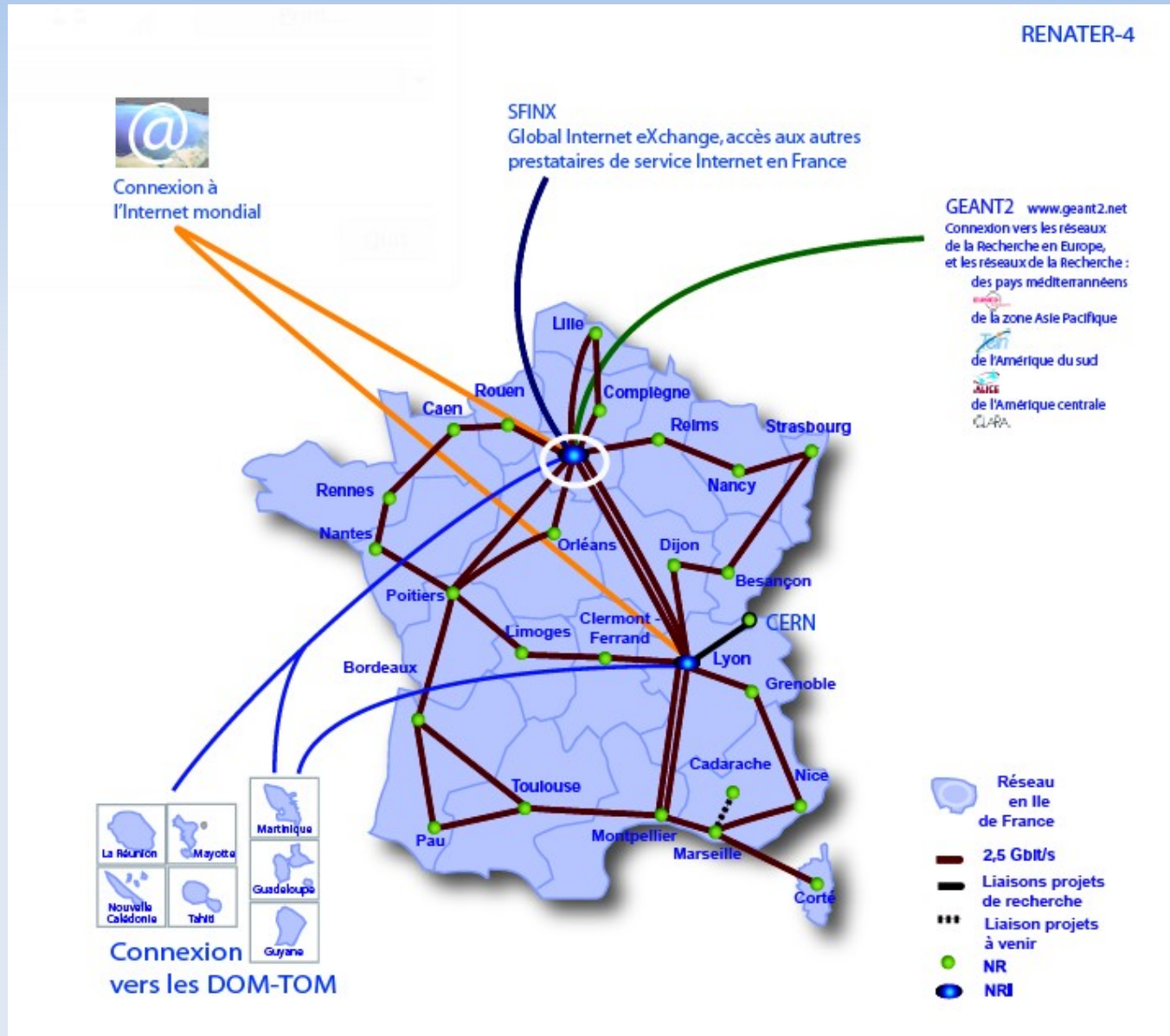
Source : Wikipedia

# Le réseau Internet

- Interconnexion de multiples réseaux hétérogènes et distants...



# Exemple de Renater

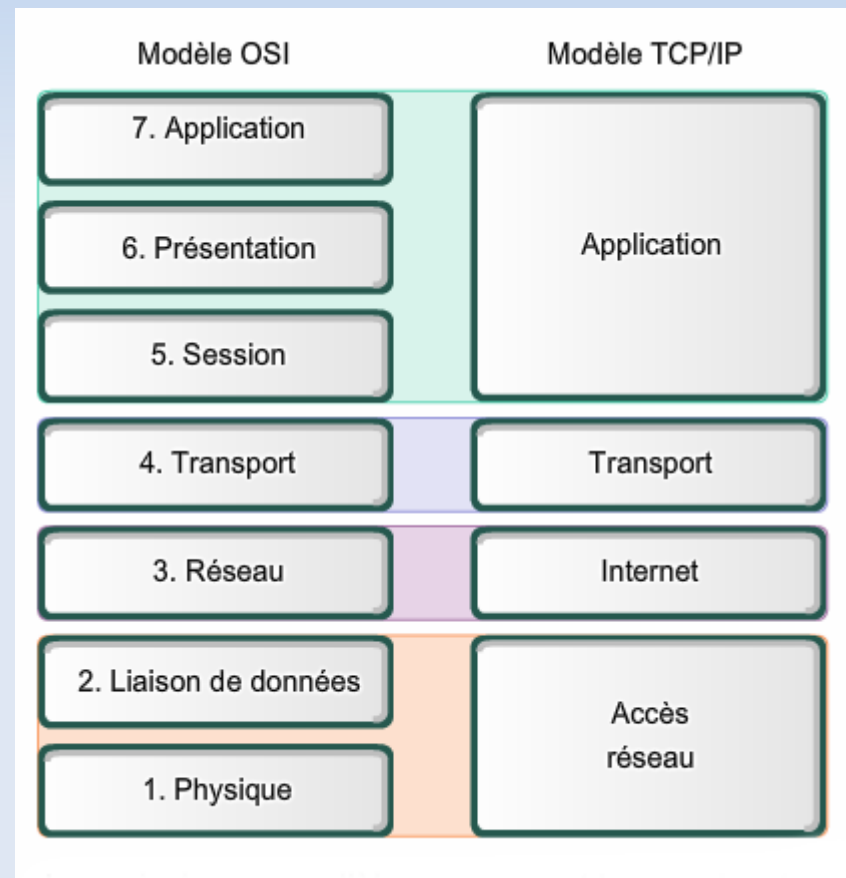


# Notion de protocole

- Protocole : Spécification de plusieurs règles pour communiquer sur une même couche d'abstraction entre deux machines
- Exemple du protocole HTTP
- Plusieurs niveaux d'interactions
  - **le niveau de l'application** : le client clique sur un lien, le serveur renvoie une page web
  - **le niveau des messages** : le client envoie un message contenant une URI, le serveur renvoie un message contenant un fichier HTML
  - **le niveau des paquets** : le message du client est découpé en paquets, les différents routeurs du réseau les acheminent vers le serveur (idem pour le retour)
  - **le niveau de la transmission des bits** : pour envoyer les paquets, chaque bit (0 ou 1) est transmis comme un signal électrique sur une ligne.
- Chaque niveau utilise les fonctions du niveau inférieur

# Modèle OSI & TCP/IP

- Modèle en couches





# Les couches OSI

## (1) Couche physique (physical layer)

- transmission effective des signaux entre les interlocuteurs
- service typiquement limité à l'émission et la réception d'un bit ou d'un train de bit continu

## (2) Couche liaison de données (datalink layer)

- communications entre 2 machines adjacentes, i.e. directement reliés entre elle par un support physique

## (3) Couche réseaux (network layer)

- communications de bout en bout, généralement entre machines : adressage logique et routage des paquets

## (4) Couche transport (transport layer)

- communications de bout en bout entre processus

# Les couches OSI

## (5) Couche session (session layer)

- synchronisation des échanges et transaction, permet l'ouverture et la fermeture de session

