

# Applications réseau

## Cours 1

Corentin Travers  
travers@labri.fr

AMU - L3 info

2021-2022

# Objectifs du module

- Comprendre le concept de réseau en couches
- Savoir programmer une connexion réseau
- Savoir gérer les problématiques de la gestion d'une connexion coté serveur et coté client
- Connaître différents types d'organisation réseau d'une application : client/serveur, pair-à-pair. . .
- Acquérir les bases de l'administration d'un serveur

# Plan (tentative)

1. Introduction, modèles en couches
2. TCP/UDP, sockets java
3. TCP : entête, connexion, contrôle de congestion
4. IPv4/IPv6, Couche application (HTTP, SMTP, DNS)
5. Programmation serveur : clients multiples, E/S asynchrones
6. Programmation serveur : threads, select, E/S asynchrones
7. Systèmes distribués, architectures N-tiers, systèmes pairs à pairs
8. Systèmes pairs à pairs (suite)
9. Administration réseaux
10. Sécurité

# Quelques mots sur le projet

Le but est de **programmer une application réseau.**

- Connexion entre machines
- Communication
- Gestion des différentes connexions

**Créer un programme qui tourne sur plusieurs machines, et qui gère les communications entre celles-ci.**

## Page Ametice du module

Page ametice :

<https://ametice.univ-amu.fr/course/view.php?id=86638>

On y trouvera :

- Informations sur les dates des cours/td/tp
- Supports de cours/td/tp
- Lien vers ressources
- Forum pour échanger, poser des questions, etc.

Si vous n'avez pas accès à la page : m'écrire ([travers@labri.fr](mailto:travers@labri.fr)) en précisant votre *nom*, *prénom*, *numéro d'étudiant* et *email AMU*.

# Intervenants

- :
- Florian Beguet TD/TP (Luminy)
  - Basile Couetoux TD/TP (Luminy)
  - Shantanu Das TD/TP (Luminy)+ Cours/TD/TP (Aix)
  - Corentin Travers Cours/TD (Luminy) [travers@labri.fr](mailto:travers@labri.fr)

# Modalités du contrôle de connaissances

- Session 1 :  $NF = 0,1 * Pa + 0,15 * Tp + 0,35 * Pr + 0,4 * EF$
- Session 2 :  $NF = EF$

NF note finale

Pa partiel

Tp Compte rendus de Tps

Pr Projet (avec soutenance)

EF épreuve finale

# Distanciel/Présentiel

- 3 premières semaines de janvier :
  - Cours/TD en télé-enseignement
  - TP en présentiel
- Ensuite :
  - Cours/TD/TP en présentiel (sauf contre-ordre)



# Les réseaux informatiques

- Omniprésents
- indispensables à connaître pour un informaticien
- Le plus connu : Internet, qui est un réseaux de réseaux
- Beaucoup d'autres réseaux :
  - réseaux bancaires, réseaux de capteurs, réseaux ad-hoc etc.
  - réseaux non informatiques : réseaux ferroviaires, routiers, aériens, etc. , réseaux sociaux
  - réseaux biologiques : réseaux de neurones, chaînes alimentaires, d'interactions intra/inter-espèces, etc.

# Les réseaux informatiques

En tant qu'informaticien :

- Utilisation :
  - Programmation
  - Gestion de systèmes
  - Bases de données
  - ...
- Administration :
  - Conception de réseaux
  - Administration de réseaux
  - Sécurité (défense ou tests d'intrusion)

# Un peu d'histoire

Avant l'informatique :

- 1838 : télégraphe
- 1844 : invention du code Morse
- 1858 : premier câble transatlantique
- 1876 : téléphone

## Un peu d'histoire

Premiers réseaux informatiques :

