

# Informatique 2M

A. Ridard

## Réseaux





# Table des matières

- I. Internet, c'est quoi? . . . . . 4
- II. Couche 1 : accès au réseau . . . . . 6
- III. Couche 2 : internet . . . . . 8
  - 1. Adresse IP . . . . . 9
  - 2. Routage . . . . . 10
- IV. Couche 3 : transport . . . . . 12
  - 1. UDP . . . . . 12
  - 2. TCP . . . . . 13
- V. Couche 4 : application . . . . . 14
  - 1. DNS . . . . . 14
  - 2. HTTP(S) . . . . . 15

# I. Internet, c'est quoi ?

**Internet** vient de la contraction de « **Inter**connected » et « **net**works », ou encore **réseaux interconnectés** en français. Ce **réseau global** est donc constitué de multiples réseaux locaux.

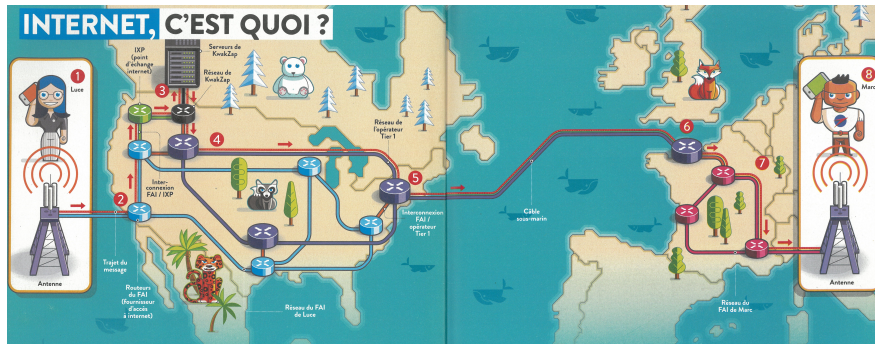
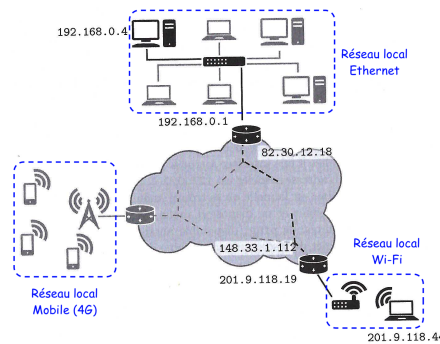


Illustration du Science & Vie Junior HS 162 (novembre 2023)

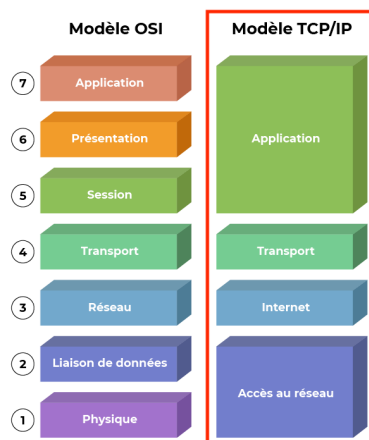
Le site [he.net/3d-map](http://he.net/3d-map) permet de visualiser tous les câbles sous-marins dans le monde.

De manière générale, un **réseau** est un ensemble d'éléments reliés entre eux. Il permet la distribution de flux (électricité, eau) ou de quantités discrètes (personnes, courriers).

Un **réseau informatique** permet la distribution d'informations, et la nature des liens diffère selon le type du réseau (on y reviendra).



Malgré ces natures différentes, tous ces réseaux interconnectés se comportent bien ensemble grâce à une « **organisation en couches** ».



Cette organisation en couches a été formalisée en 1984 dans le **modèle OSI** <sup>[1]</sup> correspondant à la norme ISO <sup>[2]</sup> 7498 : 1984. Ce modèle que l'on peut qualifier de théorique sert de guide pour les processus de certification des réseaux informatiques.

