NSI 1ère - Réseaux

QK

Réseaux

Programme :

Architecture

- Transmission de données dans un réseau
- Protocoles de communication
- Architecture d'un réseau

Transmission de données dans un réseau

Comment communiquer ?

- Déjà difficile de communiquer à 10, alors comment communiquer tous ensemble, en même temps ?

De quoi a-t-on besoin pour communiquer ?

Exemple 1 : la parole

- un émetteur
- un récepteur
- un support de transmission (l'air)

Exemple 2 : téléphone

On doit ajouter un intermédiaire entre la parole et l'électronique. On transforme la parole en signaux électriques, ils sont transmis et à nouveau transformés en paroles. Il y a *encapsulation* de l'information.

Exemple 3 : le courrier

- un émetteur
- un récepteur
- un support de transmission (la lettre)
- un contentant (l'enveloppe)
- un intermédiaire (la poste)

On retrouve l'encapsulation de l'information

Comment appliquer l'encapsulation aux ordinateurs ?

Le modèle OSI

- Apparu en 1984 (donc après internet !) le modèle OSI est un ensemble de normes que doivent respecter les ordinateurs pour communiquer sur internet.
- Il tient compte des communications existantes mais aussi de leurs évolutions futures

C'est un modèle en couches successives depuis le métal jusqu'à l'utilisateur.

Le modèle OSI est un modèle *théorique* dont TCP/IP s'est inspiré

Les couches du modèle OSI

1. Physique : Support de transmission pour la communication

   Associée au *hub*

2. Liaison : Connecter les machines entre elles sur un *réseau local*

   Associée au *switch*

3. Réseau : Interconnecter les réseaux entre eux

   Associée au *routeur*

4. Transport : Gérer les connexions applicatives
5. pas utilisé dans TCP/IP
6. pas utilisé dans TCP/IP
7. L'application : le patron

   Associée au *proxy*

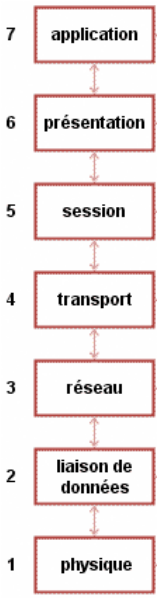


Figure 1: Le modèle OSI

Principes

Les grandes règles du modèle OSI

- Chaque couche est indépendante
- Chaque couche ne peut communiquer qu'avec une couche adjacente

Chaque couche est indépendante

- Les informations d'une couche ne peuvent être utilisées dans une autre
- Exemple : l'adresse IP (couche 3) ne pourra être utilisée ailleurs
- Cela permet l'évolution des communications dans le temps
- Elles sont interchangeables : IPv4 va devenir IPv6 sans qu'on doive tout réécrire

Chaque couche ne peut communiquer qu'avec une couche adjacente

Exemple : on entre `google.com` dans le navigateur.

Le navigateur (application) s'est adressé aux couches réseau (1 à 4) pour qu'elles

transmettent l'information à l'application sur la machine demandée (le serveur web de google)

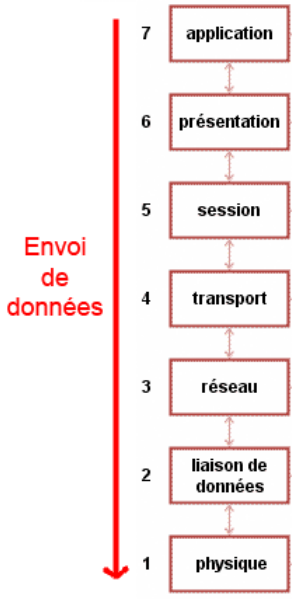


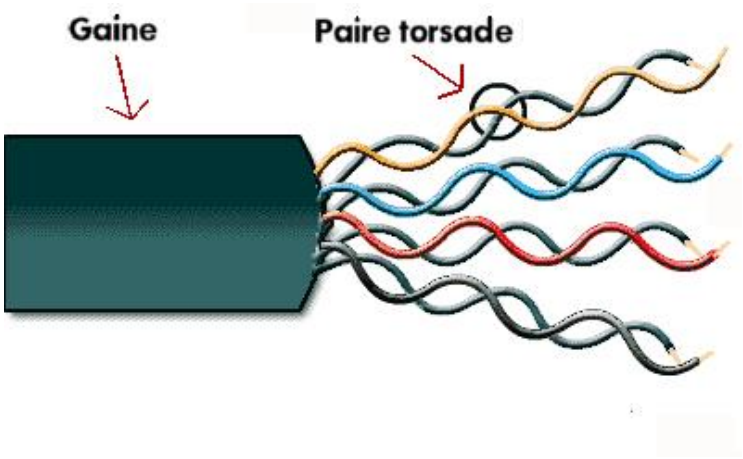
Figure 2: Toutes les couches sont parcourues de haut en bas pour transmettre

Couche 1 : physique

Matériel de la couche 1

Zappons l'aspect historique des câbles coaxiaux qui ne sont plus utilisés.

Les cables torsadés



Les cables torsadés

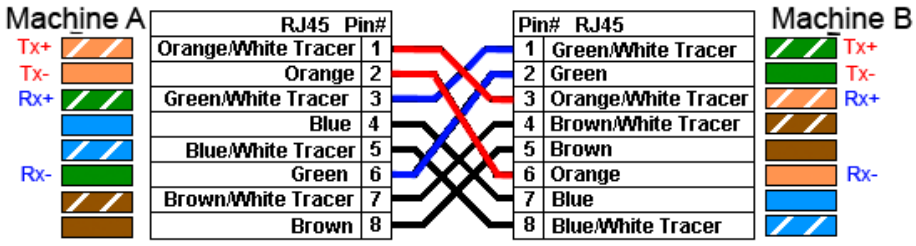
Une gaine plastique protège 8 fils, eux mêmes protégés et torsadés par paires (meilleure protection contre les champs magnétiques).

- Généralement une paire pour envoyer, une paire pour recevoir
- Parfois les 8 fils sont utilisés
- Selon le débit : 10BT (10 Mbps), 100BT (100 Mbps), 1000BT (1000 Mbps)
- 90% des cablages courants : économique, robuste.
- La prise est RJ45
- On la branche dans un *switch* ou une prise femelle

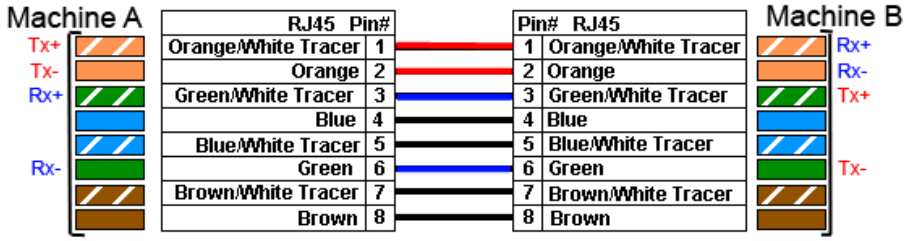
Câbles croisés, câbles droits

- Deux vieux ordinateurs entre eux (rare), il faut **croiser** les câbles : l'*émission* sur la *réception*

Figure 3: prise RJ45



- Un ordinateur et un switch : on peut utiliser des **câbles droits**



Matériel réseau : le hub

Le **hub** relie plusieurs prises RJ45 entre elles.

