

HAI404I : Licence 2 Informatique

## Réseaux : IP, Protocoles et Communications

---

Anne-Elisabeth Baert – [baert@lirmm.fr](mailto:baert@lirmm.fr)

2021-2022





## Chapitre 3 : Des noms et des adresses

---



## Caractéristiques d'Internet

---

# Caractéristiques d'Internet

- Un ensemble de réseaux physiques disparates, interconnectés
- utilisant un ensemble de protocoles commun, regroupés dans l'appellation TCP/IP
- services connus : messagerie, transfert de fichiers, connexion à distance, serveurs de noms, partage de ressources.
- propositions et normes de facto : RFC (Request For Comment)

Les RFC sont des documents de référence, parfois bien lisibles, parfois non, avec un index riche, contenant un marquage d'obsolescence. **Recommandation** : Consulter au moins les plus connus, protocoles de messagerie, ceux de la toile, des protocoles communs de l'Internet.

## Présentation du problème

---

# Identifier pour s'y retrouver ...

## Le problème des adresses

On a un ensemble de machines qui ont des identifiants (différents ou pas) et on veut qu'elle communique entre elles. Il faut donc identifier ces machines intelligemment par rapport au réseau (en général) et par rapport à sa carte réseau en particulier.

## De l'ordre avec le nom de domaine

Nommage hiérarchisé par domaines : on a un domaine racine (=domaine de premier niveau) , puis un sous-domaines ( = de second niveau) , etc, jusqu'à l'hôte.

## Représentation

nom-hôte . sous-dom . .... domaine . dom-racine

Exemple : courses.carrefour.fr



Un nom, c'est bien, une adresse, c'est mieux ....

---

# Il faut nommer les choses

## Problème

Les noms sont un bon moyen pour désigner les hôtes et un très mauvais moyen pour acheminer des paquets.

On va associer une adresse aux hôtes : un **entier**. Dans la version 4 du protocole IP (version en cours) c'est un entier de **32** bits. Cette version s'appelle IPV4.

## Adresse/Paquets

Cet entier sera l'adresse de l'hôte et figurera ainsi dans tous les paquets de la couche IP.

## Historique : l'adressage par classes

---

# Adressage par classes sur IP

## Historique

Dans la préhistoire de l'Internet (1980-1990), les adresses étaient attribuées par classe, selon l'importance du réseau à administrer. Retour aux classes de services grâce à IPV6. - représente un bit affecté pour l'adresse du réseau ;

x représente un bit affecté pour l'adresse de l'hôte.

classe	octet1	octet2	octet3	octet4
A	0--- ----	xxxx xxxx	xxxx xxxx	xxxx xxxx
B	10-- ----	---- ----	xxxx xxxx	xxxx xxxx
C	110- ----	---- ----	---- ----	xxxx xxxx
D	1110	multiadressage futur !!!!!		
E	1111			

# Adressage par classes

Mais aujourd'hui : adressage sans classes sur IPV4. On y remédie en utilisant les masques le sous-adressage et le sur-adressage.

## Exercice

Combien de réseaux respectivement de classe A,B,C sont possibles dans ce monde ?

Combien d'hôtes sont possible dans chacun des cas ?

Calculer les bornes dans chaque cas.



# Adresses spécifiques

## Adresse du réseau

Lorsque la partie allouée à l'hôte est toute à zéro, cette adresse le réseau. Elle sert dans les algorithmes de routage (voir ci-après).

## Adresse tous

**Tous** dans un réseau : lorsque la partie allouée à l'hôte est toute à 1 binaire, cette adresse désigne **tous** les hôtes du réseau. Elle sert lorsqu'on veut expédier un paquet à l'ensemble des hôtes.

## Adresse du réseau

- 198.211.18.47 désigne un hôte déterminé,
- 198.211.18.0 désigne l'adresse du réseau et cette adresse n'est utilisée que dans l'algorithme de routage ; elle ne figurera jamais dans un paquet,
- 198.211.18.255 désigne **tous** les hôtes du réseau ci-dessus ; elle peut figurer dans un paquet.

## Remarque :

on verra que les adresses réseau et tous n'ont pas forcément les suffixes respectifs 0 et 255.

Adressage CIDR

---

# Adressage CIDR

## Classless Inter-Domain Routing

L'adressage sans classes=adressage CIDR est un système de gestion et d'allocation d'adresses IP le plus utilisé aujourd'hui. Ce système a été conçu pour remplacer l'adressage par classes ( RFC 1518 et 1519).

## Définition

Une adresse IP avec CIDR ressemble à une adresse IP normale, à la différence près qu'elle se termine par une barre oblique suivie d'un nombre, appelé préfixe du réseau IP.

Les adresses CIDR réduisent la taille des tables de routage et rendent plus d'adresses IP disponibles au sein des organisations.

Et les adresse Ips, c'est quoi ?

---



# Protocole IP

## Définition

On appelle **datagramme** un paquet vu de la couche réseau.

## Ce que garantit IP :

- ☐ Acheminement de datagrammes sans connexion ;
- ☐ décision selon l'adresse réseau du destinataire ;
- ☐ décision à chaque datagramme indépendamment du passé ;
- ☐ redécoupage possible ;
- ☐ boucles possibles ;
- ☐ acheminement **au mieux** (best effort), donc pas de garantie de livraison : un paquet peut être perdu, supprimé...