Mais bien avant la publication de cette norme, un réseau informatique global s'est développé en suivant ses propres principes, le réseau Internet. Initialement conçu en 1970 comme un projet militaire américain, le projet ARPANET a rapidement évolué pour devenir en 1974 la paire de protocoles TCP/IP :

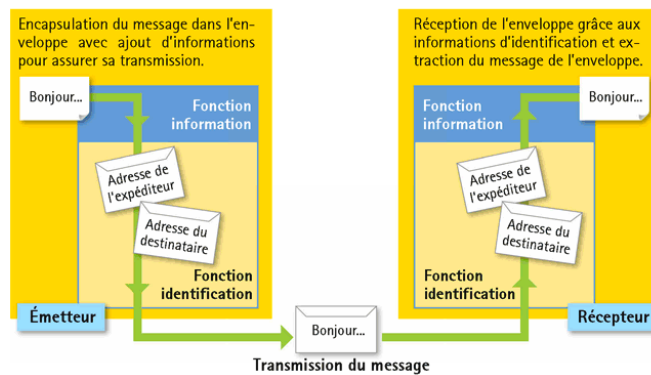
- TCP <sup>[3]</sup> permet la communication entre deux applications distantes <sup>[4]</sup>
- IP <sup>[5]</sup> assure l'acheminement des données jusqu'au destinataire

Ces protocoles sont standardisés par l'IETF <sup>[6]</sup> qui propose un modèle OSI « simplifié » constitué de quatre couches : le **modèle TCP/IP**.

Un **protocole de communication** est un ensemble de règles communes suivies par les dispositifs désirant effectuer un échange d'informations.

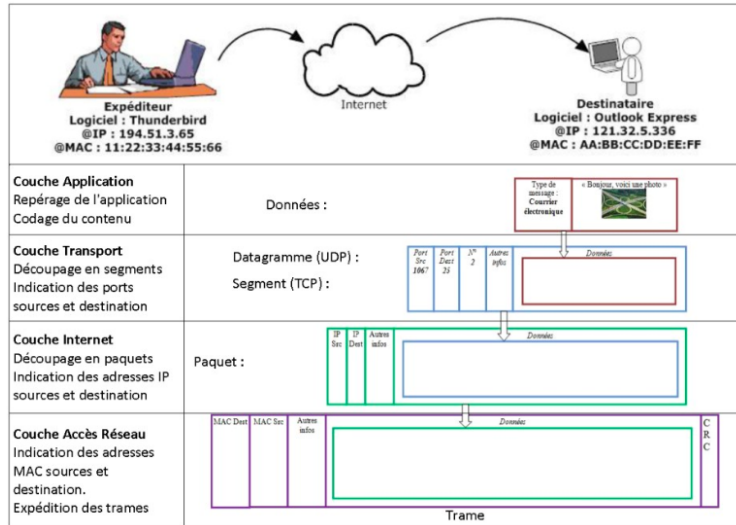
n°	Couche	Exemples de protocoles
4	Application	HTTP(S), DNS, SMTP, IMAP
3	Transport	TCP, UDP
2	Internet	IP
1	Accès au réseau	Ethernet, Wi-Fi

Lorsque des données sont envoyées à travers un réseau en utilisant un protocole, ce dernier va y ajouter des **méta-données**, c'est à dire de l'information permettant de s'assurer que la communication se passe bien.



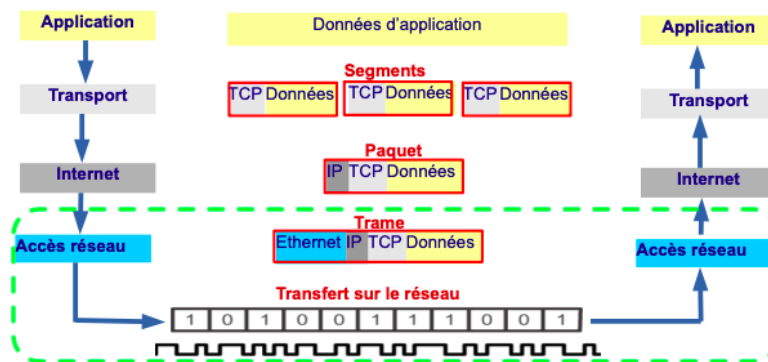
[1]. Open Systems Interconnection  
 [2]. Organisation internationale de normalisation  
 [3]. Transmission Control Protocol  
 [4]. Par exemple, entre un navigateur Web et le serveur Web qui possède la page Web demandée  
 [5]. Internet Protocol  
 [6]. Internet Engineering Task Force

Ces ajouts de méta-données, à chaque niveau, suivent le principe d'**encapsulation**.