Défaut majeur : envoie toutes les infos à tout le monde !

Énorme gaspillage de bande passante (mais ça fonctionne)

Matériel réseau : le switch

Le **switch** adresse les infos aux *machines destinataires*

On économise de la bande passante… mais c'est plus compliqué.

Le Wi-Fi

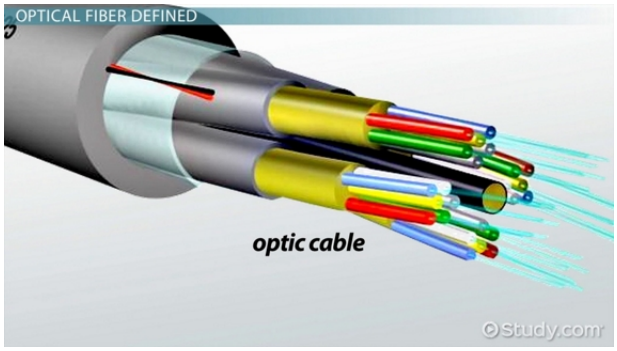
Apparu à la fin des années 1990, il permet un cablage virtuel entre deux machines.

Il utilise la modulation des ondes radio. Les différentes normes 802.11 représentent les variations de débit.

La fibre optique

Transporte l'information avec la **lumière** et non plus des 0 et 1.

Coûteuse mais beaucoup plus efficace. Utilisée au *coeur du réseau.*



Deux types de fibres

- **Monomode** : une seule longueur d'onde (une seule couleur). Plus fiable, longues distances
- **Multimode** : toutes les longueurs d'onde (toutes les couleurs). Faible distance.

Record : 8000 km avec une seule fibre.

En pratique : on relie l'Europe aux USA en monomode avec des répétiteurs tous les 60 km.

Topologie d'un réseau

La **topologie d'un réseau** décrit la manière dont sont reliées les machines. On en rencontre souvent 2 :

- Réseau en étoile localement (routeur / switch au centre)
- Réseau maillé (mesh) sur internet

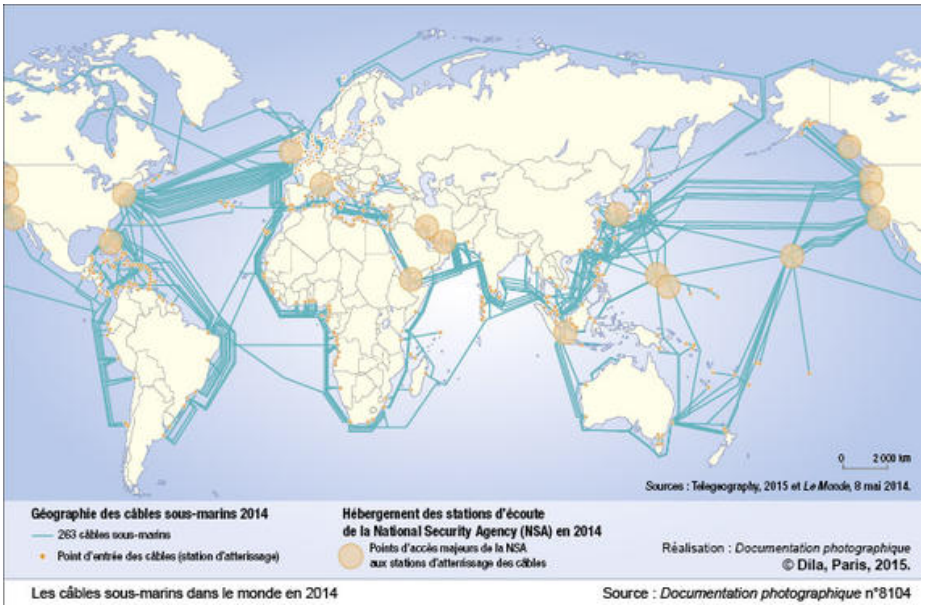


Figure 4: Carte des câbles sous marin

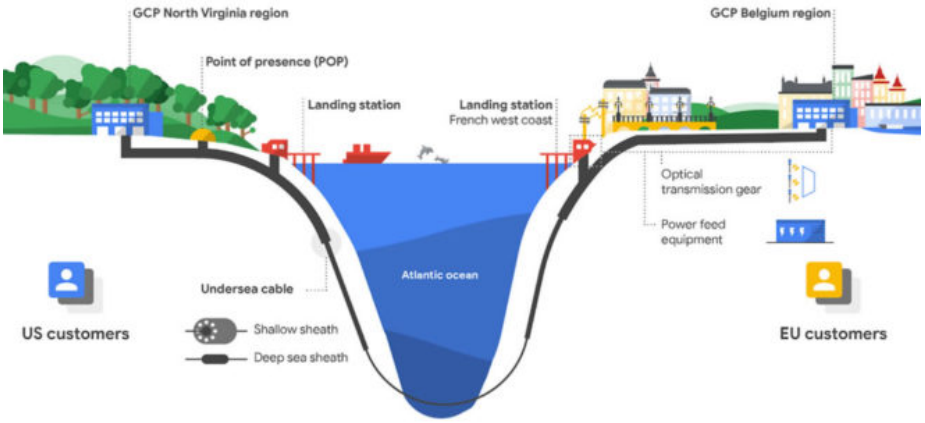
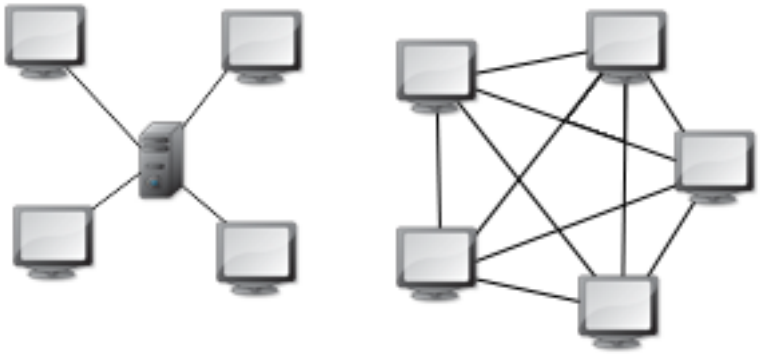


Figure 5: Câbles sous marin en détail



Comment éviter les collisions d'information ?

Deux machines se parlent en meme temps : **collision.**

Solution : organiser **le droit à la parole.**

1. Deux machines A et B parlent en même temps.
2. Elles détectent la collision.
3. Elles attendent toutes les deux un temps aléatoire. 2 s pour A et 3 s pour B.
4. Après 2 s, A recommence à parler.
5. Après 3 s, B voit que A parle et attend son tour.
6. Dès que A a fini, B peut parler.

Couche 2 : liaison

Connecter des machines sur un réseau local

Les rôles de la couche 2 sont de :

- connecter des machines sur un réseau local,
- détecter des erreurs de transmission (détection seulement…).

L'adresse MAC

Une adresse physique **unique au monde** par carte réseau : **l'adresse MAC**

Codée sur 6 octets en hexadécimal. Exemple : `44:8a:3a:2d:b2:f4`

Adresse particulière : *broadcast* (diffuser) : `ff:ff:ff:ff:ff:ff` Ecrire à cette adresse revient à écrire à toutes les machines locale.