# Format du Paquet IP

octet 1		octet 2		octet 3		octet4	
Vers.	lg. ent.	type service		lg. paquet			
Identification				drapeaux		place frag.	
durée vie		proto. suiv.		contrôle entête			
adresse IP source							
adresse IP destination							
options ...							
...				bourrage			
Données							
...							

L'entête classique, sans options, fait 20 octets.

Des adresses IP pour le routage

---

# Routage - Généralités

## Le routage

Les routeurs font du routage selon l'adresse du **réseau** destinataire (et non l'hôte destinataire).

## Les algorithmes

Algorithmes plus ou moins sophistiqués (parcours dans un graphe dynamique).

## Acheminement

Tous les hôtes impliqués dans l'acheminement doivent résoudre le problème du routage pour chaque paquet à expédier : à qui envoyer ce paquet ? Ici, tous les hôtes impliqués sont l'hôte expéditeur et tous les routeurs intermédiaires, jusqu'au dernier routeur localisé sur le même réseau que l'hôte destinataire.

## Remarques :

- Ne pas oublier que chaque acheminement hors du réseau local implique un acheminement local (le routeur local).
- La commande `netstat -r` permet de visualiser la table de routage.
- Cette table peut être statique ou dynamique (cf. chapitre Routage).



# Table de Routage - Exemple

## Tables de routage

Un hôte dans un réseau local de technologie ethernet, avec un seul routeur vers le monde extérieur aura une table de routage de cette forme :

Destination	Contact	Interface
201.202.203.0	direct	eth0
autre	201.202.203.1	eth1

≡ Passerelle  
dans l'affichage des  
table.

## Réseau local

Pour contacter tout hôte du réseau local, pas besoin d'un intermédiaire ; on envoie les paquets directement au destinataire, en les expédiant sur la carte réseau dont l'adresse est *eth0*.

## Autre réseau

Pour contacter tout autre hôte, expédier le paquet vers la machine dont l'adresse réseau est 201.202.203.1, toujours par la carte réseau d'adresse *eth0*.

# Des Problèmes en perspective

## Les problèmes

La table de routage donne l'adresse réseau du contact. Or l'adresse du destinataire dans le paquet de la couche réseau **doit** être celle du destinataire final, de bout en bout !

Il faudra disposer de l'adresse physique du routeur, c'est-à-dire l'adresse physique correspondant à l'adresse réseau 201.202.203.1.

Même dans le cas d'un contact direct, on ne dispose que de l'adresse réseau du destinataire.

## Les solutions

Dans tous les cas, à la fin de l'algorithme de routage, on obtient comme résultat une adresse réseau (routeur ou destinataire final).

Il faudra obtenir l'adresse physique si on veut acheminer ce paquet.

## Les adresses physiques

---

# Adresses MAC

## Définition

Une adresse MAC est un identifiant de 12 chiffres hexadécimaux attribué à chaque carte réseau : c'est une adresse physique

Par convention on place des : tous les 2 chiffres

## EXEMPLE :

12 :34 :56 :78 :91 :BC

Dans un même sous réseau, la communication entre machines est possible avec des adresse MAC.

Les adresses MAC ne sont pas routables, on utilise donc les adresses IPs.



Les serveurs de noms : du nom aux adresses

---

# Serveurs de Noms

C'est une des premières applications importantes quoique invisible dans les réseaux.

## Le problème

Connaissant le nom d'un hôte, trouver son adresse. Elle est indispensable si l'on veut construire un paquet qui lui est destiné.

## Principe

Une base de données distribuée, où chaque administrateur mettra à jour les données relatives à son réseau. Il mettra en place une application appelée serveur de noms (DNS) qui répondra à chaque requête contenant un nom, par l'adresse correspondante.

# Algorithme de recherche

## Généralisation du problème

Connaissant une caractéristique d'un hôte, trouver toutes les informations enregistrées à son sujet.

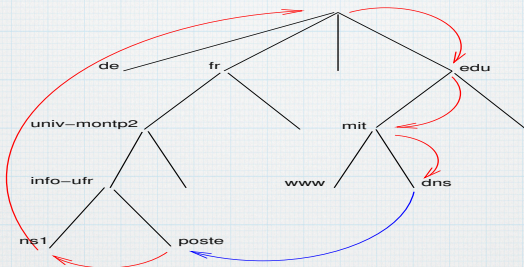
## Algorithme de recherche

Basée sur une recherche en arbre plutôt originale.

## Principe de l'algorithme

Un hôte demande à son serveur de noms local les coordonnées globales pour un nom d'hôte. Pour toute réponse concernant un hôte non local, le serveur doit disposer d'au moins une adresse d'un serveur extérieur qui donnera la réponse ou pourra faire suivre la requête.

## Serveurs de noms - Exemple



## Solution simple

Disposer d'au moins une adresse d'un serveur racine.

## Exemple simple

Un utilisateur navigant sur `g12@info-ufr.univ-montp2.fr` veut contacter `www.mit.edu`.

L'application locale va adresser la requête au serveur de noms local (**ns1** sur la figure).

**ns1** ne dispose que de la base locale (tous les noms et adresses des machines que l'administrateur veut rendre visibles à l'extérieur), et d'une adresse d'un serveur racine.



## Recherche en arbre

Le serveur racine connaît uniquement les serveurs de noms de premier niveau. Dans cet exemple, celui du domaine edu. Celui-ci connaît à son tour les serveurs de noms des domaines d'un niveau en dessous et ainsi de suite.

## Chemin inverse

Le serveur du domaine **mit** répondra, sans avoir à suivre le chemin inverse.

**Question** : pourquoi ?

L'adresse source est dans le paquet. .