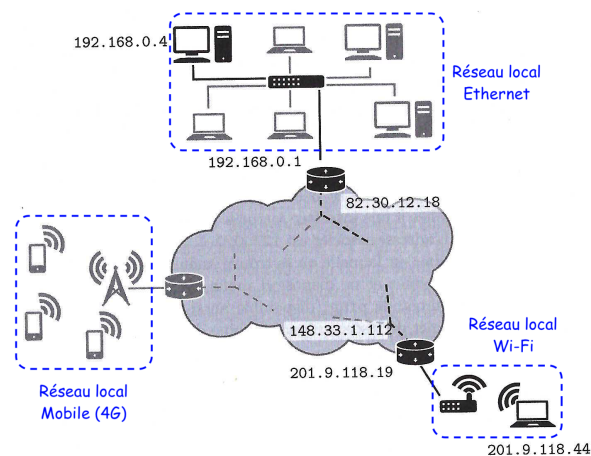


## II. Couche 1 : accès au réseau

Intéressons-nous d'abord à la **couche 1 : accès au réseau**.



Cette couche regroupe tous les **aspects physiques** du réseau. Ici, on ne considère que des **machines reliées directement** entre elles par la même technologie. On parle alors de **réseau local** ou LAN [7].



[7]. Local Area Network



Vous savez que l'information<sup>a</sup> est représentée en binaire, mais comment ces suites de 0 et de 1 sont-elles transférées d'un point du vue physique?

a. nombres, textes, images, sons, vidéos

Cela dépend du type du réseau local, de la nature des liens entre les machines :

- réseau Ethernet : câble de cuivre → signaux électriques
- réseau Mobile (4G) ou Wi-Fi : voie aérienne → ondes radios
- réseau FAI<sup>[8]</sup> : fibre optique → ondes infrarouges



Comment les ondes portent-elles les données?

**COMMENT LES ONDES PORTENT-ELLES LES DONNÉES ?**

Sur internet, les données empruntent des routes de toutes tailles, mais aussi de nature différente: câble de cuivre, fibre optique ou voie aérienne. Pourtant, les données voyagent toujours de la même façon: d'abord transformées en longues séquences de 0 et de 1, elles sont ensuite portées par des ondes. Eh oui, la lumière que l'on envoie dans les fibres optiques et le courant qui circule dans les câbles de cuivre sont aussi des ondes, c'est-à-dire des sortes de vagues d'énergie.

Comment case-t-on des 0 et des 1 dans une onde?

① On part d'une onde bien propre et régulière, appelée « la porteuse ». Pour y injecter les données (nos 0 et 1), on va moduler l'onde, c'est-à-dire triturer sa forme avec des creux et des bosses.

② Telle bosse devient un peu plus élevée, telle autre un peu plus resserrée en fonction des 0 et des 1 qu'elle porte. À l'arrivée, l'onde modulée est captée par un récepteur qui reconnaît les déformations subies par l'onde initiale. Il sait donc quelle série de 0 et de 1 elle a portée. O.F.



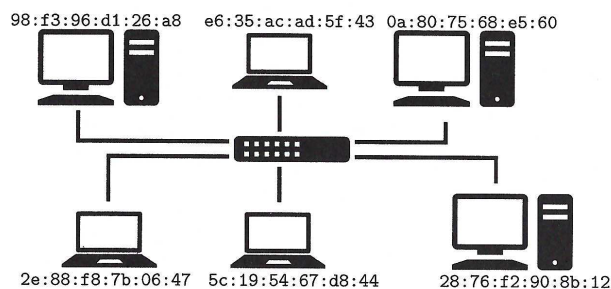
**Pour en savoir plus**

N'hésitez pas à poser des questions à votre enseignant de Physique...



Comment les trames sont-elles « distribuées » au sein d'un réseau local?

