

# Couche réseau (IP)

Module FAR

Polytech Montpellier – IG3

David Delahaye

# Couche réseau (IP) – Plan

1. Interconnexion IP
2. Relayage et routage IP
3. Adressage et nommage
4. Adressage IP
5. Datagramme IP
6. Fragmentation
7. ICMP

# Couche réseau (IP) – Plan

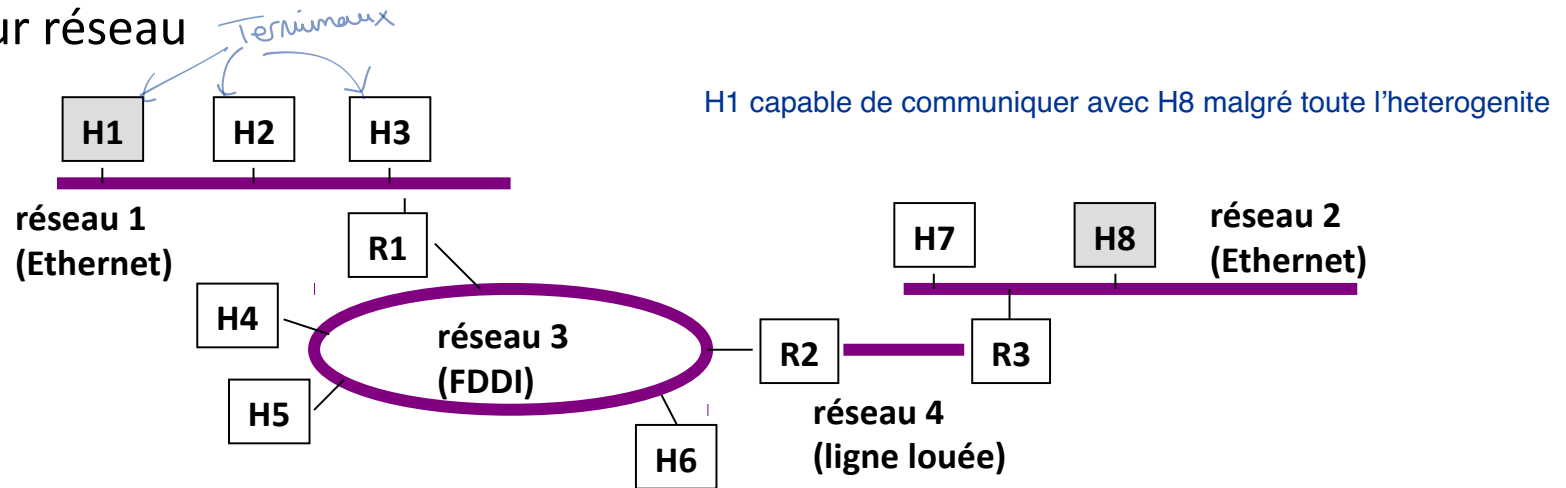
1. **Interconnexion IP**
  - ✓ concept d'interconnexion
  - ✓ rôle de IP
  - ✓ service de IP
2. Relayage et routage IP
3. Adressage et nommage
4. Adressage IP
5. Datagramme IP
6. Fragmentation
7. ICMP

# Couche réseau (IP) – Plan

1. **Interconnexion IP**
  - ✓ concept d'interconnexion
  - ✓ rôle de IP
  - ✓ service de IP
2. Relayage et routage IP
3. Adressage et nommage
4. Adressage IP
5. Datagramme IP
6. Fragmentation
7. ICMP

# Interconnexion

- une concaténation de réseaux
  - à l'intérieur d'un réseau, les nœuds utilisant la technologie spécifique de leur réseau



- l'**interconnexion** consiste à faire transiter des informations depuis une machine sur un réseau vers une autre machine sur un autre réseau
  - les différences entre tous les réseaux ne doivent pas apparaître à l'utilisateur

# Interconnexion

C:\Users\David Delahaye>tracert www.google.com Trace tous les noeuds du réseau qui sont utilisés pour se connecter à google