Est-ce sécurisé ? Pas du tout.

Il est facile de se faire passer pour un autre en changeant temporairement d'adresse MAC. (On le fera)

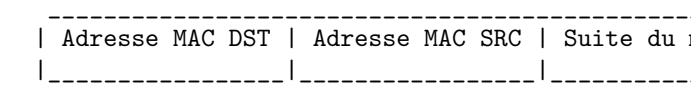
Connaître ses adresses :

Sous UNIX avec `ip a` ou `ifconfig`, sous windows avec `ipconfig`

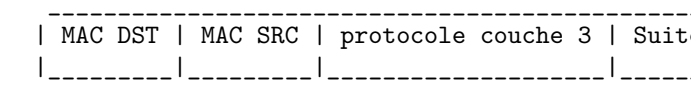
```
quentin@q ~ $ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group defa
    link/ether 44:33.5b:dd:2b:16 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.40/24 brd 192.168.0.255 scope global dynamic enp3s0
       valid_lft 33778sec preferred_lft 33778sec
    inet6 fe80::468a:5bff:fe5d:b2f4/64 scope link
       valid_lft forever preferred_lft forever
```

Le protocole Ethernet

Format d'une trame Ethernet



En pratique, **avant d'arriver en couche 2**, le message est passé par la couche 3 ! La `Suite du message` contient donc le protocole de celle-ci.



Et les erreurs ?

Elles sont détectées grace à un **CRC**, code de correction d'erreur.

- Chaque message a un CRC calculé par l'émetteur ajouté à la fin de celui-ci.
- Le récepteur calcule le CRC du message, **s'il correspond à celui reçu, il n'y a pas d'erreur** (ou pas souvent).

Trame complète

```
 ____________________________________________________
| MAC DST | MAC SRC | prot couche 3 | Suite msg | CRC |
|_________|_________|_______________|___________|_____|
```

Taille d'une trame Ethernet

- les adresse MAC font 6 octets
- le protocole de couche 3 est codé sur 2 octets
- le CRC sur 4 octets

18 octets pour l'en-tête (fixe)

La trame complète varie entre 64 octets et 1518 octets

Le matériel de couche 2 : le switch

Le **switch** (commutateur) sert à aiguiller les message selon leur adresse MAC

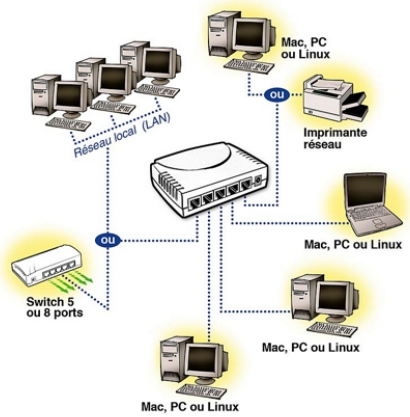


Figure 6: Réseau local

VLAN : séparer un switch

Les plus gros switch ont régulièrement 256 ports !

Par sécurité / commodité on sépare les machines, par exemple :

- Réseau des élèves

- Réseau administratif

On crée alors deux VLAN au sein desquels les machines communiquent. Une machine d'un VLAN ne communique donc pas avec une d'un autre VLAN (Virtual Local Area Network).

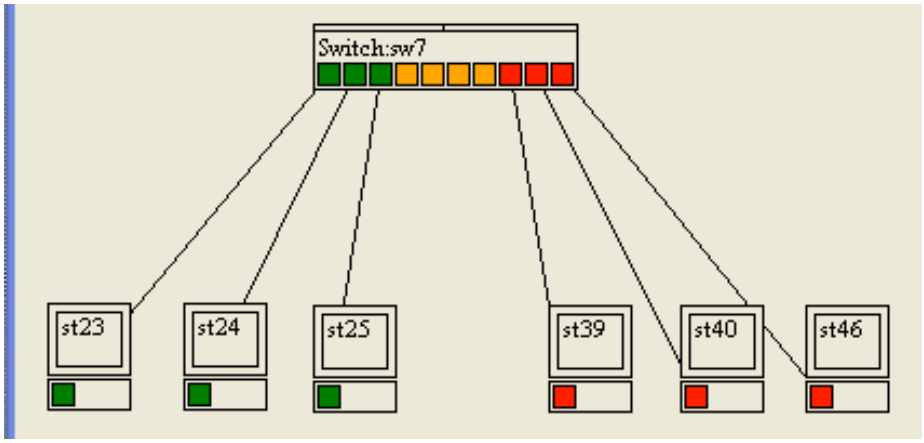


Figure 7: 2 VLAN sur un même switch

Le protocole IP et la couche 3

La couche 3

Les couches 1 et 2 permettent de communiquer dans un réseau local

Le rôle de la couche 3 et d'interconnecter les réseaux

Comment découvrir le chemin emprunté ?

Sous windows il existe `tracert` et sous UNIX `traceroute` qui renvoient le chemin emprunté pour nous relier à une machine

```
10:06 quentin@q ~ $ traceroute google.com
traceroute to google.com (172.217.18.206), 30 hops max, 60 byte packets
 1  gateway               (192.168.0.254)      0.286 ms
 2  can59-6-247-31-24.fr  (247.31.24.254)     23.055 ms
 3  213.228.12.190        (213.228.12.190)    25.696 ms
 4  p11-crs16-1-be1122.fr (194.149.163.145)   31.385 ms
```

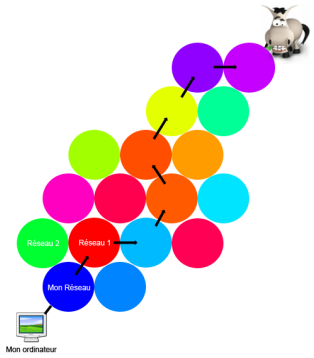


Figure 8: Plusieurs routes sont possibles

```
...
8  66.249.94.83           (66.249.94.83)      28.309 ms
9  par10s38-in-f14.1.fr   (172.217.18.206)    27.182 ms
```

9 étapes pour rejoindre Google en Californie. *Résultat épuré*

Lire un traceroute

- Chaque ligne représente une étape depuis la ligne précédente
- On voit l'adresse IP de la machine qu'on a atteint
- le temps est celui qu'il faut pour établir la connexion jusqu'à cette machine
- Certaines étapes sont parfois masquées quand on entre dans un sous réseau

Les adresse IP

Une adresse IP est à la fois **celle du réseau et de la machine**

Il existe plusieurs normes :

- IPv4 : codée sur 32 bits (= 4 octets)
- IPv6 : codée sur 128 bits (= 32 octets)

On s'intéressera d'abord à IPv4 qui est plus fréquente

- Chaque octet est en décimal, séparé par un point : 192.168.0.1
- La première est 0.0.0.0 et la dernière 255.255.255.255
- Pour utiliser, l'IP suffit. Pour administrer, il faut le masque

Masque de sous réseau

Le masque indique la séparation entre partie réseau et partie machine de l'IP

J'éviterai volontairement les difficultés mais sachez qu'elles sont nombreuses.

Étude détaillée en poursuite d'étude en info.