Dans un réseau local de type Ethernet, chaque machine possède une carte réseau avec une **adresse MAC** associée (6 octets représentés en hexadécimal et séparés par :).



Dans un tel réseau, les machines sont reliées par un **commutateur** qui envoie la trame qu'il reçoit uniquement sur le câble correspondant à l'adresse MAC de destination.

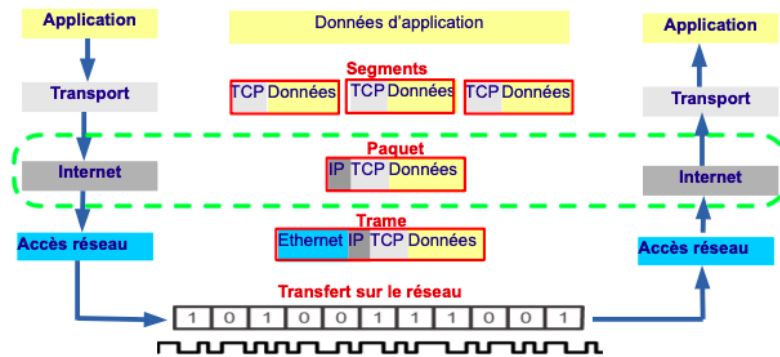
[8]. Fournisseur d'Accès à Internet

### > Filius (1/4)

- Exercice 1 : connexion pair à pair de deux machines
- Exercice 2 : interconnexion de plusieurs machines avec un switch

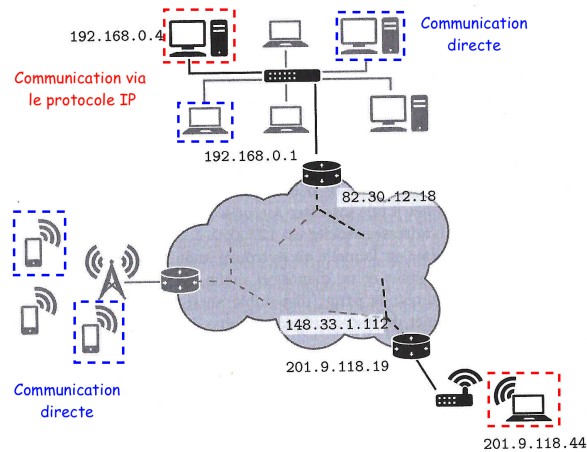
## III. Couche 2 : internet

Intéressons-nous maintenant à la **couche 2 : internet**.



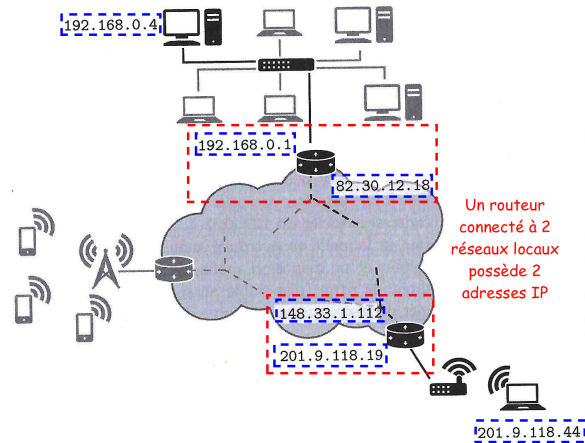
Si deux machines appartiennent au même réseau local, elles sont reliées physiquement et peuvent donc communiquer directement (cf. couche 1).

Si deux machines appartiennent à des réseaux locaux différents, c'est le **protocole IP** qui leur permet de communiquer.



Ce protocole IP offre deux fonctionnalités principales :

- il permet d'associer à chaque machine un identifiant (unique) appelé **adresse IP**
- il assure le **routing** c'est à dire la transmission des données de proche en proche



## 1. Adresse IP

Chaque machine, accessible depuis Internet, est identifiée par une adresse IP.