- 1955 : SABRE (Semi Automated Business Related Environment), premier réseau commercial par IBM Utilisé pour la réservation de billets d'avion
- 1958 : premier MODEM pour communiquer sur une ligne téléphonique
- 1969 : lancement d'ARPANET, réseau militaire américain, ancêtre d'Internet, avec 4 machines
- 1972 : première application significative de messagerie électronique
- 1973 : TCP/IP
- 1979 : Ethernet
- 1984 : DNS
- 1986 : plus de 5000 machines sur Internet
- 1990 : plus de 100 000 machines connectées

# 2021 *This Is What Happens In An Internet Minute*

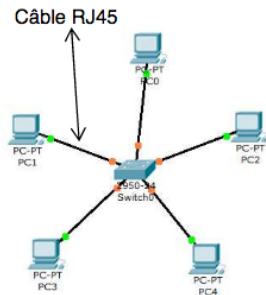


# Internet en 2021

- Environ 5,1 milliards d'utilisateurs (65% de la population mondiale)
- Environ 5,6 milliards de recherches google *par jour*
- Environ 1,7 milliards de site web (l'la moitié en activité)
- Trafic annuel de l'ordre du zetta octect ( $10^{21}$  octets)
- Environ 40 % du trafic mondial généré par des bots

# Topologie (structure) d'un réseau

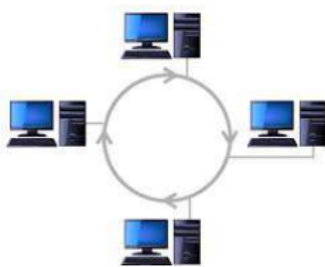
- Un réseau connecte des machines entre elles
- Différentes topologies possibles
  - Structure en étoile :



Facile à mettre en place, pas tolérant aux pannes

# Topologie (structure) d'un réseau

- Un réseau connecte des machines entre elles
- Différentes topologies possibles
  - Structure en étoile
  - Structure en anneau :

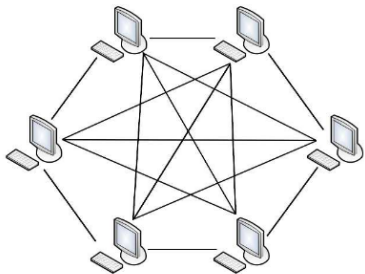


Historique (peu utilisé actuellement), une panne bloque tout le réseau



# Topologie (structure) d'un réseau

- Un réseau connecte des machines entre elles
- Différentes topologies possibles
  - Structure en étoile
  - Structure en anneau
  - Structure en maillage :



Tolérant à un certain nombre de pannes

# Topologie (structure) d'un réseau

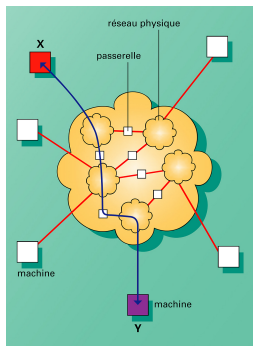
- Un réseau connecte des machines entre elles
- Différentes topologies possibles
  - Structure en étoile
  - Structure en anneau
  - Structure en maillage

Il en existe d'autres (exemple : structure en arbre)

## Internet

## INTERconnected NETworks

- Internet est un réseau de réseaux
- De (très, très. . . ) nombreux réseaux connectés entre eux

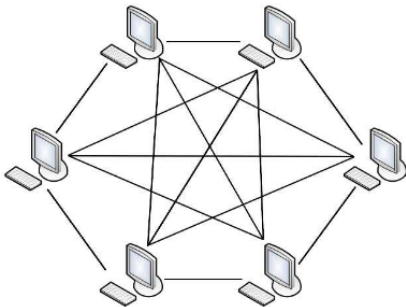


- Pour transmettre l'information entre les différents réseaux :  
roulage

# Internet

## INTERconnected NETworks

- Internet a une structure en maillage (entre les différents réseaux qui le composent) :



- Choix historique pour résister à des attaques nucléaires  
Ancêtre d'Internet : ARPANET, réseau militaire
- Extrêmement robuste au niveau global

# Protocoles

- Communication  $\Rightarrow$  besoin de **normes**
- Pour communiquer, les humains utilisent des langues (français, anglais, etc. . .)
- Les machines utilisent des **protocoles**