Détermination de l'itinéraire vers www.google.com [173.194.40.115]

avec un maximum de 30 sauts :

1	1 ms	1 ms	<1 ms	livebox.home [192.168.1.1]
2	16 ms	17 ms	17 ms	80.10.121.68
3	17 ms	18 ms	17 ms	10.123.89.10
4	24 ms	23 ms	24 ms	ae48-0.nilyo101.Lyon.francetelecom.net [193.252.101.158]
5	29 ms	29 ms	32 ms	81.253.184.78
6	32 ms	28 ms	35 ms	dns100ge16-0-0.marcr2.Marseille.opentransit.net [193.251.133.241]
7	30 ms	31 ms	30 ms	81.52.186.142
8	30 ms	30 ms	30 ms	209.85.252.194
9	35 ms	34 ms	35 ms	216.239.58.3
10	34 ms	36 ms	35 ms	209.85.245.73
11	35 ms	35 ms	35 ms	209.85.243.45
12	34 ms	35 ms	35 ms	par10s09-in-f19.1e100.net [173.194.40.115]

Itinéraire déterminé.

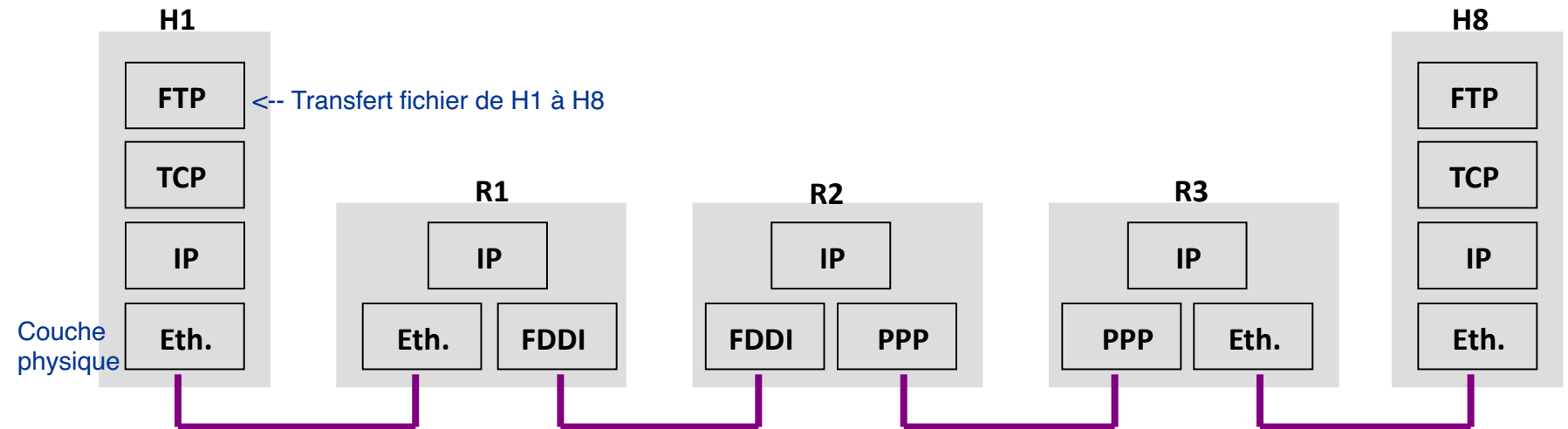
# Interconnexion

- Principe
  - mise en œuvre d'une couche réseau
    - masquant les détails de la communication physique du réseau (On route la trame peut importe ce qu'elle contient)
    - détachant les applications des problèmes de routage
- le logiciel d'interconnexion
  - fait apparaître l'ensemble des réseaux disparates comme un seul et unique réseau
  - offre un service commun à toutes les applications

# Interconnexion IP

- la glue qui lie l'Internet : le protocole **IP** (*Internet Protocol*)
- une pile de protocoles

Architecture tcpip (4 couches)