### 💡 Analogie avec le courrier postal

- Chaque résidence, accessible depuis la route, est identifiée par une adresse postale

Afin d'assurer le routage de l'information, toute adresse IP est composée de 2 parties :

- **ID sous-réseau** qui identifie le sous-réseau logique<sup>[9]</sup> concerné
- **ID machine** qui identifie la machine au sein de son sous-réseau logique

### 💡 Analogie avec le courrier postal

- ID sous-réseau → NPA et ville
- ID machine → Rue et numéro

Dans la version 4 du protocole IP (IPv4)<sup>[10]</sup>, une adresse IP est :

- composée de quatre entiers compris entre 0 et 255 (4 octets)
- complétée par un **masque de sous-réseau**<sup>[11]</sup> : /8, /16, /24, /19, ...



Le masque de sous-réseau /24 correspond à l'adresse IP 255.255.255.000 :

$$\underbrace{11111111 . 11111111 . 11111111}_{24 \text{ bits égaux à } 1} . \underbrace{00000000}_{8 \text{ bits égaux à } 0}$$



Considérons la machine d'adresse IP 132.1.65.2 avec le masque /19

1. Écrire les 4 octets de cette adresse IP et ceux du masque.
2. Effectuer l'opération logique ET sur ces deux mots binaires.
3. Que représente le résultat ?

[9]. Pensez à un réseau physique où les machines peuvent communiquer directement (sans routeur)

[10]. La version 6 qui sort du cadre de ce cours a été définie à la fin des années 1990, et est toujours en cours de déploiement.

[11]. à l'aide de la notation CIDR (Classless InterDomain Routing)



Considérons les adresses IP suivantes avec le masque /21 :

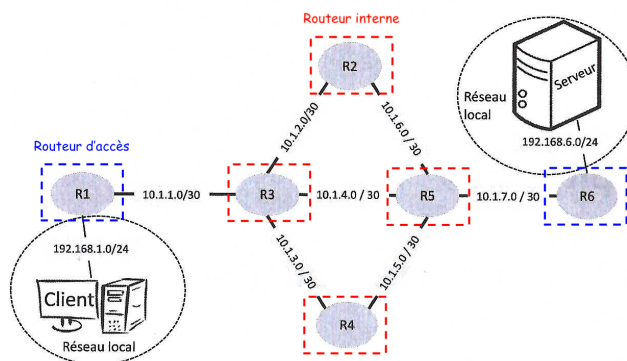
- 192.168.129.10
- 192.168.135.200
- 192.168.145.1

Lesquelles dénotent des machines du même sous-réseau?

## 2. Routage

Ce sont les **routeurs** qui permettent d'acheminer les paquets dans le réseau.

Plus précisément, les machines qui cherchent à communiquer sont connectées à des **routeurs d'accès** via leur réseau local (Ethernet, Wi-Fi, 4G). Puis, ce sont les **routeurs internes** qui prennent le relais, ils sont connectés entre eux sur de plus longues distances à l'aide de fibres optiques, de câbles téléphoniques ou de liaisons par satellite.



Lorsqu'un routeur reçoit un paquet, il récupère l'adresse IP de destination et choisit vers quel routeur voisin retransmettre ce paquet. Pour cela, il utilise une **table de routage** qui associe des adresses de destination à des adresses de routeurs. Si l'adresse de destination est inconnue, le paquet est orienté vers un routeur par défaut.



S'il existe plusieurs routes pour acheminer un paquet, comment le chemin est-il choisi? Les tables de routage évoluent-elles de manière dynamique au gré des pannes, des liaisons rompues ou des liens nouveaux?

Ces questions sont résolues grâce aux **protocoles de routage** qui sont des algorithmes qui permettent aux routeurs de s'échanger les informations dont ils disposent sur les routes du réseau, de manière à pouvoir choisir la « meilleure » route possible.

Il s'agit d'un processus **décentralisé** et **dynamique**, c'est à dire que les routeurs ont tous la même fonction, aucun d'eux n'a de privilège pour diriger ces configurations, et les tables de routage sont recalculées dès qu'un changement apparaît dans le réseau.

Illustrons le fonctionnement du protocole RIP<sup>[12]</sup> au niveau du routeur R1.

