

# Réseaux et algorithmes de routage

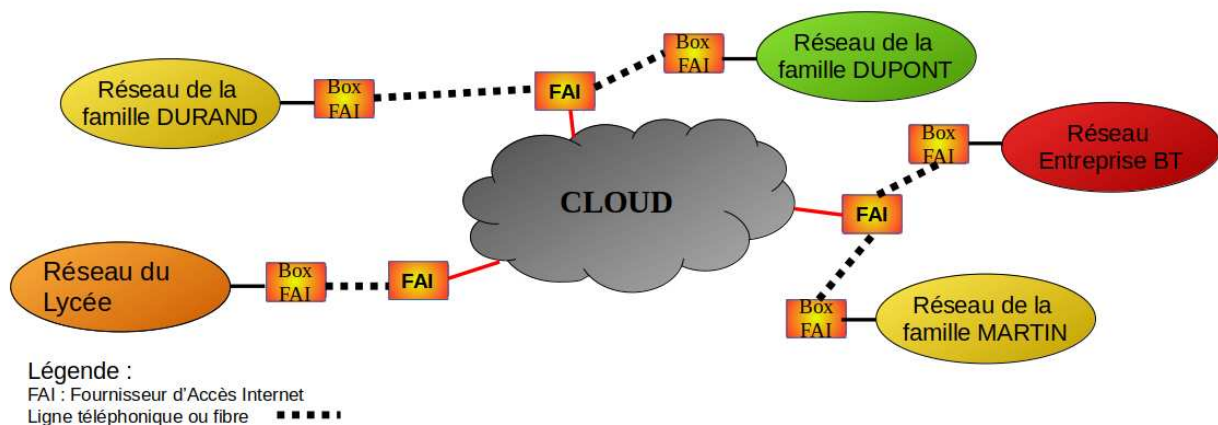
## 4.1 Rappels

### a. les différents type de réseaux

On classe les réseaux suivant leur proximité :

- **LAN (Local Area Network)** : réseau local, ensemble d'éléments reliés entre eux dans un même lieu géographique
- **WLAN (Wireless Local Area Network)** : version sans fil du LAN
- **WAN (Wide Area Network)** : réseau étendu, interconnecte plusieurs réseaux LAN éloignés géographiquement (le plus connu des réseaux WAN est Internet)

Les réseaux privées sont connectés entre eux par un WAN, représenté ci-dessous par le nuage.



Il est possible, à partir du réseau de la « famille Durand » de se connecter au réseau de l'« Entreprise BT ». Pour cela, la trame expédiée passera dans le « CLOUD » et sera acheminée par des routes qui seront sélectionnés selon des critères précis.

C'est ce que l'on appelle le « routage », notion qui sera développée dans les paragraphes suivants.

### b. L'adressage

Pour pouvoir communiquer entre eux, les différents objets le constituant doivent suivre des protocoles (modèle par couches, encapsulation des données).

Pour identifier un objet sur un réseau, on utilise une adresse IP. Dans ce cours seul le système d'adressage **IP v4** sera utilisée et présentée. Une adresse IPv4 est codée sur 32 bits et elle est souvent présentée par paquet de 4 octets notés en décimal  $a.b.c.d$  avec

$$0 \leq a, b, c, d \leq 255$$

Dans une adresse IP, une partie de l'adresse identifie le réseau et l'autre partie l'objet (une machine au sens large) sur ce réseau. Pour déterminer à partir d'une adresse, quelle est l'adresse du réseau et quelle est l'adresse de la machine dans ce réseau on utilise la notion de **masque de sous réseau**.

- **Masque de sous-réseau** : détermine l'adresse du réseau à partir de l'adresse de l'objet exemple de masque : 255.255.255.0, l'adresse du réseau est  $a.b.c.0$
- **Passerelle** : adresse IP du routeur
- **Serveur DNS (Domain Name Server)** est un serveur qui fait le lien entre une adresse avec nom de domaine et une adresse IP.

#### \* Adressage par classe

Il existe 5 classes, notées de A à E, dont les spécifications sont données à titre d'information :

**classe A** identifiant réseau codé sur le 1er octet ; ses valeurs vont de 1 à 126 et l'adresse IP aura la structure suivante :  $[1-126].x.y.z$  où  $x.y.z$  est l'adresse de la machine dans le réseau plus de 16 millions ( $2^{24}$  de machines par réseau) et 126 réseaux

**classe B** identifiant réseau codé sur 2 octets; ses valeurs vont de 128.0 à 191.255 et l'adresse IP aura la structure : [128-191].[0-255].y.z où y.z est l'adresse de la machine plus de 65 000 machines par réseau  $[(256*256) - 2 = 65\,534]$ , plus de 16 000 réseaux (16 383)

**classe C** identifiant réseau codé sur 3 octets; ses valeurs vont de 192.0.0 à 223.255.255; la structure de l'adresse IP est : [192-223].[0-255].[0-255].z 254 machines par réseau, plus de 2 000 000 réseaux  $[32*256*256 = 2\,097\,152]$

Les types D et E sont relativement particuliers ne seront pas traités ici.