Protocoles pour :

- Utiliser un support physique (réseau local)
- Transporter l'information (routage)
- Utiliser l'information (application)

# Performances

Deux principaux critères de performance :

- Le débit
  - Mesure la quantité d'information qui peut transiter en un temps donné
  - bits/s : nombre de bits transmis en une seconde
  - Exemples :  
Ethernet : 10Mbits/s, 100 Mbits/s ou 1Gbit/s  
Fibre : 1Gbit/s, 10 Gbit/s ou plus  
Modem téléphonique : jusqu'à 56 kbits/s...
- La latence
  - Mesure le temps entre l'émission et la réception
  - L'aller-retour est connu sous le nom de ping

# Protocoles

## Exemples de protocoles :

- Réseau local :
  - Ethernet, Token Ring, 802.11a/b/g/n/ac (WiFi), ...
- Routage :
  - IP, ATM, ...
- Applications :
  - HTTP, FTP, SMTP, POP3, Bittorrent, SIP, ...

## Le modèle en couches

Pour mettre en place un réseau, de nombreux mécanismes sont nécessaires.

Pour gérer cela, le modèle en couche :

- Permet de séparer différentes fonctionnalités
- Permet de faire des choix pratiques de conception pour une couche sans affecter les autres (idéalement)

Exemple :

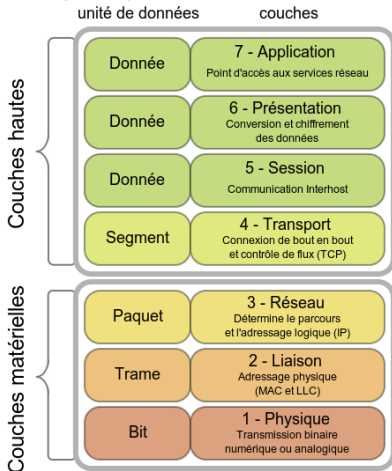
IP peut fonctionner sur Ethernet, Wifi, Tokenring, etc. . . sans modification, et même sur ces différents types de réseaux en même temps.

À chaque couche correspond un protocole



# Le modèle OSI

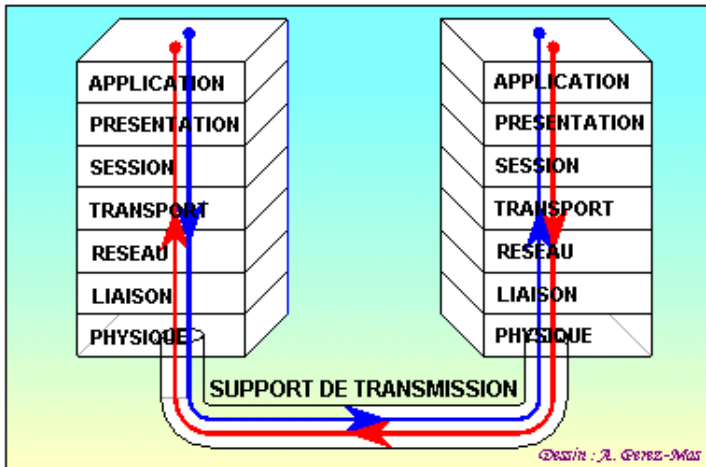
## Open Systems Interconnection



Dans ce cours, nous nous concentrerons surtout sur les couches hautes (couches basses en M1).