## (6) Couche présentation

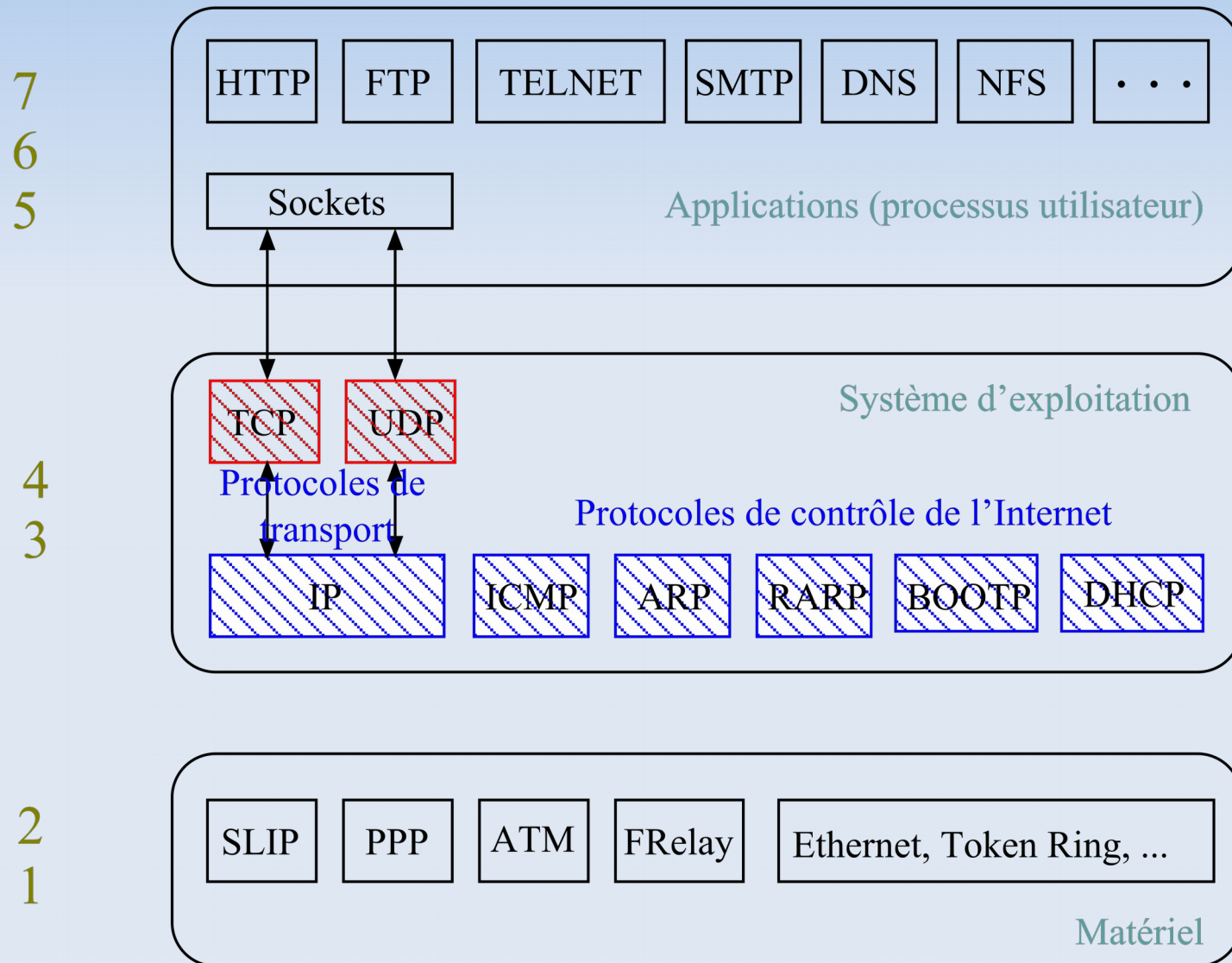
- codage des données applicatives, et plus précisément conversion entre données manipulées au niveau applicatif et chaînes d'octets effectivement transmises

## (7) Couche application

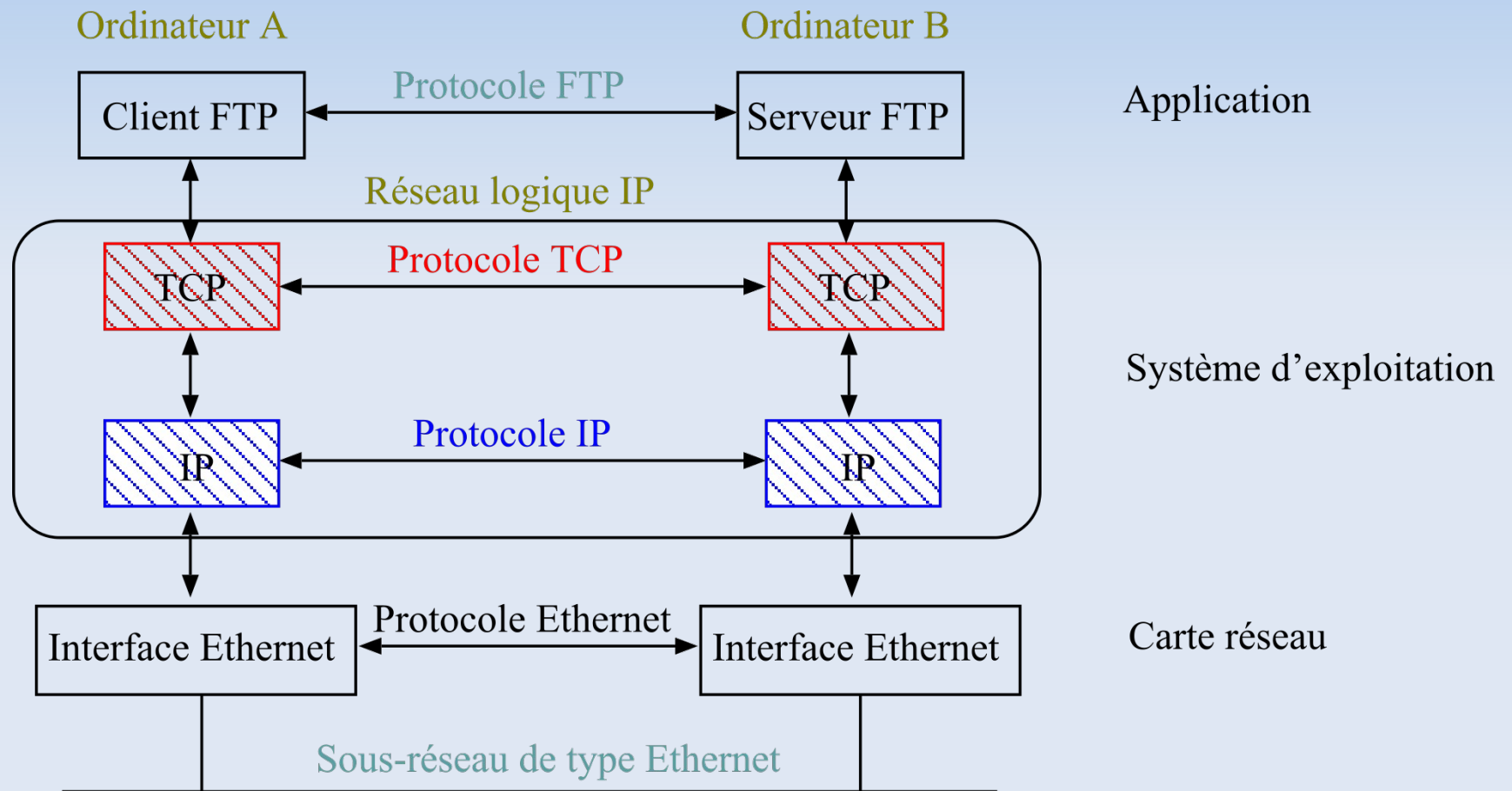
- point d'accès aux services réseaux
- elle n'a pas de service propre spécifiable et entrant dans la portée de la norme