Exemple de masque de sous réseau

Exemple : IP : 192.168.0.1, Masque : 255.255.0.0

Traduites en binaire cela donne :

```
255.255.0.0  -->   11111111.11111111.00000000.00000000
192.168.0.1  -->   11000000.10101000.00000000.00000001
```

Les bits à 1 du masque indiquent le réseau, les autres indiquent la machine.

Donc : `réseau = 192.168` et `machine = 0.1`

Deux adresses IP qui n'ont pas la même partie réseau ne sont pas dans le même réseau

Masque et découpage de réseaux : difficile.

Tant que la coupure se fait entre deux octets c'est facile.

Dans ce cas il est inutile de passer en binaire !

À partir d'ici, j'ampute largement ce qui est pénible

Les masques ne peuvent pas être écrits n'importe comment

Une précision importante : dans le masque, les bits à 1 se suivent.

```
Masque : 11111111.11111111.00000000.00000000 : correct
Masque : 11111111.11100011.00000000.00000000 : incorrect
```

Exemple de masque quelconque en décimal

- Les masques commencent éventuellement par une série de 255
- Ils se terminent par : 0, 128, 192, 224, 240, 248, 252, 254 ou 255

Le seul cas facile est 255 (dommage).

- `255.255.224.0` est correct

- `255.192.224.0` est incorrect

Plage d'adresse

Plus il y a des bits à 0 dans le masque, plus il y a d'adresses disponibles.

Ce sont les **plages d'adresses**

Exemple `255.255.0.0`, il y a $2^{16} = 256 \times 256 = 65.536$ adresses disponibles

Première et dernière adresse d'une plage

Exemple : IP : 192.168.0.1, Masque : 255.255.0.0

Traduites en binaire cela donne :

```
11111111.11111111.11110000.00000000 --> 255.255.240.0
11000000.10101000.00000000.00000001 --> 192.168.0.1

La plage s'étend de :
11000000.10101000.00000000.00000000 --> 192.168.0.0
à
11000000.10101000.00001111.11111111 --> 192.168.15.255
```

Adresse de réseau, adresse de broadcast

- La première adresse d'un réseau désigne le réseau lui même.
- La dernière désigne le broadcast. **broadcast** = diffusion. Écrire au broadcast = écrire à tlm

Des adresses particulières : RFC 1918

RFC : norme d'une technologie utilisée sur Internet

Certaines adresses IP sont réservées. Grosso modo on distingue :

- Sa proche machine : `127.0.0.0 = localhost`
- Réseau local :

```
192.168.x.x / 255.255.0.0
172.16.0.0  / 255.240.0.0  <-- où s'arrête cette plage ?
10.x.x.x    / 255.0.0.0
```

- Internet : à peu près tout le reste

Notation simplifiée des masques

Les masques ne servant réellement qu'à l'administration, on peut se permettre de simplifier leur notation.

On donne souvent un seul nombre : **le nombre de bits à 1 dans le masque.**

Exemples

```
Masque : 255.255.255.0 <=> /24
Masque : 255.255.240.0 <=> /22
```

On rencontre alors des IPv4 notées : `162.168.0.1 /24`

Où réside la difficulté ?

Les adresses d'un réseau doivent se suivre (comme les bits à 1 du masque)

La difficulté est donc de découper un gros réseau **en sous réseaux de tailles suffisantes** pour donner une adresse à chacun

Un exemple amputé et non justifié

`IP 192.168.0.1, Masque 255.224.0.0`

- On prend 224 et 168 (le même octet que le dernier significatif du masque).
- Nombre magique : 256 - 224 = **32** (parce que).
- On cherche, parmi les multiples de 32 (≤ 256) le dernier avant **168** : 160
- On prend le multiple suivant moins 1 : 192 - 1 = 191
- Première adresse : `192.160.0.0` –> réseau
- Dernière adresse : `192.191.255.255` –> broadcast

Ça fonctionne mais les justifications prennent 5 pages.

L'avenir des IP : IPv6

- IPv4 : 32 bits (décimal pointé)
- IPv6 : 128 bits (hexadécimal :)

Exemple `fe80::468a:5bff:fe5d:b2f4 / 64`

Combien d'adresses IPv4 ? $2^{32} \approx 4 \times 10^9$

On manque cruellement d'adresse IPv4 sur internet.

De nombreux réseaux se partagent la même IPv4… Ça devient bien compliqué et on ne sais pas à qui on parle.

Nous n'étudierons pas les IPv6

Le routage : protocole IP

IP (*Internet Protocol*) désigne une adresse (IPv4, IPv6) mais aussi le **protocole de communication** de la couche 3.

Il suffit de connaître l'IP d'une machine pour lui écrire.

- On connaît son propre masque, donc son propre réseau.
- On peut donc calculer sa propre plage réseau qui est imposée par le masque.
- Une IP dans notre plage est dans notre réseau.
- Une IP en dehors n'est pas dans notre réseau.

Il n'est pas nécessaire de connaître le masque pour envoyer un message à une machine ! (Ouf)

Il reste une question : **comment sont dirigés les messages en dehors de notre réseau ?**

Routage

Un message IP est un **paquet** (datagramme pour être précis)

```
 ________________________________________
| ??? |IP source |IP destination | Données |
|_____|__________|_______________|_________|
```

- `???` : des infos (longueur, protocole du reste du message, checksum etc.)
- Certains messages ne peuvent être lus que s'ils sont complets, on peut *découper un paquet en fragments*.

  La partie initiale (`???`) contient alors de quoi situer le fragment dans l'ensemble.
- `IP source` : en premier cette fois, *la couche inférieure a déjà éliminé les message ne nous intéressant pas.*

Encapsulation

Le paquet n'est qu'une partie du message. Il est *encapsulé* dans la *trame* (couche 2).

```
7 Application                    | Bonjour |
4 Transport                  | 4 | Bonjour |
3 Réseau                 | 3 | 4 | Bonjour |
2 Liaison            | 2 | 3 | 4 | Bonjour | CRC |
```

```
1 Physique              des impulsions électriques
```

- Départ : on part d'en haut et on ajoute des capsules
- Arrivée : on part d'en bas et on en enlève

Reprenons la trame Ethernet

Elle ressemble un peu plus à :

```
_____________________________________________________
| MAC | MAC | protocole | En tête | prot | msg | CRC |
| DST | SRC |     3     |   3     |   4  |     |     |
|_____|_____|___________|_________|______|_____|_____|
```

On utilisera **Wireshark** (logiciel de capture de paquets) pour analyser quelques trames.

Relier les réseaux : le travail du routeur

Un **routeur** est une **ordinateur** qui dispose de plusieurs interfaces (=cartes réseau) et est configuré pour aiguiller l'information.

- Votre box : internet (ADSL, Fibre, Satellite etc.), 100BT / 1000BT (RJ45), Wifi
- N'importe quel PC muni de 2 cartes réseau et configuré comme tel.
- Un téléphone en "partage de connexion" (*Tethering*). Le téléphone crée un réseau local (Wifi) quand on partage la connexion. Cette possibilité a longtemps été interdite puis facturée par les opérateurs.

Résumé matériel :

1. Physique : **câble** 1000BT + **prise** rj45
2. Liaison : **Hub** (équipement réseau)
3. Réseau : **Switch** (équipement réseau)
4. Transport : **Routeur** (Ordinateur)

Plan d'un réseau complet

Internet, deux routeurs, un hub, un switch et un serveur FTP.