Les réseaux de classe A, B et C utilisent par défaut **un masque prédéfini** :

- pour la classe A 255.0.0.0
- pour la classe B 255.255.0.0
- pour la classe C 255.255.255.0

L'adresse du réseau s'obtient en faisant un ET logique entre l'adresse IP et l'adresse du masque. Tandis que l'adresse de la machine dans le réseau s'obtient en faisant ET entre l'adresse IP et le complément à 1 du masque.

**Exemple :**

	Notation décimale	Notation binaire
Adresse IP	192.168.25.3	1100 0000. 1010 1000.0001 1001. 0000 0011
Masque	255.255.255.0	1111 1111. 1111 1111.1111 1111. 0000 0000

C'est l'occasion de revoir les opérations logiques en Python :

```
>>> 0b11000000 & 0b11111111
192
>>> 0b10101000 & 0b11111111
168
>>> 0b00011001 & 0b11111111
25
>>> 0b000000011 & 0b00000000
0
```

L'adresse du réseau est donc 192.168.25.0 pour obtenir l'adresse de la machine dans ce réseau est donc 0.0.0.3

## \* Adressage CIDR

Confrontés à la pénurie prévisible d'adresses, les organismes chargés d'allouer les adresses ont finalement abandonné le découpage en classes pour le « CIDR » Classless Inter Domain Routing qui a pour but :

1. d'alléger les tables de routage sur les routeurs
2. d'optimiser l'allocation d'adresses et d'éviter le gaspillage induit par les classes

Dans ce système on donne un réseau se voit attribué une adresse et un lot d'adresses calculer au plus juste.

**Exemple :** je souhaite adresser 52 machines. Étant donné que  $32 < 52 < 64$  il suffit d'utiliser 6 bits ( $2^6 = 64$ ) sur les 32 bits pour pouvoir adresser toutes les machines de ce réseau.

En notation CIDR on note ainsi 204.21.10.0/26 ce qui signifie que :

- l'adresse du réseau est 204.21.10.0
- le masque est constitué de 26 bits à 1 et 6 à 0 1111 1111.1111 1111.1111 1111.1100 0000  
soit 255. 255.255.192
- les adresses IP disponibles de ce réseau sont au nombre de 64 de 204.21.10.0 à 204.21.10.63, seules  $64 - 2 = 62$  sont adressables.

**N.B :** dans le cas d'un adressage dynamique un serveur DHCP affectera à une machine une adresse IP et lui transmettra toutes les informations nécessaires (masque, adresse de la passerelle ...).

Pour un adressage statique toutes ces informations doivent configurer manuellement machine par machine.

---

## 4.2 Routage

Des machines d'un même réseau sont directement reliées le plus souvent par un switch ou commutateur. Elles peuvent communiquer directement.

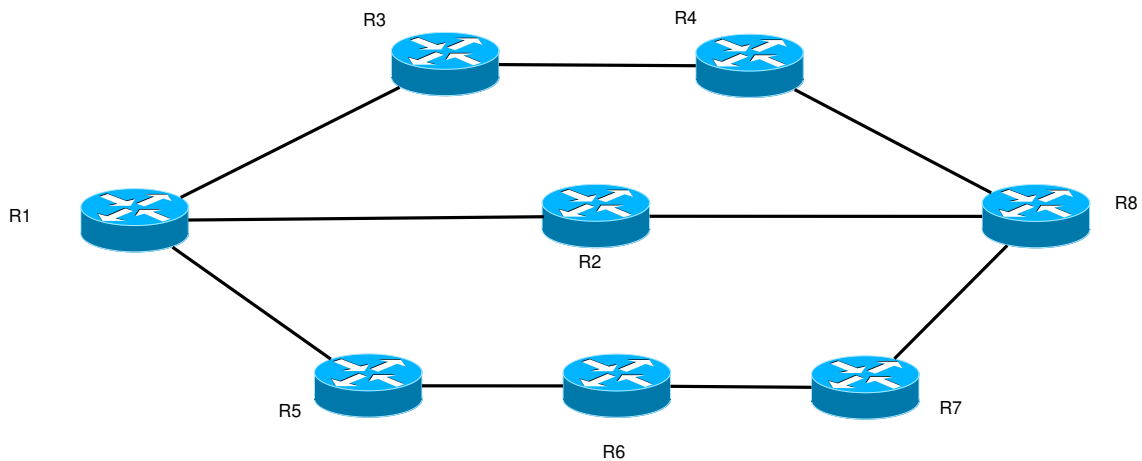
Par contre lorsqu'une machine veut communiquer avec une machine en dehors de son réseau, elle emprunte obligatoirement une **passerelle** ou **gateway** pour interroger le **routeur**.

Ce sont les routeurs qui permettent de passer d'un réseau à l'autre. Chaque routeur possède une table de routage qu'il utilise pour acheminer les paquets à la bonne destination.

Pour cela il existe plusieurs méthodes, qui reposent sur des algorithmes sur des graphes, permettant de déterminer une route.

Chacun de ses algorithmes dispose de sa propre métrique (critère choisi pour mesurer le coût de chaque route) afin de choisir parmi toutes les routes possibles la plus optimale.

### a. RIP



### b. OSPF