# Illustration

OSI

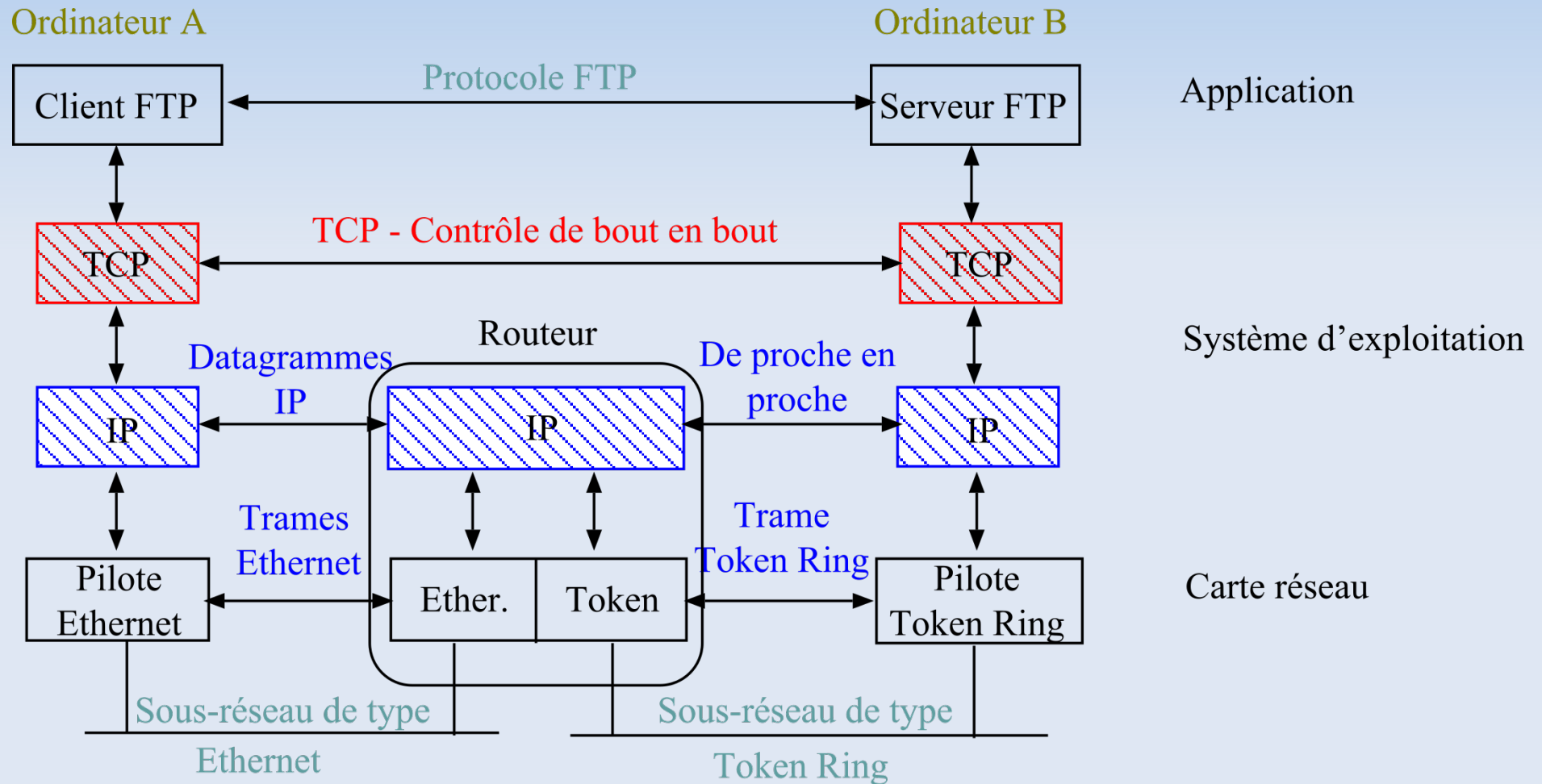


# Piles de protocoles



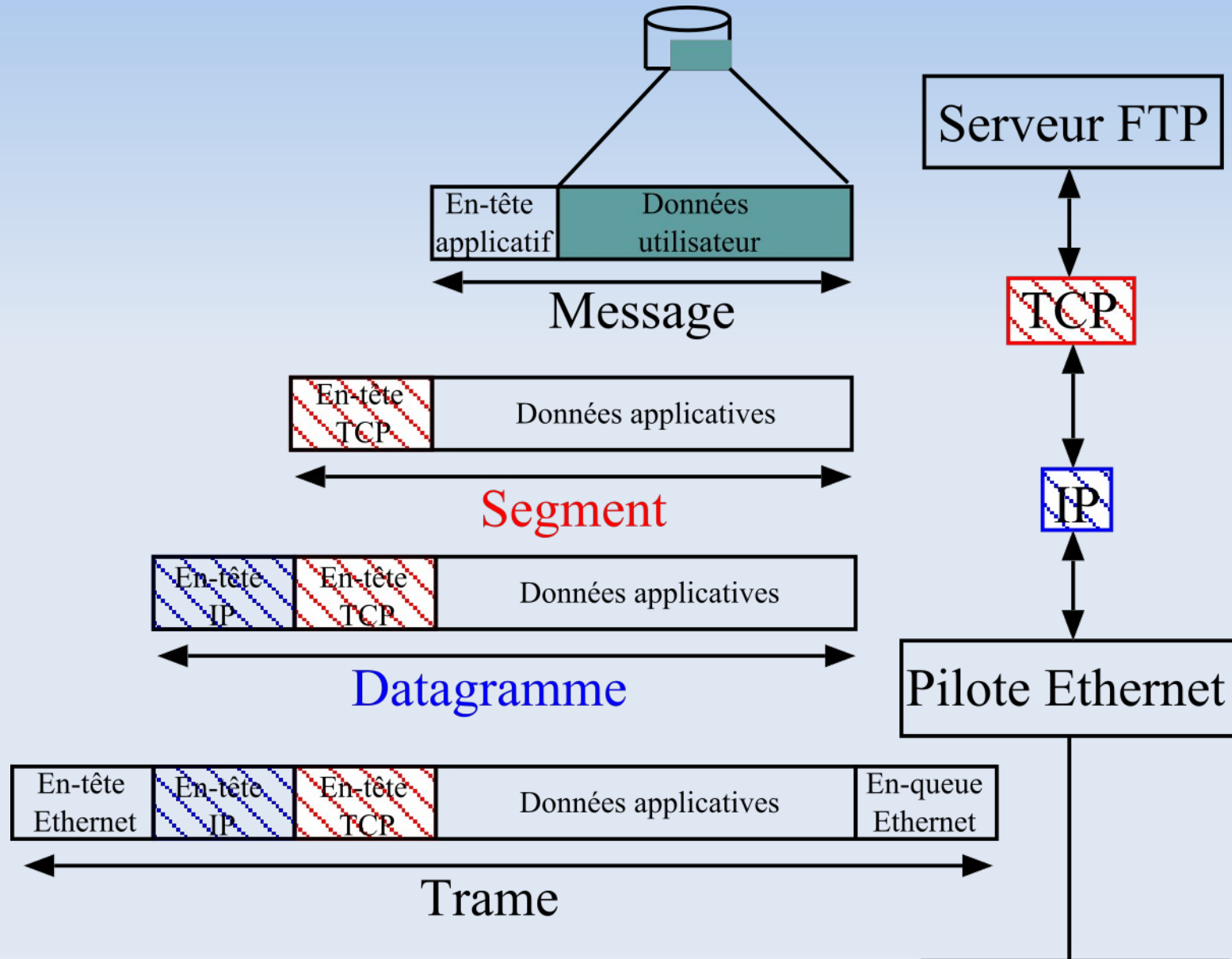
Deux machines dans un même réseau

# Piles de protocoles



Prise en compte de l'hétérogénéité

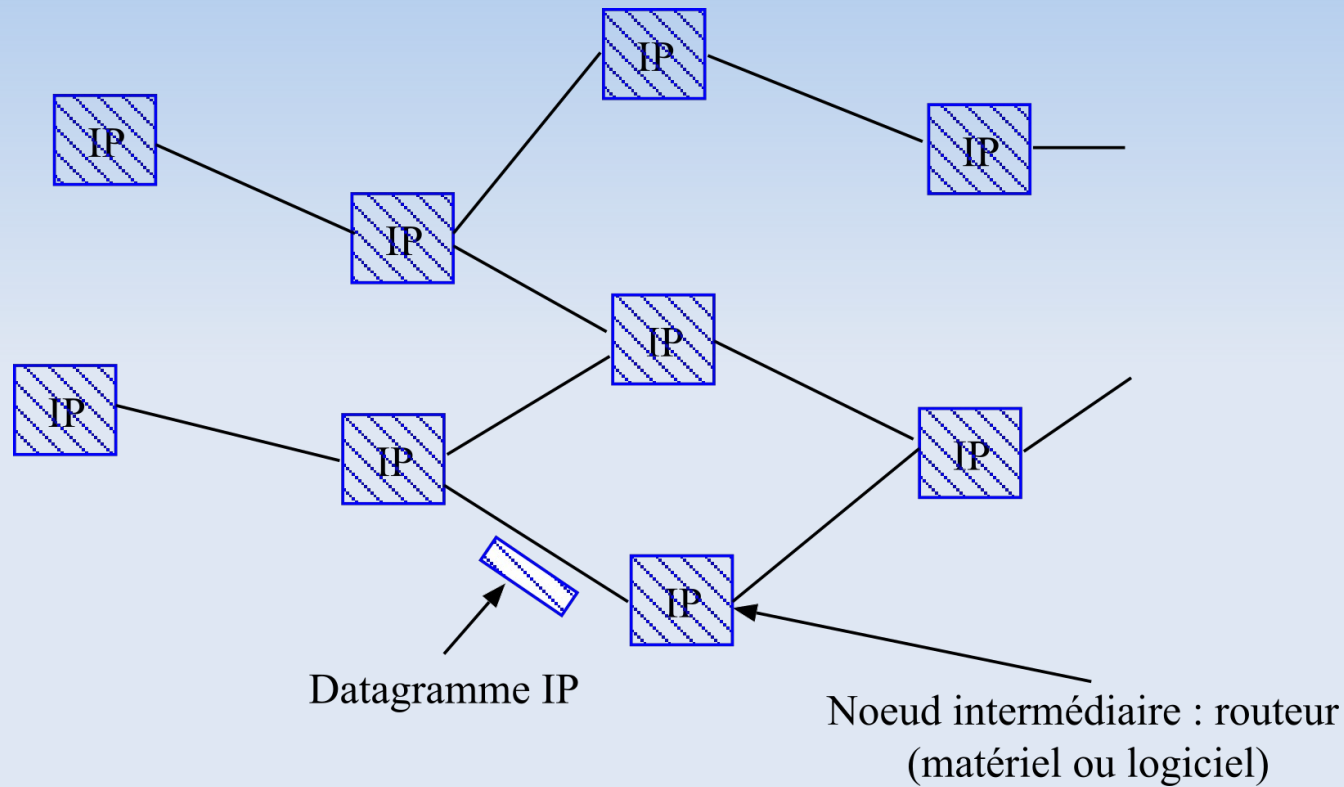
# Encapsulation





# Le protocole IP

Couche réseau : communications entre machines

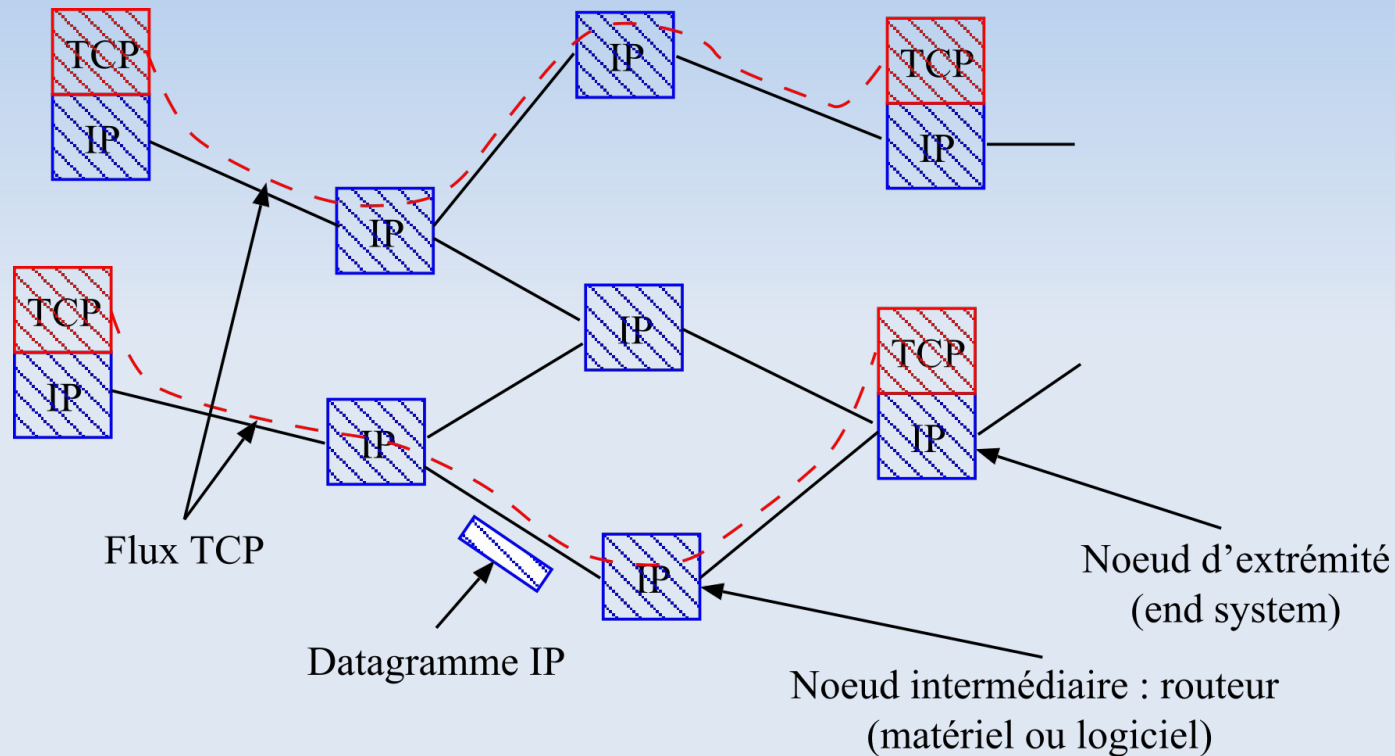


IP - protocole d'interconnexion, **best-effort**

- acheminement de datagrammes (mode **non connecté**)
- peu de fonctionnalités,
- pas de garanties simple mais robuste (défaillance d'un noeud intermédiaire)