Figure 9: Des routeurs Cisco

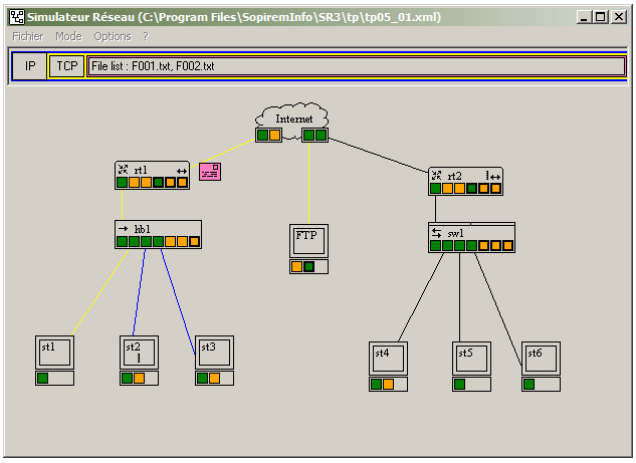


Figure 10: Un réseau complet

Mais internet avec l'ADSL ?

Internet par **ADSL** utilise *les fils de cuivre* (installés gratuitement par **l'état** dans les années **70** !!) et des prises RJ11 originellement dédiés à la *téléphonie analogique*. On transmettait la voix sous forme de *modulation de fréquences*.

La technologie ADSL est plus proche de la téléphonie analogique classique que du 1000BT + RJ45 ou de la fibre.

Il faut un MODEM pour Moduler (envoyer) / Démoduler (recevoir) l'information.

Plan d'un réseau avec ADSL

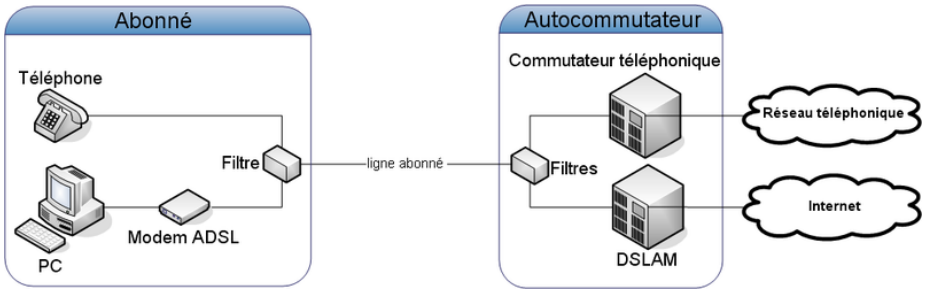


Figure 11: Principe de l'ADSL

Revenons à nos trames : dans la peau du routeur

Routeur : adresse MAC 00:11:22:33:44:55, adresse IP 192.168.0.1/24 Reçoit la trame suivante (de haut en bas) :

```
00:11:22:33:44:55
01:2B:45:56:78:ED
Protocole IP
Entête IP ???
IP : 10.0.0.1
IP : 136.42.0.28
Données à envoyer
CRC
```

Quelques questions :

- Quelle est l'adresse IP dont provient cette trame ?
- Quelle est l'adresse MAC dont provient cette trame ?

- Le destinataire est-il dans notre réseau ?

Quelques réponses :

- Quelle est l'adresse IP dont provient cette trame ?

  10.0.0.1

- Quelle est l'adresse MAC dont provient cette trame ?

  Impossible de la connaître ! 10.0.0.1 n'est pas dans notre réseau, on ne voit que l'adresse mac du dernier routeur

- Le destinataire est-il dans notre réseau ?

  Destinataire : 136.42.0.28, notre IP : 192.168.0.1/24 donc… Non ! Ce paquet est destiné à un autre réseau à qui nous allons le transmettre.

Que se passe-t-il quand notre routeur reçoit la trame ?

- La trame arrive à ma carte réseau (011001) qui envoie à mon système d'exploitation (OS).
- La couche 2 de mon OS interprète les 0 et 1 pour me donner l'adresse MAC de destination
- C'est mon adresse MAC 00:11:22:33:44:55 !

  Je lis la suite de l'en-tête de la trame pour voir :

  - qui m'envoie cette trame
  - et à quel protocole de couche 3 la couche 2 doit l'envoyer : IP

- J'envoie la trame en enlevant l'en-tête Ethernet, ce qui donne le paquet IP, au protocole IP.
- La couche 3 (le protocole IP) lit l'ensemble des informations de l'en-tête IP, puisque nous savons maintenant que ce datagramme nous est destiné. Et là, l'adresse IP de destination du datagramme n'est pas la nôtre…

Routage

Sans entrer dans les détails des algorithmes de routage…

- **Un routeur est un ordinateur qui possède plusieurs interfaces réseaux et qui accepte de relayer des paquets qui ne lui sont pas destinés.**
- Votre box internet est donc un ordinateur.

- Un routeur possède une *table de routage* qui lui indique où envoyer les paquets qui ne lui sont pas destinés.
- **Le routage est la base du fonctionnement d'internet.**
- réseaux immenses qui changent souvent : algorithmes de routage très complexes

Table de routage

Elle contient la liste des routeurs auxquels je peux envoyer mon paquet pour rejoindre une destination (un réseau)

Ces routeurs sont des *passerelles* entre deux réseaux.

```
Table de routage
Réseau à joindre      passerelle
192.168.1.0/24        10.0.0.253
192.168.122.0/24      10.0.0.45
192.168.8.0/24        10.0.0.254
...
```

Il existe aussi une **route par défaut** qu'on emprunte quand on ne sait pas où se rendre. Notée **defaut** ou **0.0.0.0/0**

Exercice de routage

Adresses des machines : questions

- Quelles sont les adresses des différentes clients ?
- Quelles sont les deux adresses du routeur ?

Adresses des machines : réponses

- Quelles sont les adresses des différentes clients ?

  haut : 192.168.0.1, 192.168.0.2, 192.168.0.3,

  bas : 192.168.1.1, 192.168.1.2

- Quelles sont les deux adresses du routeur ?