Initialement, le routeur R1 dispose des informations concernant ses voisins immédiats.

destination	passerelle	interface	distance
10.1.1.0/30		eth0	1
192.168.1.0/24		wlan0	1

[12]. Routing Information Protocol



Cette table nous indique que R1 est directement<sup>a</sup> connecté au sous-réseau :

- 10.1.1.0/30 via une interface Ethernet (eth)
- 192.168.1.0/24 via une interface sans fil (wlan)

<sup>a</sup>. pas de passerelle (interface routeur) ou distance égale à 1

Après cette phase d'initialisation, le routeur R1 poursuit le protocole en échangeant des demandes RIP avec R3 qui doit accuser réception en lui envoyant sa table en réponse.

R1 peut alors mettre à jour sa table en ajoutant de nouvelles destinations et/ou en optimisant les destinations connues (distance minimale).

destination	passerelle	interface	distance
10.1.1.0/30		eth0	1
192.168.1.0/24		wlan0	1
10.1.2.0/30	10.1.1.2	eth0	2
10.1.3.0/30	10.1.1.2	eth0	2
10.1.4.0/30	10.1.1.2	eth0	2

En répétant ces demandes RIP et en mettant à jour leurs tables, les routeurs vont finir par avoir la même « vision » du réseau et les meilleures routes à suivre.

destination-réseau	passerelle	interface	distance
10.1.1.0/30		eth0	1
192.168.1.0/24		wlan0	1
10.1.2.0/30	10.1.1.2	eth0	2
10.1.3.0/30	10.1.1.2	eth0	2
10.1.4.0/30	10.1.1.2	eth0	2
10.1.7.0/30	10.1.1.2	eth0	3
192.168.6.0/24	10.1.1.2	eth0	4



Que devient la table de routage de R1 si le sous-réseau 10.1.4.0/30 tombe en panne et si le routeur R4 est hors-service ?

Dans ce protocole, les distances ne peuvent pas dépasser 15 de manière à diminuer le délai de convergence, c'est à dire le temps nécessaire pour que tous les routeurs aient la même vue du réseau. Ce protocole est donc limité à des réseaux de petites tailles.



Voyez-vous une autre limite à ce protocole, concernant l'optimisation des routes ?

La notion de distance utilisée pour optimiser les routes ne garantit pas qu'elles soient les meilleures en terme de débit puisque la nature des liaisons (fibre optique, câbles téléphoniques, satellite) n'est pas intégrée dans les messages RIP.

C'est pour pallier ces défauts que le protocole OSPF<sup>[13]</sup> a été développé dans les années 90, mais nous nous arrêtons là pour la couche 2.

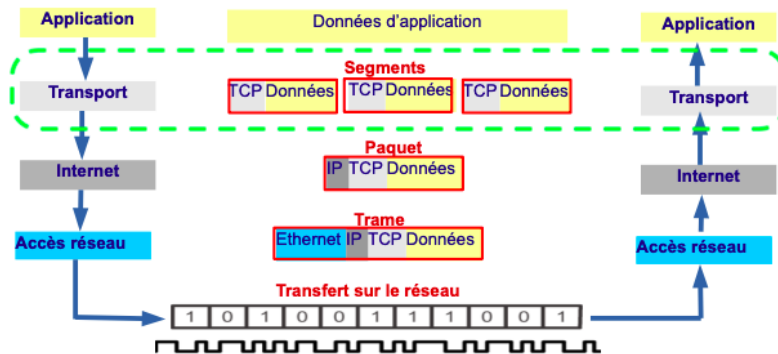
#### Filius (2/4)

- Exercice 3 : interconnexion de réseaux avec un routeur
- Exercice 4 : routage dans une interconnexion de réseaux

[13]. Open Shortest Path First

## IV. Couche 3 : transport

Intéressons-nous ensuite à la **couche 3 : transport**.



### 1. UDP

Le protocole IP permet de gérer une communication entre deux machines physiques distantes, appartenant à des réseaux locaux différents.

Mais dans la pratique, on a en fait besoin d'établir une communication entre deux applications (programmes) qui s'exécutent sur ces machines physiques distantes, l'une appelée « client » et l'autre « serveur ».