# Le protocole TCP

Couche transport : communications entre applications



TCP - protocole de transport **de bout en bout**

- uniquement présent aux **extrémités**
- transport **fiable** de segments (mode **connecté**)
- protocole complexe (retransmission, gestion des erreurs, séquençement, . . .)

# IPv4

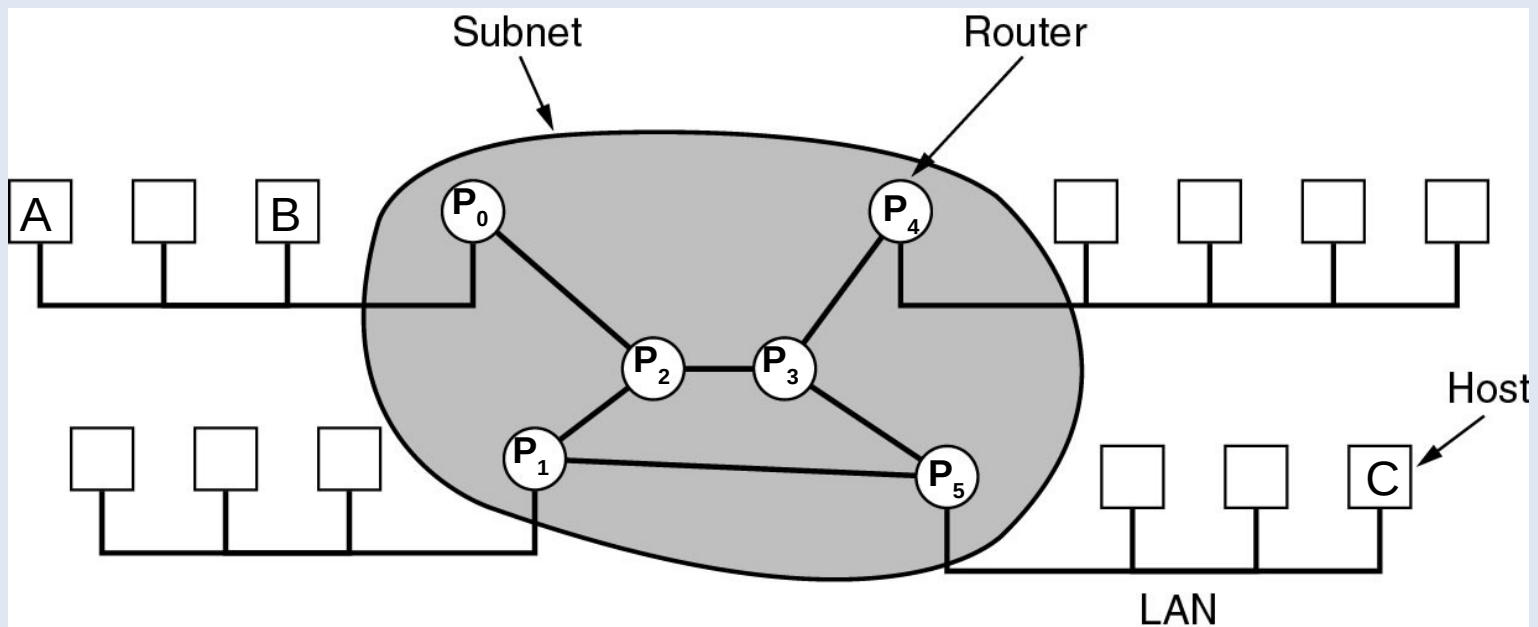
- Internet Protocol (IP)
  - communication de bout en bout entre des machines qui ne sont pas connectés directement, c'est-à-dire situées dans des réseaux locaux différents (géographie, technologie)
  - adressage logique : identifier les machines indépendamment de l'adressage physique (Ethernet, ...)
  - routage : acheminement des données entre les réseaux via des routeurs/passerelles intermédiaires
- Versions
  - IPv4, RFC 791, sept. 1981 ( $2^{32}$  adresses)
  - IPv6, le successeur de IPv4, RFC 2460, déc. 1998 ( $2^{128}$  adresses)

# IPv6

- Adresse IPv6 (8 groupe de 2 octets, noté en hexa)
  - Adresse de 128 bits
  - Exemple : 2001:0db8:0000:85a3:0000:0000:ac1f:0001
  - Forme canonique : 2001:db8::85a3:::ac1f:1
- Epuisement des adresses IPv4
  - En février 2011, IANA annonce qu'il n'y a plus de bloc d'adresse libre !
- Combien d'adresse IPv6 par mm<sup>2</sup> de surface terrestre ?
  - 667 millions de milliards d'appareils connectés sur chaque millimètre carré de la surface de la Terre !
- A compléter : <https://fr.wikipedia.org/wiki/IPv6>

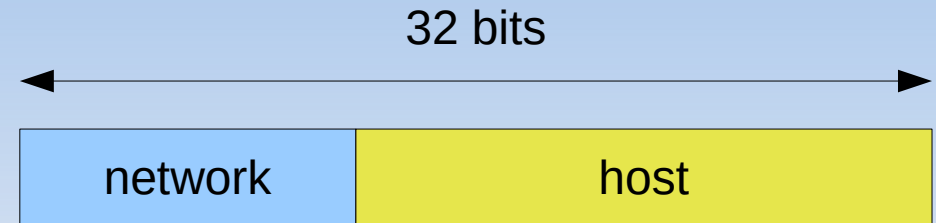
# Protocole IP

- Communication directe de A vers B
- Communication de A vers C via  $P_0$ , puis  $P_2$ , ...
  - la passerelle permet de passer d'un réseau à un autre ; elle possède donc deux interfaces réseau



# Adressage IP

- Format des adresses IP



- Les adresses spéciales

- Adresse de la boucle locale (loopback) : 127.0.0.1 ou *localhost*
- Adresse d'un réseau : tous les bits de l'adresse hôte à 0  
**192.168.10.0** / 24
- Adresse de diffusion d'un réseau : tous les bits de l'adresse hôte à 1  
**192.168.10.255**
- Adresse du routeur (par convention) : adresse de diffusion – 1

- Les masques

- 255.255.255.0 ↔ /24



# Routage

- Principe
  - Mécanisme par lequel le message d'un expéditeur est acheminé jusqu'à son destinataire, même si aucun des deux ne connaît le chemin complet que le message doit suivre...
- Deux types logiques d'ordinateur dans le WAN
  - les hôtes (hosts) ou stations, qui sont reliés à un seul réseau et qui ont par conséquent une table de routage simple
  - les routeurs/passerelles (gateway), qui relient au moins deux réseaux et possèdent une table de routage plus complexe
- Deux types d'algorithmes : statiques et dynamiques (OSPF, BGP, ...)

# Table de routage

- Un nœud (un hôte ou un routeur) a besoin des informations sur le routeur suivant (**next-hop**) vers lequel il doit envoyer un paquet pour atteindre la destination.
- La **table de routage** fournit cette information

**\$ route -n**

<i>Destination</i>	<i>Gateway</i>	<i>Genmask</i>	<i>Flags</i>	<i>Interface</i>
140.252.13.64	140.252.13.35	255.255.255.224	UG	eth0
127.0.0.1	*	0.0.0.0	UH	lo
140.252.13.32	*	255.255.255.224	U	eth0
default	140.252.13.33	0.0.0.0	UG	eth0



Next hop

# Algorithme de routage

- Algorithme exécuté sur chaque intermédiaire (R)
- Supposons que **destFinal** est l'adresse de destination de B du paquet à transmettre, **destAddr** est une adresse dans la table de routage.

si (**destAddr** == **destFinal**)

envoyer le paquet au **next-hop** de l'entrée;

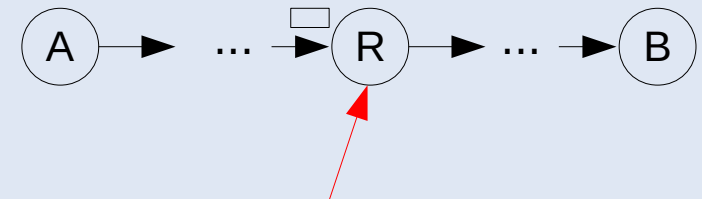
sinon

si (**destAddr** & Masque == **destFinal** & Masque)

envoyer le paquet directement à **destAddr**;

sinon

envoyer au **next-hop** de l'entrée par défaut;