  réseau 0 : 192.168.0.254,

  réseau 1 : 192.168.1.254

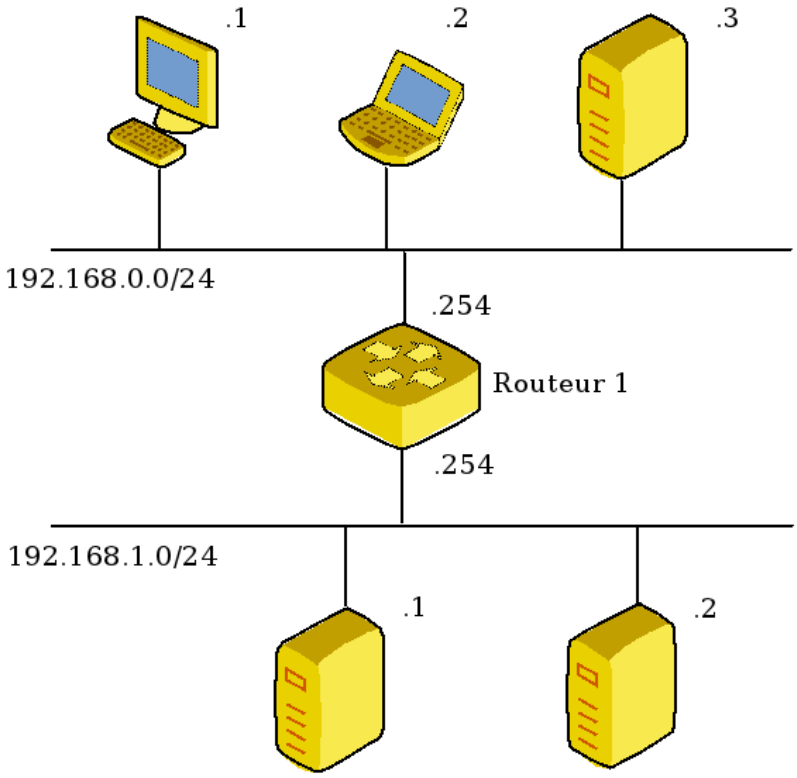


Figure 12: Réseau simple

Table de routage du routeur 1

1. indiquer les réseaux auxquels ma machine est connectée
2. route par défaut
3. tous les autres réseaux que je ne peux encore joindre
4. indiquer les passerelles

Construction de la table du routeur 1 :

1. indiquer les réseaux auxquels ma machine est connectée

```
Table de routage de routeur 1
Réseau à joindre      passerelle
192.168.0.0/24        ?
192.168.1.0/24        ?
```

2. route par défaut

   Dans notre cas particulier c'est inutile, le routeur est déjà relié à tout le monde.

3. tous les autres réseaux que je ne peux encore joindre

   Même remarque : je peux déjà joindre tout le monde.

Fin de la table

4. passerelles : la passerelle pour joindre un de **mes** réseaux est **mon** adresse.

```
Table de routage de routeur 1
Réseau à joindre      passerelle
192.168.0.0/24        192.168.0.254
192.168.1.0/24        192.168.1.254
```

Est-ce suffisant pour faire communiquer les machines ?

- Hélas non, nos clients ne savent pas à qui écrire !
- Les clients ont aussi une table de routage (tous les ordinteurs d'un réseau en ont une)

Table de routage d'un client

On applique la même méthode 1 2 3 4.

Construction de la table d'un client

IP du client : 192.168.0.1 /24

1. réseaux auxquels ma machine est connectée
2. route par défaut.

```
Table de routage de 192.168.0.1
Réseau à joindre      passerelle
192.168.0.0/24        ?
défaut                ?
```

3. indiquer tous les autres réseaux Nous avons déjà tout indiqué. La table ne change pas.
4. Passerelle. Quelle adresse de R1 indiquer ?

   Toujours celle de notre côté. Toujours indiquer une adresse **de son propre réseau.**

```
Table de routage de 192.168.0.1
Réseau à joindre      passerelle
192.168.0.0/24        192.168.0.254
défaut                192.168.0.254
```

Exercice complet : table de chaque routeur et d'une machine de chaque réseau.

Compléments sur les protocoles

ARP

Déclarer son adresse MAC

- D'après nos tables de routage précédentes, on écrit à des machines dont on connait l'IP.

  Mais… nos switch utilisent l'adresse MAC !

- Nécessité : un autre protocole permettant l'échange des adresses MAC.
- Comment envoyer un message à une machine dont on ne connaît pas l'adresse MAC ? Broadcast !

  On envoie au broadcast une **requête ARP** : *est ce que 192.168.0.254 peut m'envoyer son adresse MAC ?*

- Les broadcasts pouvant saturer le réseau, on crée donc une **table ARP**

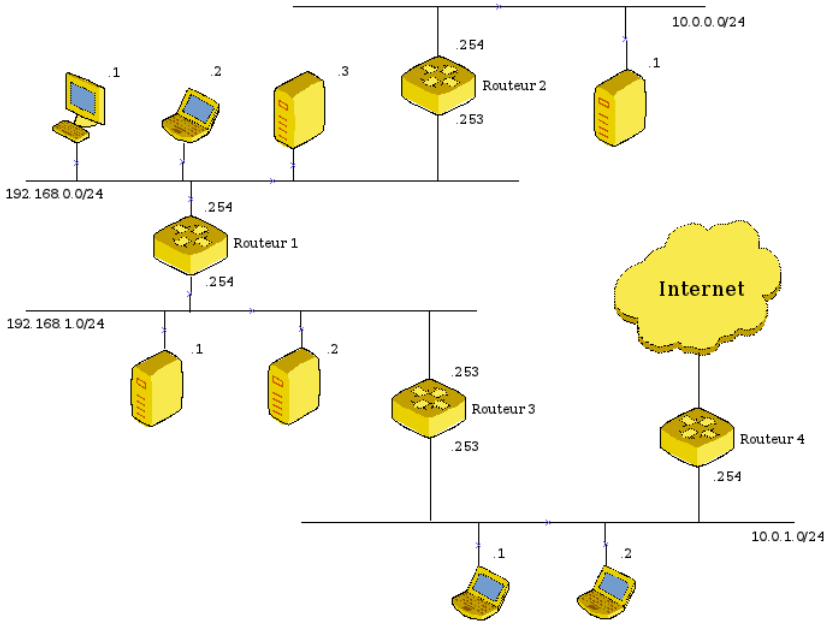


Figure 13: Réseau plus complexe

ICMP

Outil de diagnostique du réseau.

Ping et traceroute sont deux programmes qui utilisent ICMP.

- ICMP sert à indiquer automatiquement des erreurs quand elles surviennent ;
- ICMP peut fournir des outils pour étudier un problème réseau.
- On utilise astucieusement le TTL d'un message (nombre d'étapes qu'il peut encore franchir avant d'être détruit) pour savoir où celui-ci se situe après n étapes.

Couche application

Serveur / Client

Un serveur est un programme qui est en écoute sur une machine

Un client est une programme connecté à un service pour l'utiliser

Exemple : la machine A accède à un site web sur une machine B.

- Machine A : client web (navigateur) de la machine B
- Machine B : serveur web, serveur SQL, client SQL
  - serveur web (Apache, nginx, flask…) qui aussi client MySql
  - serveur MySql

Le principe client / serveur est très utilisé sur internet (Ftp, mail, messagerie)

P2P : peer to peer

Il existe d'autres façons de se connecter. En **peer to peer** chacun peut-être client et serveur. Chaque machine possédant la ressource est client et client et serveur.

Lorsqu'on télécharge un torrent, on partage aussi ce qu'on a déjà téléchargé aux autres clients.

Le p2p permet de partager facilement et massivement de gros programmes comme des distributions linux. Cela évite de saturer les serveurs ftp.

On utilise aussi le P2P dans des messageries sécurisées et les cryptomonnaies.

Couche applicative

Une machine doit pouvoir se connecter à plusieurs machines en même temps et l'adresse IP ne suffit pas. On utilise le **port**. C'est l'adresse de l'application sur une machine. Il y a $2^{16} = 65536$ ports possibles.

Certains services utilisent principalement certains ports :

- 80 pour le web http
- 443 pour le web sécurisé https
- 22 pour ssh (secured shell)
- 21 pour FTP (file transfer protocol) etc.
- 25 mail
- 53 dns
- 6112 jeux Blizzard

Port en écoute.