Par exemple, un navigateur Web est le **client** qui va demander au **serveur** Web de lui transmettre la page Web associée à l'URL exprimée par l'utilisateur dans la barre d'adresse pour pouvoir l'afficher (on y reviendra).

?

Sur une machine physique, il existe en général plusieurs applications capables de communiquer en réseau. Comment peut-on alors identifier, pour une même adresse IP, ces différentes applications ?

C'est le **numéro de port** qui résout ce problème d'identification. Par convention :

- port **80** : consultation d'un serveur Web via le protocole **HTTP**
- port **443** : consultation sécurisée d'un serveur Web via le protocole **HTTPS**
- port **53** : résolution d'un nom de domaine en adresse IP via le protocole **DNS**
- port **25** : envoi d'un courrier électronique via le protocole **SMTP**
- port **143** : récupération d'un courrier électronique via le protocole **IMAP**

Le protocole UDP se limite à cette fonctionnalité d'identification, il ne garantit pas :

- la bonne réception d'un datagramme UDP (équivalent d'un segment TCP)
- l'ordre de réception de plusieurs datagrammes UDP

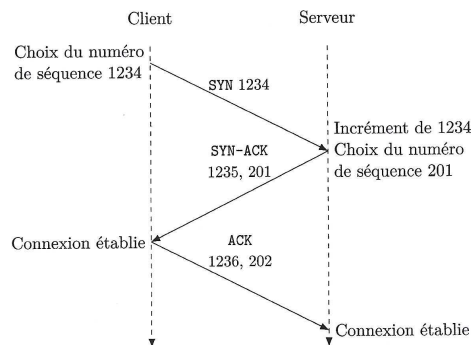
Il est utile pour transmettre « en temps réel » de petites quantités de données lorsque la perte est préférée à l'attente de la retransmission. Le DNS (on y reviendra), la voix sur IP (appels WhatsApp) ou les jeux en ligne en sont des utilisations typiques.

## 2. TCP

Le protocole TCP résout les limites du protocole UDP grâce à un système d'accusés de réception. Il est utilisé, entre autres, par les protocoles HTTP(S), SMTP et IMAP.

Il permet d'envoyer à une application, toujours identifiée par un port, des données (éventuellement très grande) dans l'ordre, de détecter les erreurs de transmission, et de retransmettre si besoin automatiquement les segments perdus voire corrompus <sup>[14]</sup>.

Avant de démarrer l'échange des données entre les deux applications (Client/Serveur), le protocole TCP établit une connexion entre elles en 3 étapes :



Une fois la connexion établie, le client et le serveur partagent une information commune (la paire de numéros de séquence), ils peuvent alors garantir des échanges sans perte d'information et dans l'ordre. Notons que lors du dernier ACK, le client peut déjà envoyer des données.

Imaginons (de manière simplifiée) que le client souhaite envoyer un segment de 3000 octets au serveur, découpé <sup>[15]</sup> en 3 segments de 1000 octets.

Il envoie donc ( $S_1$ , 1236), ( $S_2$ , 2236) et ( $S_3$ , 3236). En réponse, le serveur renvoie des ACK étiquetés avec les numéros correspondants (1236, 2236 et 3236).



Comment ce protocole permet-il de gérer :

- la perte d'un segment?
- l'ordre des 3 segments?

### ➤ Filius (3/4)

- Exercice 5 : ajout d'un serveur Web

### ➤ Client/Serveur avec Python (1/2)

Écrire ce programme dans un fichier nommé **serveur.py** :

```
from socket import socket

serveur = socket()
serveur.bind( ('0.0.0.0', 9999) )
serveur.listen()

while True :
    (sclient, adclient) = serveur.accept()
    donnees = sclient.recv(1000)
    while donnees :
        print(donnees.decode(), end='')
        donnees = sclient.recv(1000)
    sclient.close()
```

Exécuter le dans un terminal.