**\$ route -n**

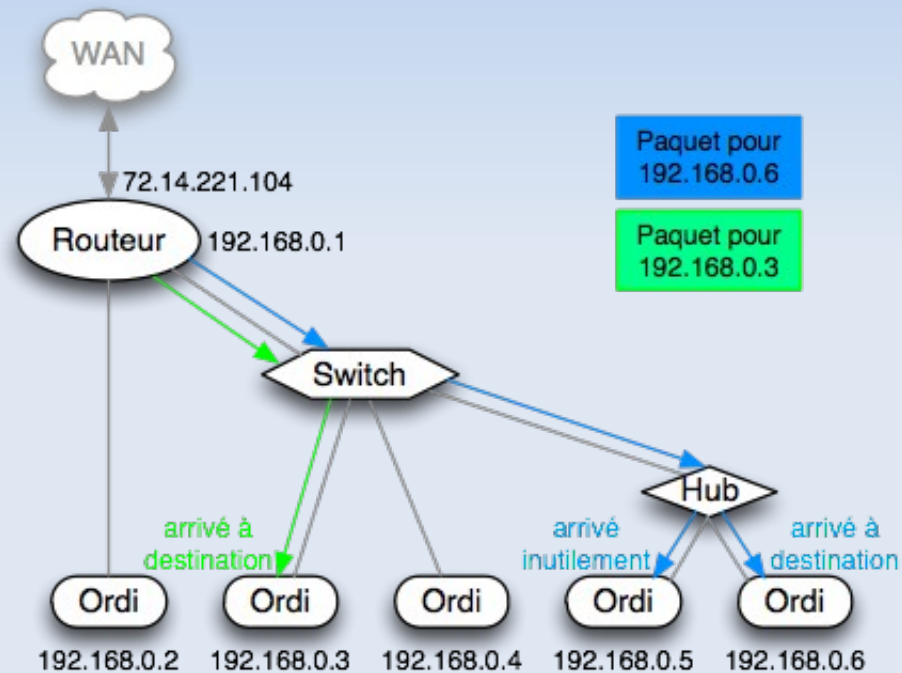
Destination	Gateway	Genmask	Flags	Interface
140.252.13.64	140.252.13.35	255.255.255.224	UG	eth0
127.0.0.1	*	0.0.0.0	UH	lo
140.252.13.32	*	255.255.255.224	U	eth0
default	140.252.13.33	0.0.0.0	UG	eth0

# Le protocole ARP

- Au sein d'un réseau local, les adresses physiques (Ethernet par exemple) sont utilisées pour communiquer
- Comment faire le lien entre les adresses IP et les adresses Ethernet ?
- Address Resolution Protocol (ARP), RFC 826
  - Protocole de résolution des adresses IP
  - Récupérer l'adresse Ethernet correspondant à une adresse IP
  - Mécanisme à base de **broadcast** dans le réseau :
    - Diffusion de la requête “qui a l'adresse @IP<sub>dest</sub> répondre à @E<sub>source</sub>”
    - La machine ayant pour adresse @IP<sub>dest</sub> répond au message fournissant ainsi son adresse Ethernet

# Comment connecter des machines entre elles?

- Hub, Switch

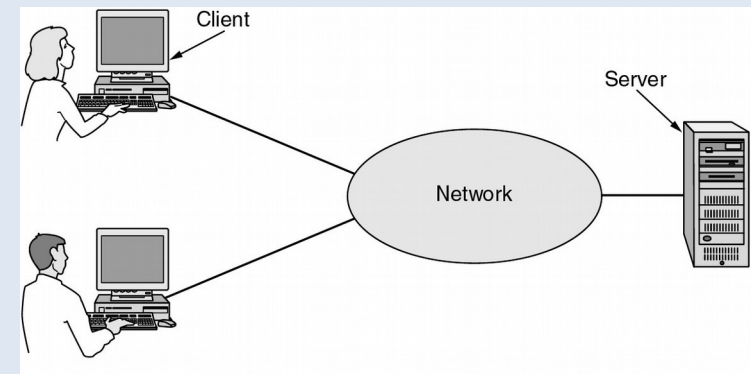


Source : <http://bencello.net/Tutos.php>

- **Passerelle/routeur** Matériel reliant deux réseaux différents et les faisant communiquer

# Le modèle Client / Serveur

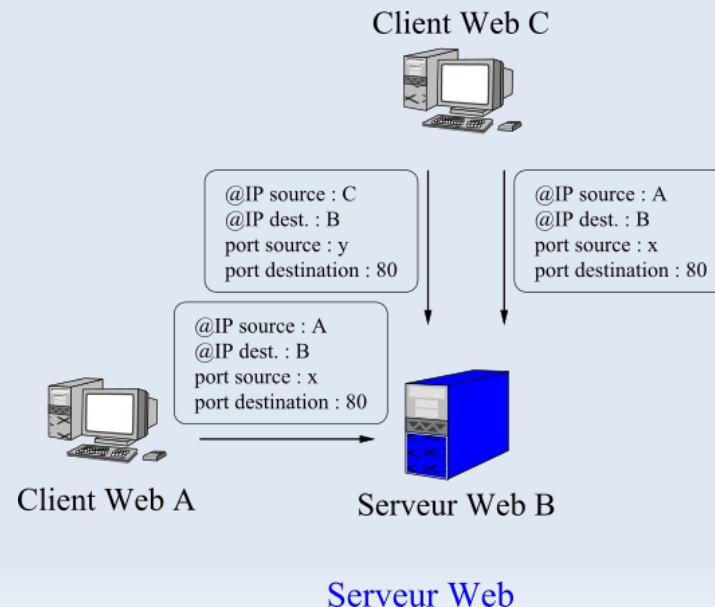
- **Le client**
  - effectue une demande de service auprès du serveur (requête)
  - initie le contact (parle en premier), ouvre la session
- **Le serveur**
  - est la partie de l'application qui offre un service
  - est à l'écoute des requêtes clientes
  - répond au service demandé par le client (réponse)
- Le client et le serveur ne sont pas identiques, ils forment un système coopératif
- Un serveur peut répondre à plusieurs clients simultanément





# Numéro de ports et sockets

- Une adresse de transport = une adresse IP + un numéro de port (16 bits) → adresse de socket
- Une connexion Client/Serveur =  $(@IP_{src}, port_{src}, @IP_{dest}, port_{dest})$
- Les ports permettent un multiplexage de connexions au niveau transport
- Les ports inférieurs à 1024 sont appelés **ports réservés**



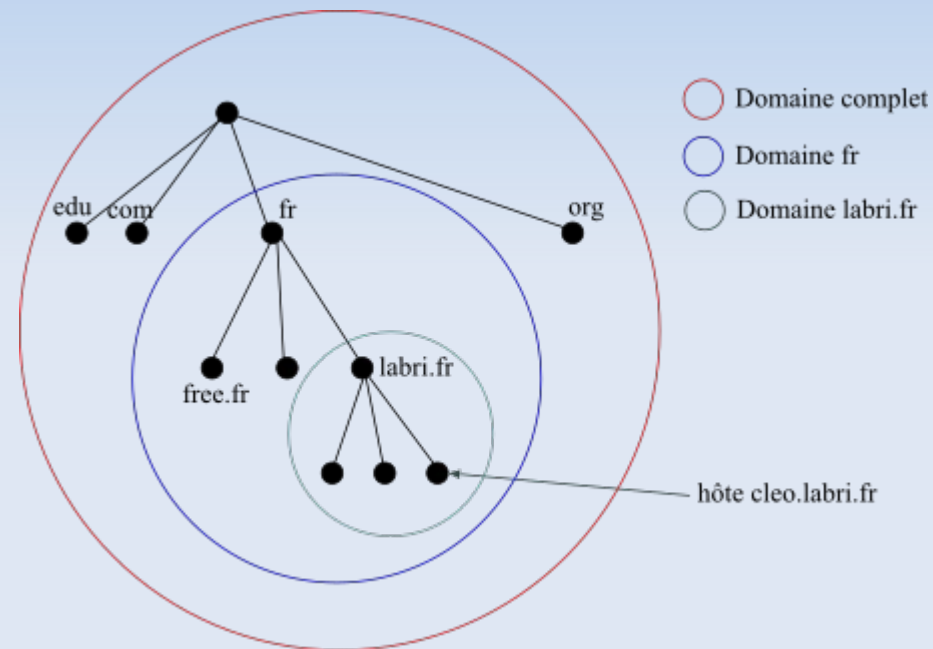
# Protocoles de Communication d'Internet et du Web

# Les services de l'Internet

- Services principaux (applications) disponibles sur l'Internet
  - Courrier électronique (mail) - protocole SMTP (Simple Mail Transfer Protocol) pour l'envoi, IMAP et POP3 pour la récupération
  - Forums de discussion (news) - protocole NNTP (Network News Transfer Protocol)
  - Transfert de fichiers - protocole FTP (File Transfer Protocol)
  - Accès à une machine distante - protocoles telnet, SSH
  - Accès au World Wide Web - protocole HTTP, formats HTML, XML, ...
  - Extensions diverses pour la sécurité (SSL/TLS, PGP, ...)
- Ces services utilisent les protocoles de transport (TCP, UDP), ainsi que le service de noms DNS (Domain Name System)