Par défaut, tous les ports sont fermés, cela signifie qu'il est impossible de communiquer avec cette machine.

Quand on lance un serveur voulant communiquer avec l'extérieur, il ouvre un port bien particulier et l'écoute. On peut alors y accéder.

Par exemple notre serveur web B écoute sur le port 80

Un client A de ce serveur devra ouvrir lui aussi un port pour recevoir l'information de B. Il ouvre alors un port (aléatoirement au delà de 1024) et demande à B de lui répondre sur ce port.

TCP et UDP les principaux protocoles de la couche applicative

TCP : lent mais très fiable. Chaque paquet est émis coûte que coûte.

Web, email, fichiers etc. sont transmis en TCP

Si un paquet est perdu, il est renvoyé.

protocole connecté.

UDP : rapide mais peu fiable. On continue d'émettre même en cas de perte.

Streaming, certains jeux vidéos, DNS (Domain Name Service), SNMP (Simple Network Management Protocol)

On ne s'intéresse même pas à la perte de paquets.

protocole non-connecté

UDP

UDP est le protocole le plus simple. On se contente du minimum.

```
Datagramme UDP : 8 octets pour l'entête
_____________________________________________________
| Port SRC | Port DST | longueur | Checksum | Données |
|__________|__________|__________|__________|_________|
```

TCP : tout envoi sera acquitté

Avant de communiquer, on assure la communication.

Comme au téléphone :

- *Patrick appelle Raoul* : Tut… Tut…
- *Raoul répond* : Allo ?
- *Patrick commence sa conversation* : Allo Raoul ? C'est Patrick ! J'ai fait du cassoulet !

Mais aussi un peu comme la tchatche :

- Jean-Killian : *tu veux bien parler avec moi ?*
- Marie-Jennifer : *oui je suis OK*
- Jean-Killian : *bien reçu, on sort ensemble*

Les drapeaux

On utilise des paquets vides dont l'entête contient une information sur l'état de la communication : des drapeaux (**flags** en anglais).

Même principe que de dire *hmm hmm* dans le téléphone pour spécifier qu'on écoute bien.

TCP : Établir une communication

La communication TCP établie toujours 2 connexions (aller / retour)

- Client : tu veux parler avec moi ? flag SYN
- Serveur : oui, et toi ? flags SYN et ACK
- Client : oui, flag ACK

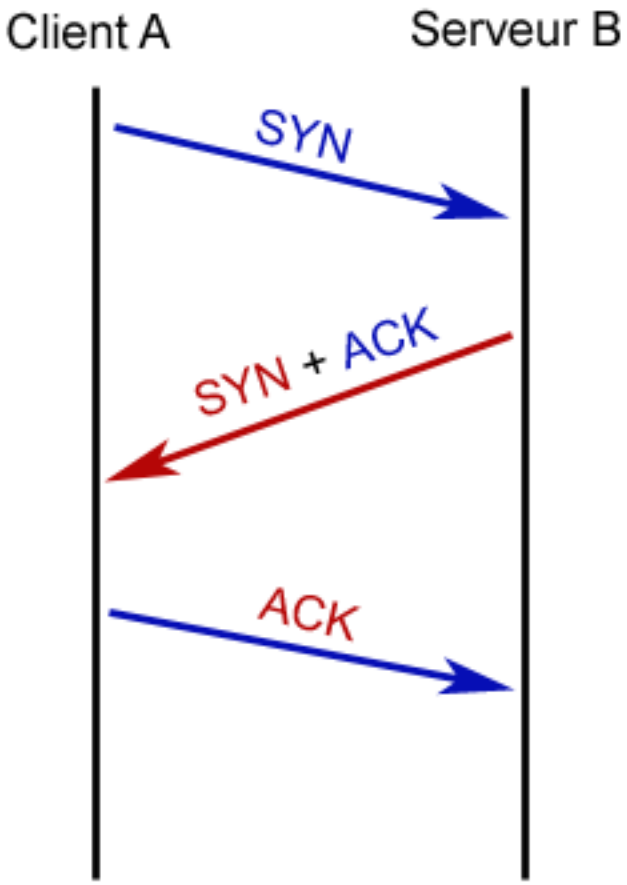


Figure 14: 3 Way Handshake

Continuité de la connexion

Tous les paquets suivants contiennent encore le flag ACK pour acquitter la réception des précédents.

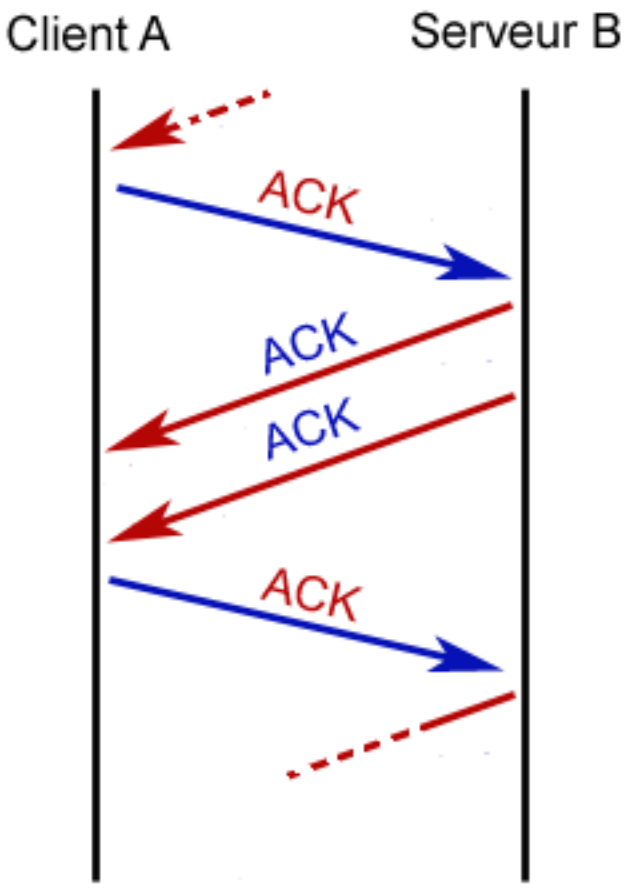


Figure 15: tcp Continuité

Fin de la connexion

Pour clore une connexion on utilise un autre flag : FIN.

- Si le client veut fermer la connexion, il acquitte aussi le précédent paquet. Il place FIN et ACK
- Le serveur répond par FIN et ACK.
- Et le client répond par ACK.