## Intégration des couches

- Au niveau de l'émetteur, chaque couche transmet à la couche inférieure, inversement pour le récepteur

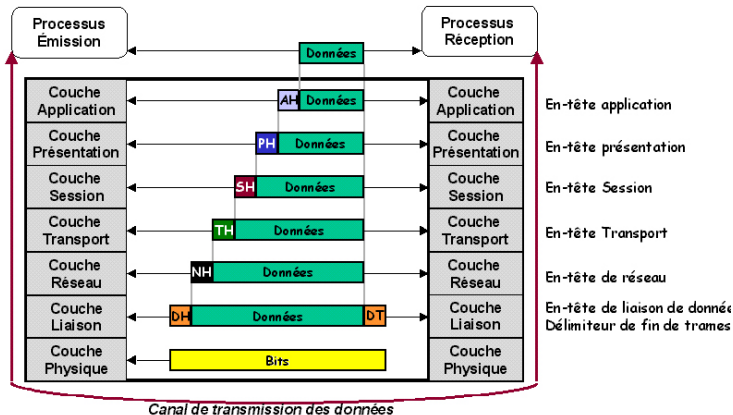


# Intégration des couches

- Chaque couche décrit dans un en-tête (et en-queue éventuel) la nature ou des détails de ce qu'elle souhaite transmettre
  - La nécessité de certaines infos est évidente. par ex, l'adresse IP
  - D'autres sont plus subtiles, par ex l'autorisation de découper un paquet
- Chaque couche s'occupe **exclusivement** de :
  - rendre les services dûs à la couche supérieure
  - utiliser les services fournis par la couche inférieure

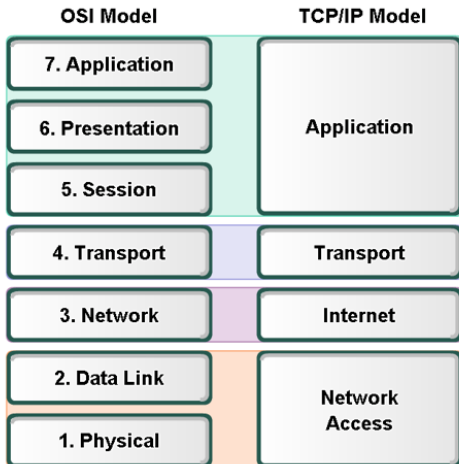
# Intégration des couches

- Chaque couche ajoute des données nécessaires à son fonctionnement au niveau de l'émetteur, les enlève au niveau du récepteur



- Principe d'**encapsulation/decapsulation**

# Le modèle TCP/IP



Le modèle TCP/IP, utilisé pour Internet, est plus ancien, basé sur des choix pratiques, et fusionne certaines couches du modèle OSI.

# Les couches physique et liaison

## Couche Physique :

- Responsable de la transmission binaire
- Concerne surtout les électroniciens, voire les physiciens

## Couche Liaison :

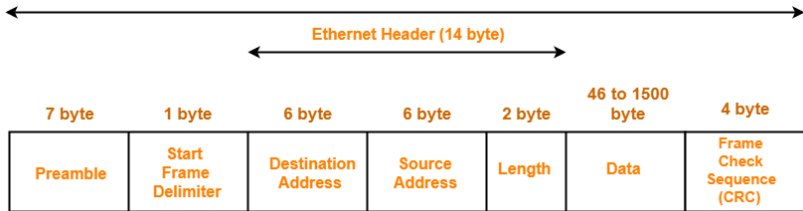
- Responsable de la transmission point-à-point (réseau local)
- Fournit des adresses physiques (locales)
- Exemples : Ethernet, Wifi, etc. . .

Dans le modèle TCP/IP, ces deux couches sont fusionnées en la couche “Accès réseau”