# Le protocole DNS

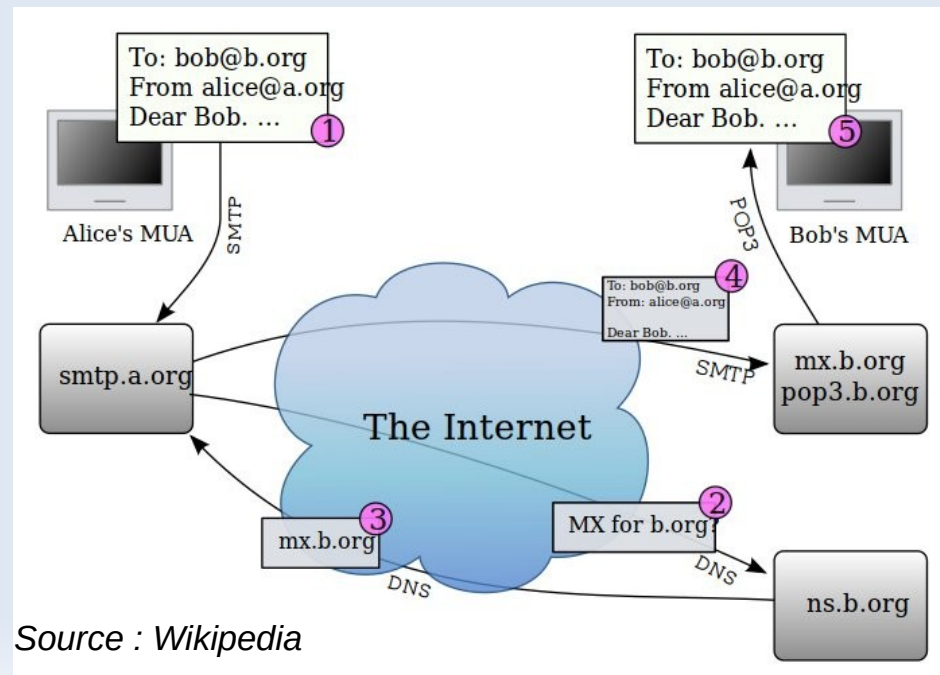
- Comment relier les adresses IP utilisées pour acheminer les paquets aux noms utilisés par les applications?
  - Utilisation de DNS
  - modèle client/serveur : un émetteur interroge un serveur de noms (serveur DNS) et attend la réponse
  - Protocole central dans Internet
  - Rarement utilisé directement par l'utilisateur
- Espace de noms hiérarchique permettant de garantir l'unicité d'un nom dans une structure arborescente



Espace de nommage

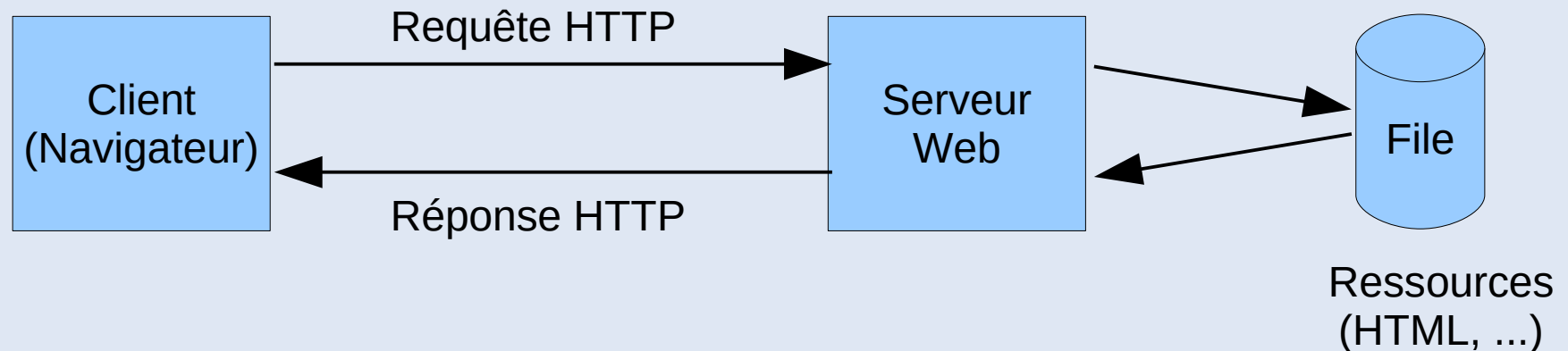
# Messagerie Electronique

- Principe d'acheminement d'un courriel
  - Envoi : lorsqu'un expéditeur envoie un courriel, son ordinateur soumet une requête au serveur sortant (SMTP), qui l'achemine vers le serveur entrant du destinataire
  - Réception : lorsqu'un destinataire relève ses courriels, ils sont téléchargés sur son ordinateur depuis le serveur entrant (POP3 ou IMAP)



# HTTP

- HTTP (HyperText Transfer Protocol)
  - Protocole *stateless* basé sur TCP/IP (port 80)
  - Le navigateur effectue une requête HTTP pour obtenir la ressource URI (Uniform Ressource Identifier)
  - Le serveur traite la requête puis retourne une réponse HTTP, typiquement une page HTML



# HTTP

- Requêtes

- GET : demander une ressource (la plus courante)
- POST : ajouter une nouvelle ressource (ex. message de forum)
- HEAD : demander uniquement l'en-tête HTTP
- TRACE : echo de la requête
- CONNECT, PUT, DELETE, ...

- Historique

- Version 0.9 : requête GET, réponse HTML
- Version 1.0 : gestion de cache, description du type MIME des ressources (content-type), ...
- Version 1.1 : connexion persistante (keep-alive), négociation de contenu (accept-\*), ...

# Un peu de HTML

- Structure classique

```
<html>
  <head>
    <title>Hello World!</title>
    ...
  </head>
  <body>
    <center>
      <h1>Hello World!</h1>
    </center>
    <h2> Subtitle </h2>
    <p> paragraph </p>
    ...
  </body>
</html>
```

- Formulaire HTML

- Passage de paramètres (POST)

```
...
<form action="/target" method=POST>
  First Name:
  <input type=text size=20 name=firstname>
  <br>
  Last Name:
  <input type=text size=20 name=lastname>
  <br>
  <input type=submit>
</form>
...
```



First Name:

Last Name:



# Exemple HTTP

- Requête

**GET /HelloWorld.html HTTP/1.1**

← commande GET

**Host:** localhost:8080

**User-Agent:** Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.15)  
Gecko/2009102815 Ubuntu/9.04 (jaunty) Firefox/3.0.15

**Accept:** text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

**Accept-Language:** en-us,en;q=0.5

**Accept-Encoding:** gzip,deflate

**Accept-Charset:** ISO-8859-1,utf-8;q=0.7,\*;q=0.7

**Keep-Alive:** 300

**Connection:** keep-alive

**If-Modified-Since:** Thu, 19 Nov 2009 14:06:01 GMT

**If-None-Match:** W/"153-1258639561000"

**Cache-Control:** max-age=0

header

# Exemple HTTP

- Réponse

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Accept-Ranges: bytes

ETag: W/"153-1258639561000"

Last-Modified: Thu, 19 Nov 2009 14:06:01 GMT

Content-Type: text/html ← *type MIME de la ressource*

Content-Length: 153

Date: Tue, 24 Nov 2009 15:48:32 GMT

Connection: close

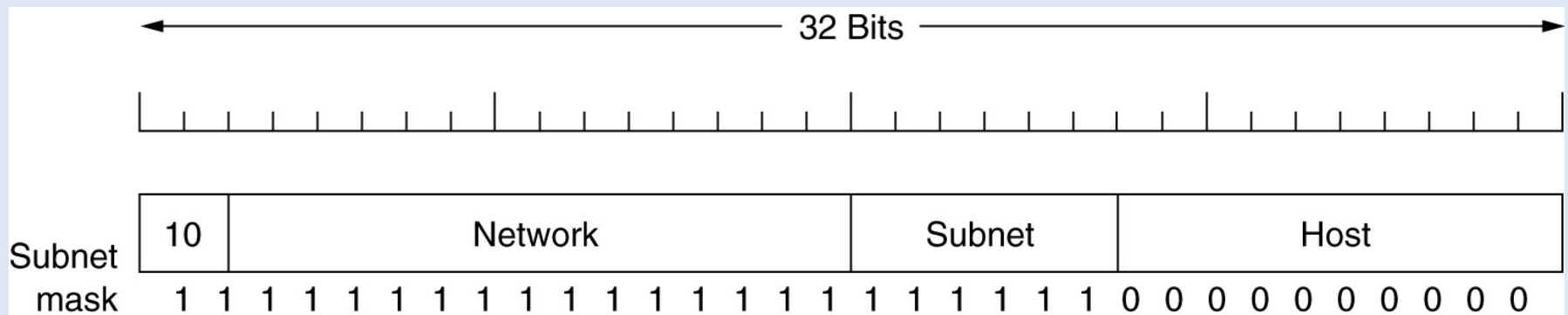
```
<html>
  <head>
    <title>Hello World!</title>
  </head>
  <body>
    <center>
      <h1>Hello World!</h1>
    </center>
  </body>
</html>
```