[14]. L'intégrité des données repose sur une somme de contrôle, mais cela sort du cadre de ce cours

[15]. pour des raisons matérielles ou de performance

## > Client/Serveur avec Python (2/2)

Écrire ce programme dans un fichier nommé **client.py** :

```
from socket import socket

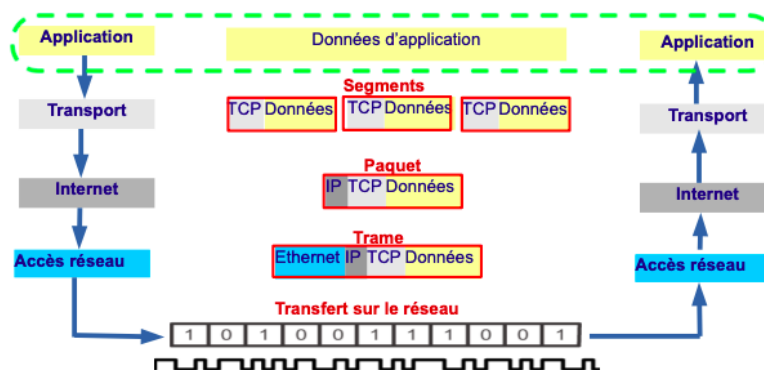
serveur = socket()
serveur.connect( ('127.0.0.1', 9999) )

phrase = input()
while phrase != "FIN" :
    phrase = phrase + "\n"
    serveur.send(phrase.encode())
    phrase = input()
```

Exécuter le dans un autre terminal, puis taper un texte...

## V. Couche 4 : application

Intéressons-nous enfin à la **couche 4 : application**.

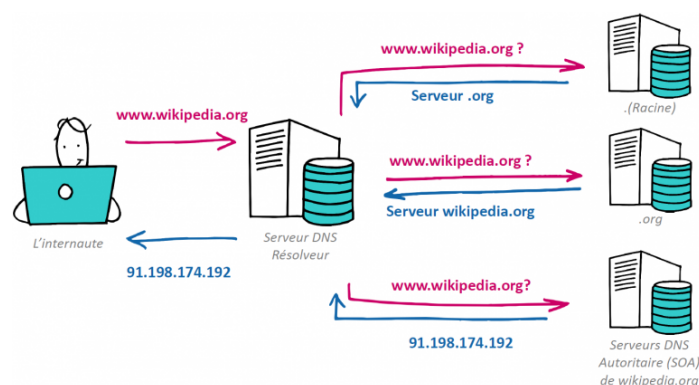


### 1. DNS

?

Le seul moyen d'identifier une machine sur Internet est son adresse IP, mais alors que se passe-t-il lorsque l'on indique au navigateur Web que l'on souhaite se connecter au site **www.wikipedia.org**?

Le protocole DNS exploite un ensemble de serveurs hiérarchisés<sup>[16]</sup> qui agissent comme des annuaires pour déterminer l'adresse IP recherchée.



[16]. Les serveurs « racines » connaissent les serveurs à contacter pour les domaines de premier niveau (.org). Ces derniers connaissent les serveurs à contacter pour les domaines de second niveau (wikipedia.org). Ces derniers connaissent les adresses des sous-domaines (www.wikipedia.org)

## Filius (4/4)

- Exercice 6 : ajout d'un serveur DNS

## 2. HTTP(S)

En réalité, l'URL <sup>[17]</sup> commence toujours par le protocole utilisé par le navigateur Web pour dialoguer avec les serveurs Web :

**`https://www.wikipedia.org`**

La syntaxe (simplifiée) d'une URL est la suivante :

**`protocole://nom-ou-adresse/document`**

Le « document » désigne une ressource (fichier html, ...) qui est stockée sur le serveur identifié par son nom ou son adresse IP, et qui sera affichée dans le navigateur.

### Passage de paramètres à un serveur

Saisir dans la barre d'adresse de son navigateur les URL suivantes :

- `https://fr.wikipedia.org/w/index.php`
- `https://fr.wikipedia.org/w/index.php?search=Informatique`
- `https://fr.wikipedia.org/w/index.php?search=Ordinateur`
- `https://fr.wikipedia.org/w/index.php?search=Informatik`
- `https://fr.wikipedia.org/wiki/Informatique#Algorithmique`

Comment le serveur interprète-t-il ces URL?

---

[17]. Uniform Resource Locator