# Les couches physique et liaison

Exemple : Ethernet

Unité de transmission : la **trame**  
64 - 1518 byte



**IEEE 802.3 Ethernet Frame Format**

**Adresses physiques** : uniquement au niveau local

# La couche Réseau

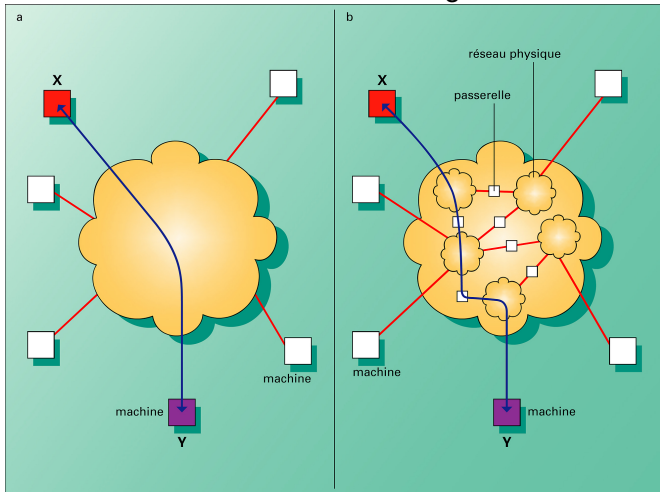
- Responsable de la transmission de bout en bout
- Assure l'interopérabilité entre différents réseaux locaux
- En particulier, détermine le parcours des données entre les réseaux et fournit des **adresses logiques (globales)**
- Exemple le plus répandu : IP

Appelée couche Internet dans le modèle TCP/IP



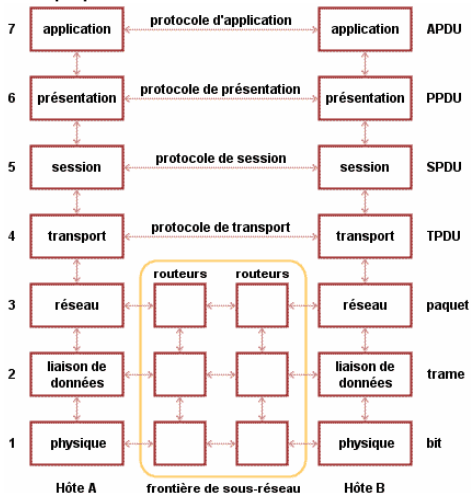
# La couche Réseau

Elle permet de faire abstraction du routage :



# La couche Réseau

Seules les couches basses (jusqu'à la couche Réseau) concernent les équipements intermédiaires

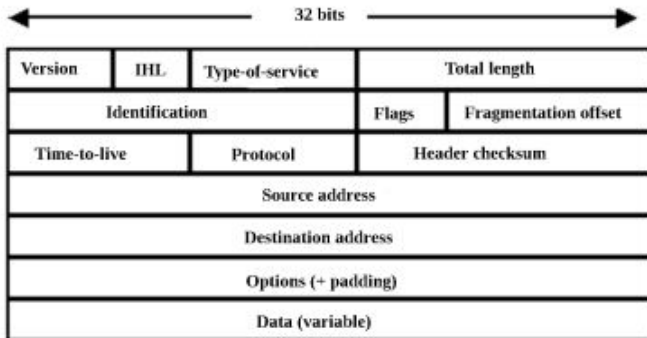


# La couche Réseau

Exemple : IP

Unité de transmission : le **paquet**

IP packet makeup



Adresses **globales** :

permettent de communiquer avec d'autres réseaux

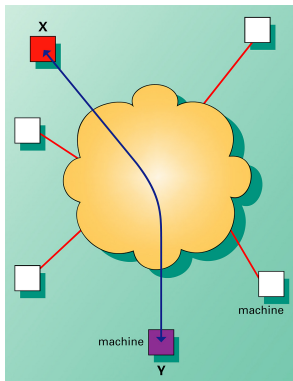
# La couche Transport

- En fonction des cas, gère la connexion, le contrôle de flux et fournit certaines garanties
- TCP :
  - Fournit une connexion
  - Garantit la fiabilité de cette connexion
  - Gère le contrôle de flux pour éviter des pertes de données
- UDP :
  - Beaucoup plus léger que TCP
  - Sans connexion
  - Pas de garantie contre les pertes
  - Pas de contrôle de flux
  - Utile pour les applications nécessitant une réception rapide des données (exemple : VoIP)

TCP et UDP ajoutent la notion de **port** : permet de contacter différents services sur une même machine

# La couche Transport

- À partir de la couche Transport, un réseau unique (d'un point de vue conceptuel)



Chaque machine a une adresse globale et peut être atteinte par toutes les autres

# La couche Session

- Gère la synchronisation des échanges
- Permet l'ouverture et la fermeture de session

N'existe pas dans le modèle TCP/IP  
(intégrée à la couche Application)

# La couche Présentation

- Chargée du codage des données applicatives
- Si besoin, chargée de l'encryption

N'existe pas dans le modèle TCP/IP  
(intégrée à la couche Application)

# La couche Application

- Contient toutes les données nécessaires à l'application elle-même
- Dépend entièrement de l'application
- Exemples :  
HTTP (Web), FTP (transfert de fichiers), SMTP (email)...