*corps de la réponse*

# Annexes Diverses

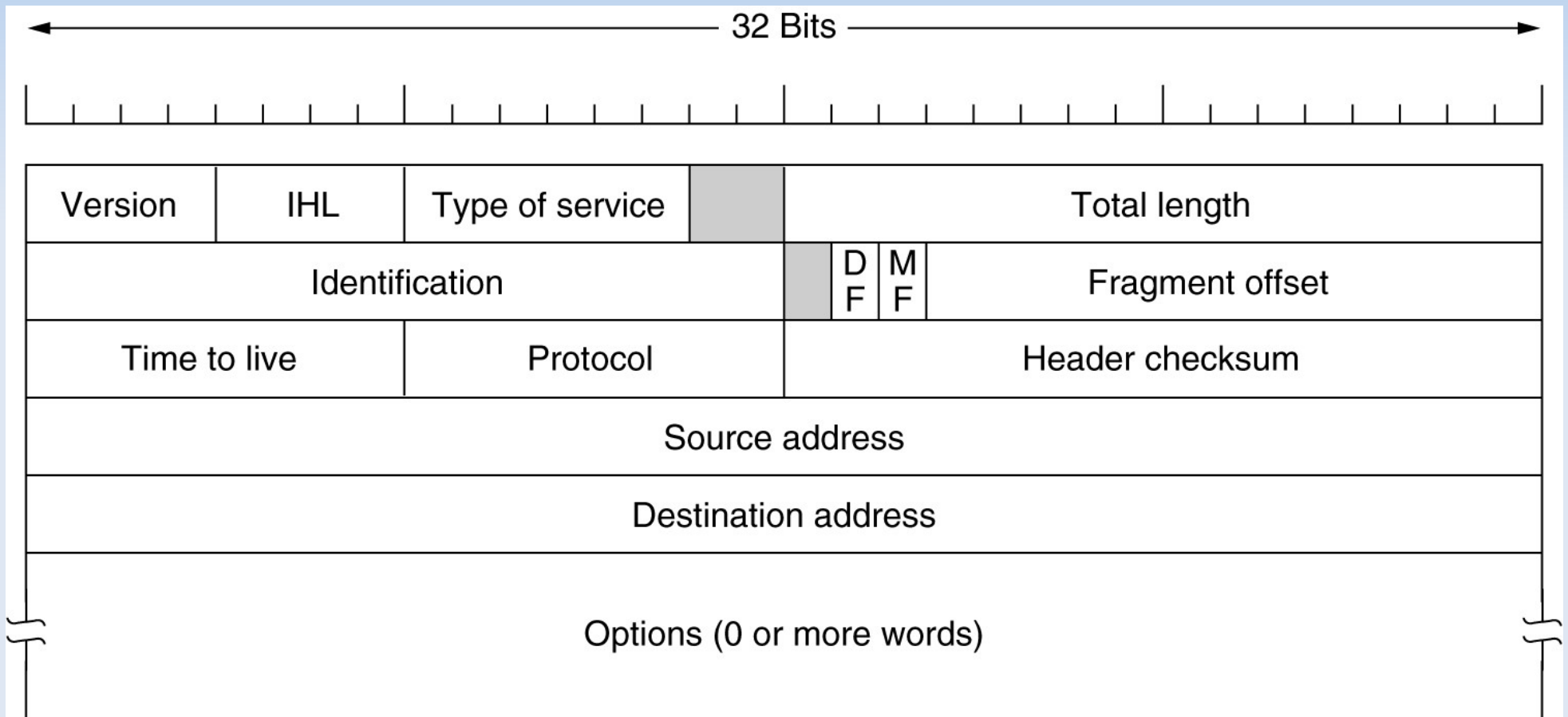
# Sous-réseaux

- Délimitation de plusieurs sous-réseaux dans un réseau
  - Adresse IP découpée en trois parties (network, subnet, host)
  - On utilise une partie des bits de l'hôte pour identifier le sous-réseau (subnet).



- Masque de sous-réseau
  - Le masque du sous-réseau s'obtient en mettant à 1 tous les bits du réseau et du sous-réseau, puis le reste à 0.
  - (adresse IP) AND (masque) = (adresse sous-réseau)

# En-tête du paquet IP (v4)



# En-tête du paquet IP (v4)

- Version : v4
- IHL (Internet Header Length) : longueur de l'en-tête en mot de 32 bits
- Type of Service : qualité de service (minimal cost: 0x02, reliability: 0x04, throughput: 0x08, low delay: 0x10)
- Identification : identifiant d'un ensemble de fragments pour leur rassemblement
- Flags : DF (Don't Fragment) / MF (More Fragment)
- Fragment Offset : position du fragment dans le message
- Time To Live (TTL) : temps de vie maximal en sec.
- Protocol : protocole de la couche supérieur encapsulé dans le paquet (ICMP, UDP, TCP, etc.)
- Header Checksum : contrôle d'erreurs de l'en-tête
- Adresses IP source et destination

# Commandes de base (1/2)

- Configuration du réseau 192.168.10.0/24
  - Configuration des interfaces réseaux de la machine A ?

\$ Ifconfig -a

**eth0** Link encap:**Ethernet** HWaddr **00:15:c5:3d:52:b6**  
inet addr:**192.168.10.1** Bcast:**192.168.10.255** Mask:**255.255.255.0**  
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
RX packets:66 errors:0 dropped:0 overruns:0 frame:0  
TX packets:70 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1000  
RX bytes:7571 (7.3 KB) TX bytes:9560 (9.3 KB)  
Interrupt:18

**eth1** Link encap:**Ethernet** HWaddr **00:13:02:dc:2a:fd**

...

**lo** Link encap:**Local Loopback**  
inet addr:**127.0.0.1** Mask:**255.0.0.0**

...

Adresse Ethernet

Adresse IP  
de la machine

Masque du Réseau IP

# Commandes de base (2/2)

- netstat -tnap (liste des connexions TCP/IP)

Proto	R-Q	S-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	127.0.0.1:2208	*.*	LISTEN	3266/hpiod
tcp	0	0	127.0.0.1:34818	*.*	LISTEN	3275/python
tcp	0	0	127.0.0.1:3306	*.*	LISTEN	3642/mysqld
tcp	0	0	0.0.0.0:25	*.*	LISTEN	3525/exim4
tcp	0	0	82.225.96.37:35551	147.210.8.143:993	ESTABLISHED	10503/mozilla
tcp	0	0	82.225.96.37:39243	147.210.13.65:22	ESTABLISHED	13758/ssh
tcp	0	0	82.225.96.37:35750	147.210.9.15:22	ESTABLISHED	13763/ssh
tcp6	0	0	*:80	*.*	LISTEN	3979/apache2
tcp6	0	0	*:22	*.*	LISTEN	3746/sshd
tcp6	0	0	*:25	*.*	LISTEN	3525/exim4



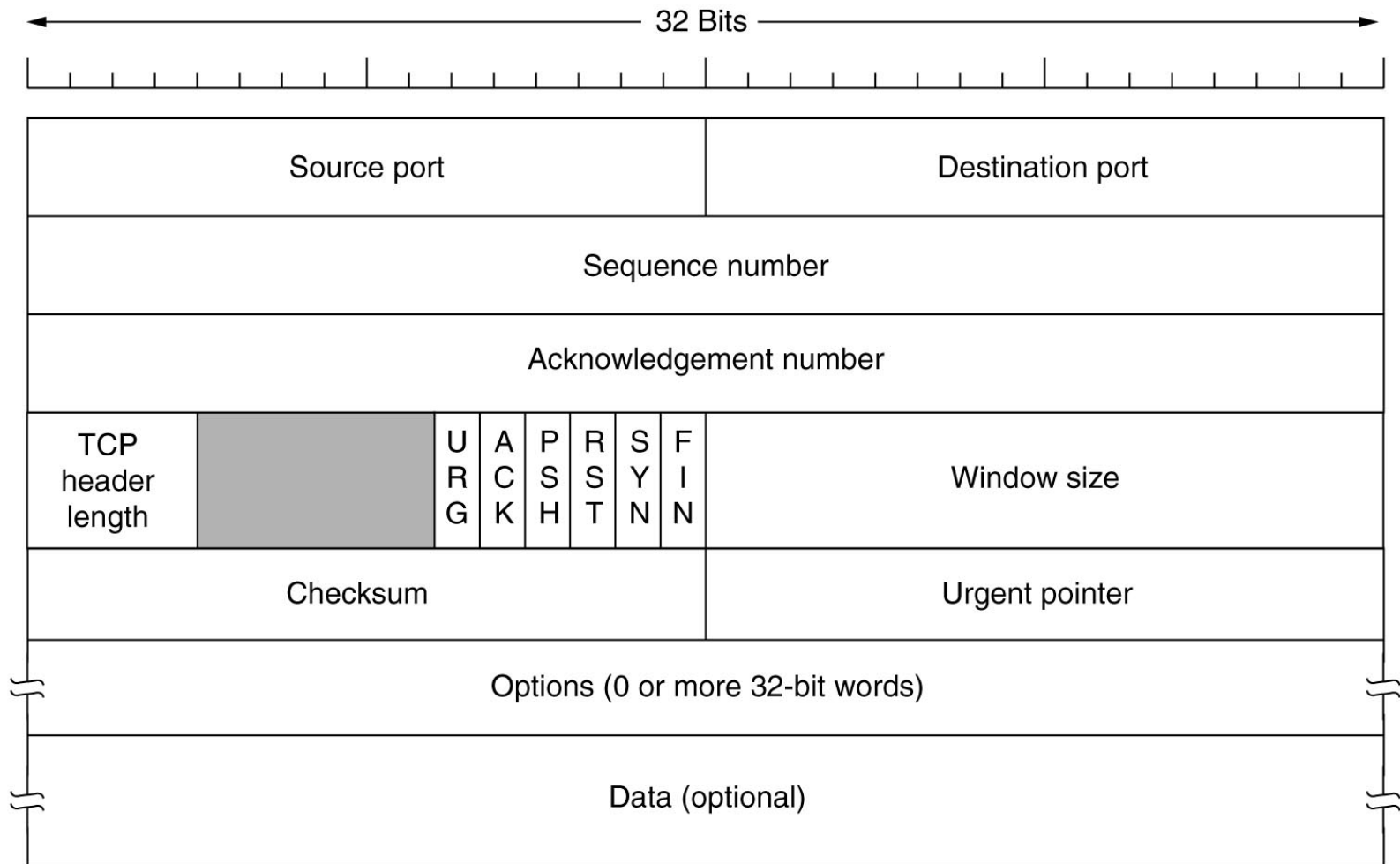
# ICMP

- Internet Control Message Protocol (ICMP), RFC 792
  - accompagne IP pour gérer les erreurs et propager des informations de routage

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

- Exemple du ping : envoi d'une requête ICMP *'echo request'* et attente de la réponse *'echo reply'*