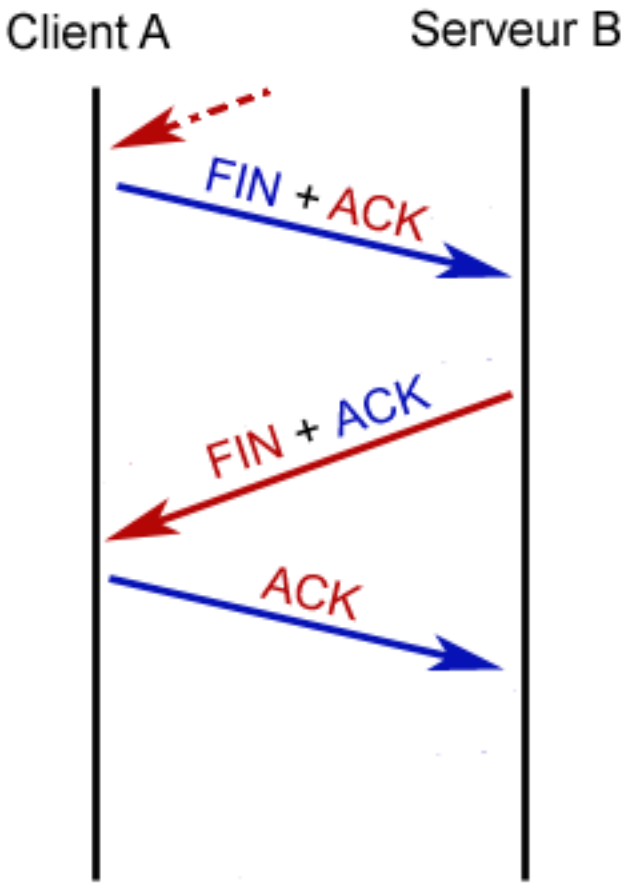


Figure 16: TCP Terminaison de la connexion

Segment TCP

```
Segment TCP : 20 octets d'entête

_____________________________________________
| Port | Port | ?? | Flags | Check | ?? | Données |
| SRC  | DST  |    |       | sum   |    |         |
|______|______|____|_______|_______|____|_________|
```

Il existe 6 flags :

- SYN
- ACK
- FIN
- RST
- PSH
- URG

Le flag RST

- Chaque octet envoyé doit être acquitté.
- S'il y a une incohérence, la machine qui s'en aperçoit le fait savoir avec le flag RST afin de clore la connexion.
- **De la même manière, lorsqu'on sollicite un port fermé, une machine répond par RST**
- PSH et URG indiquent un paquet prioritaire.

la NAT

Plusieurs machines derrière une adresse publique : la NAT

Une machine privée d'un réseau local n'a pas d'adresse propre sur internet.

C'est son routeur qui lui transmet les réponses à ses requêtes. Il utilise la NAT pour transférer à la bonne machine.

C'est la solution adoptée pour lutter contre la pénurie d'adresse IPv4.

Mais comment faire ? Le routeur attribue, pour chaque ouverture de connexion d'une machine locale un port et il transfère ce qu'il reçoit sur ce port à la bonne machine.

```
Le port est indiqué par :xx

Client A              Routeur B             Serveur Web C
connexion à C :30
                      connexion à C:3453
                                            réponse :3453
                      réponse reçue :3453
                      transmise sur :30
réponse :30
```

Port Forwarding

La NAT empêche de joindre une machine d'un réseau local

Il est possible de sortir d'un serveur mais impossible de le rejoindre de l'extérieur !

Port Forwarding

Le principe est de spécifier au routeur que **certains ports sont destinés à certaines machines.**

S'il reçoit une requête sur ce port, il la transférera sur celui de la machine.

Intérêt : **on ne rend accessible que ce qui est nécessaire.**

DHCP

DHCP : Dynamic Host Configuration Protocol

Quand le réseau est vaste, attribuer une adresse IP convenable à une machine peut-être délicat.

On utilise alors un serveur DHCP qui réalise cette tâche automatiquement.

Il peut réserver certaines adresses IP à des machines selon leur adresse MAC

On doit alors configurer sa carte réseau en mode DHCP

Rq : quand la configuration du réseau échoue, un client configuré en DHCP s'attribuera tout seul une adresse reconnaissable en 169.x.x.x

DNS

Le service DNS attribue un nom de domaine à certaines IP

Pour joindre le webmail de google, on écrit *mail.google.com* dans une barre de recherche.

mail.google.com est traduite en une adresse par un serveur DNS (Domain Name System)

Les adresses sont classées à l'aide d'un arbre :

- **.com** : top domain (comme .fr, .org, .eu …)
- **google.com** : sous domaine
- **mail.google.com** : machine hébergeant le webmail (en fait *ensemble* de machines)

L'arbre DNS

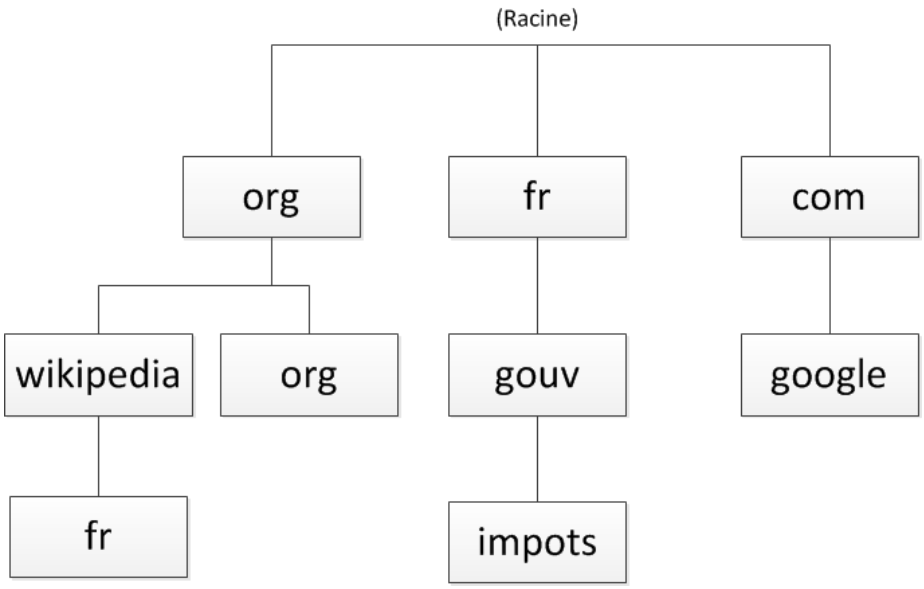


Figure 17: Domain Name System

Quelques remarques sur DNS

- Je m'attends à trouver ma messagerie quand je tape mail.google.com, pas les recettes de cassoulet de Patrick. Comment s'en assurer ?

On a crée des serveurs DNS qui se synchronisent et conservent une copie des tables de domaines.

- Les fournisseurs d'accès proposent leur serveur DNS à leurs usagers. Ils peuvent donc couper l'accès à un service en empêchant la résolution du domaine.
- Google (mais d'autres aussi) propose un serveur DNS à l'adresse IP 8.8.8.8

  Rien n'empêche de préférer un DNS plutôt qu'un autre.

DNS Injection

- Certaines attaques sur internet reposent sur l'usurpation de nom de domaine.
- Ce sont des attaques *monumentales* consistant à envoyer énormément de requêtes aux serveurs DNS pour qu'ils modifient leurs tables. .
- Elles utilisent généralement des *BotNet* : ensemble de machines piratées (caméra IP non sécurisées, par exemple) qui agissent de concert.
- Le trafic destiné à un site est alors dirigé vers celui du pirate généralement pour dérober des informations (n° de cartes bancaires, clés sécurisées etc.)

Le service Web

Web : le fondement d'internet

- Le web s'appuie sur **le protocole HTTP**
- De très nombreux serveurs, les plus utilisés sont **Apache, Nginx** et **ILLS** (Microsoft)
- La communication s'effectue à l'aide de requêtes comme **GET** ou **POST**
- On reçoit alors du code **HTML** mais aussi CSS, JS, XML, JSON, des images, des données etc.