# IPv4/IPv6

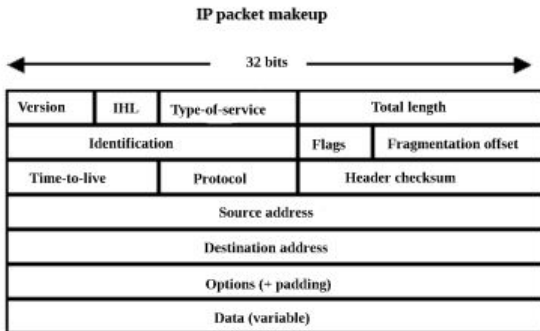
- Plusieurs versions d'IP
- Actuellement, IPv4 et IPv6 coexistent
- IPv4 :
  - Omniprésent
  - Adresses sur 32 bits : environ 4 milliards d'adresses disponibles

⇒ Pénurie d'adresses !

- IPv6
  - Pas présent partout
  - Incompatible avec IPv4
  - Introduit (principalement) pour résoudre la pénurie d'adresses :  
environ 667 millions de milliards d'adresses IPv6 disponibles par mm2 de la surface de la Terre...

# IPv4

## Paquet IPv4

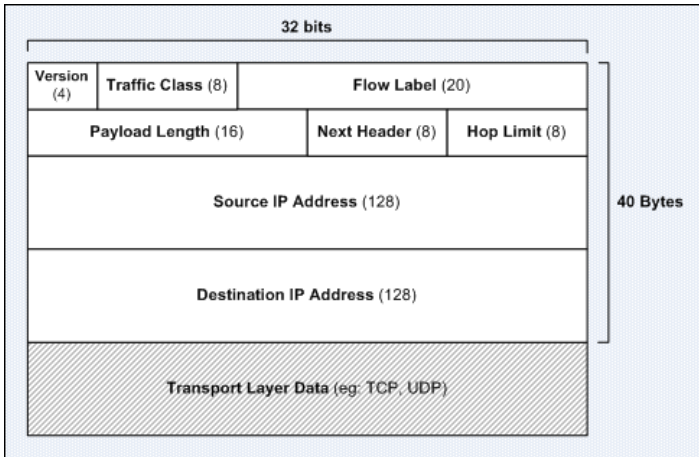


- IHL (Internet Header Length) : taille de l'entête
- Time-to-live : nombre de sauts maximum

évite qu'un paquet reste indéfiniment dans une boucle

# IPv6

## Paquet IPv6



# DNS

## Domain Name System

- Difficile de se souvenir de 139.124.244.38...
- DNS permet d'accéder à des adresses IP en utilisant des adresses mémorisables par des humains :  
`www.univ-amu.fr`
- Fonctionne pour IPv4 et IPv6
- DNS est un système hiérarchique : `www.univ-amu.fr` appartient d'abord à la zone `.fr`, puis à la zone `univ-amu.fr`.
- L'adressage de chaque zone est géré par un ou plusieurs serveurs DNS.

# DNS

## Domain Name System

Quand un client interroge un serveur DNS :

- Le serveur interroge un serveur DNS racine, qui lui renvoie l'adresse du serveur DNS qui gère le domaine
- Le serveur interroge le serveur DNS qui gère le domaine, qui renvoie l'adresse du serveur qui gère le sous-domaine
- Etc. . . jusqu'à obtenir l'adresse IP recherchée