# En-tête TCP



# En-tête TCP

- Source Port et Destination Port
- Numéro de séquence : le numéro du segment TCP
- Numéro d'accusé de réception : numéro du prochain octet attendu
- 6 flags binaires :
  - ACK : indique si le numéro d'accusé de réception est valide
  - SYN : demande d'établissement d'une connexion
  - FIN : libération de la connexion
  - RST : réinitialisation d'une connexion (reset) ; rejet d'une connexion
  - Autres : PSH, URG
- Window size : nombre d'octets souhaités pour la réception ; si 0, stoppe temporairement la transmission

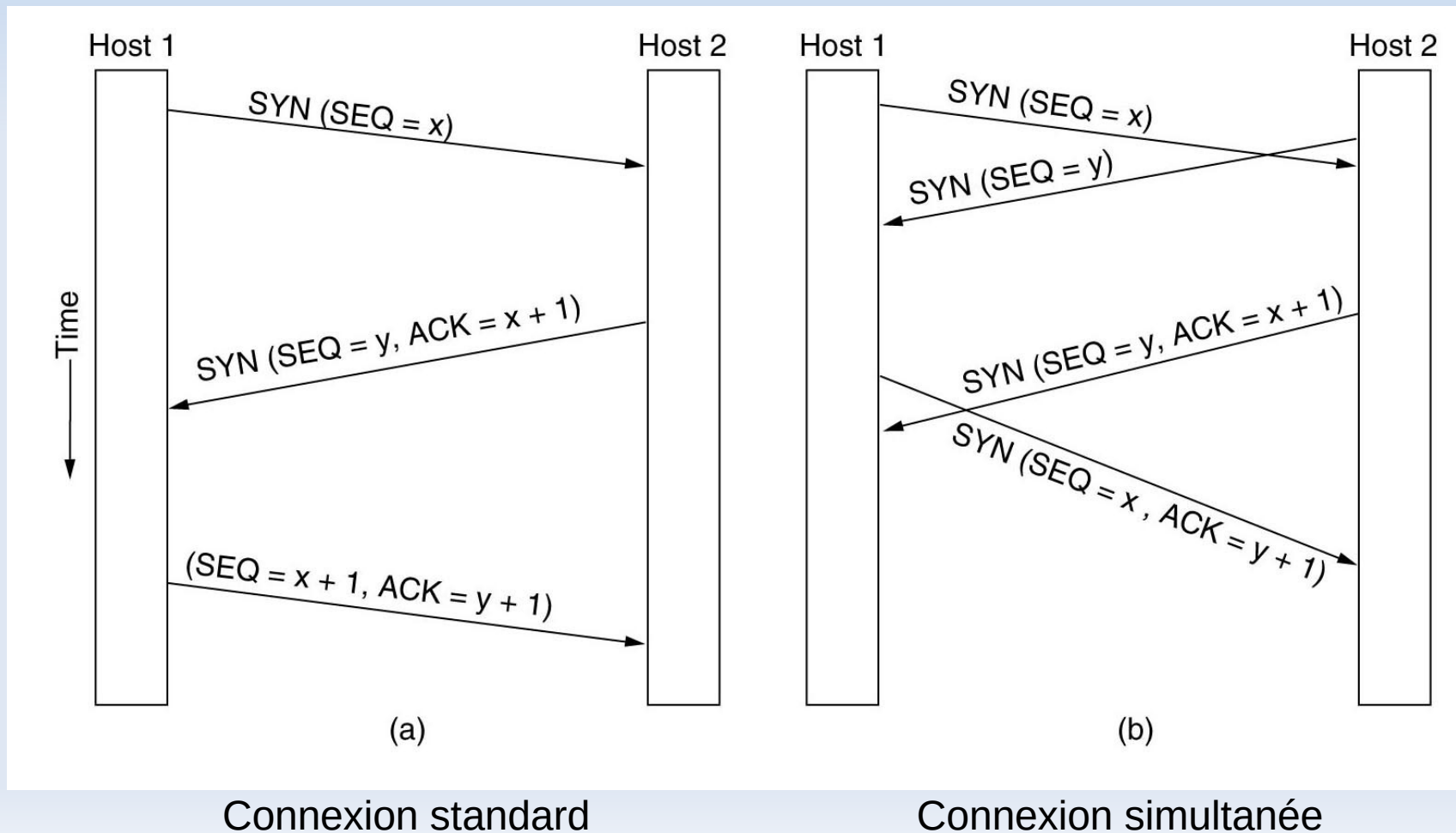
# TCP

- Exemples de service TCP standards (ports < 1024)

Port	Protocol	Use
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	E-mail
69	TFTP	Trivial File Transfer Protocol
79	Finger	Lookup info about a user
80	HTTP	World Wide Web
110	POP-3	Remote e-mail access
119	NNTP	USENET news

# Connexion TCP

- La “poignée de main” en 3 étapes
  - Synchronisation des numéros de séquence

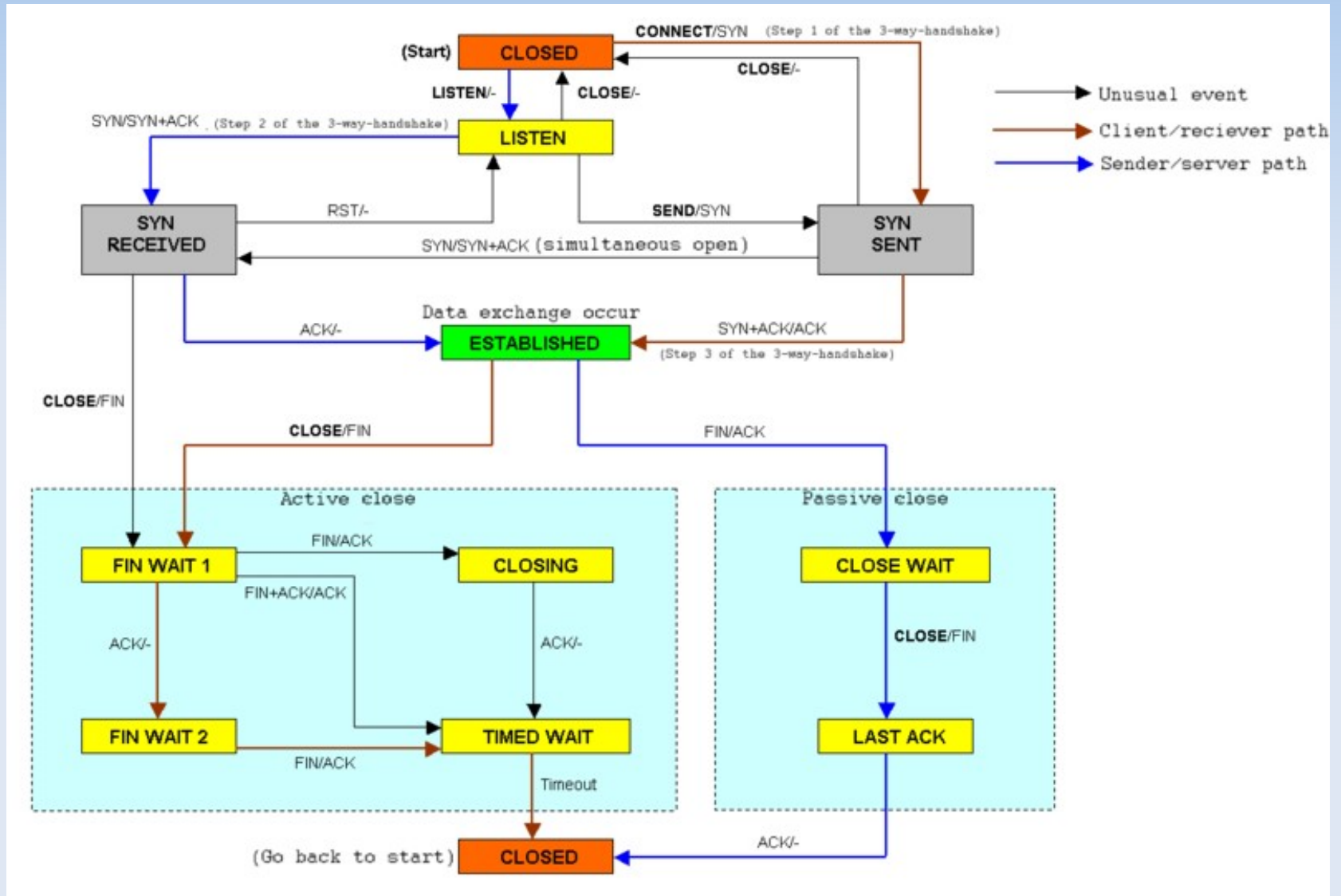


# TCP

- Les états d'une connexion TCP

State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has agreed to release
TIMED WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

# TCP

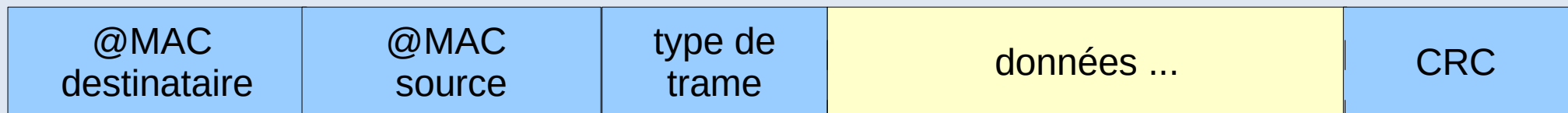


E/M

Lorsque l'évènement E se produit, envoyé le message M ou ne rien faire si M='-'.  
 →

# La trame Ethernet

- Format des trames (frames)
  - Adresse MAC du destinataire (6o) et du source (6o)
  - Type de protocole : 0800 = IP ; 0806 = ARP ; ... (2o)
  - Code CRC-32 (4o)
  - Données : au minimum 46o, jusqu'à 1500o
    - caractères de bourrage si données < 46o



Exemple du paquet IP