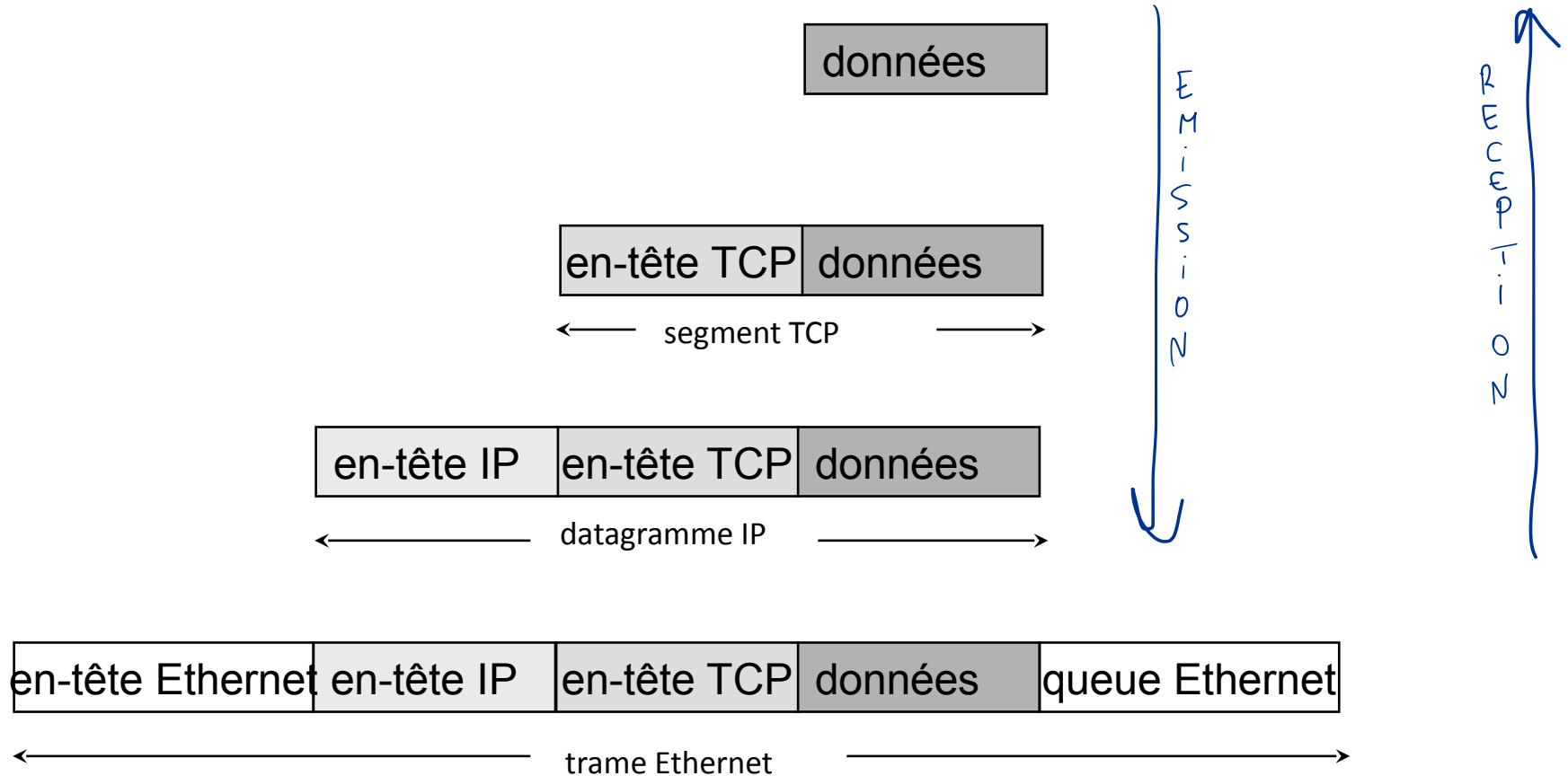


Couche IP commune à tout le monde



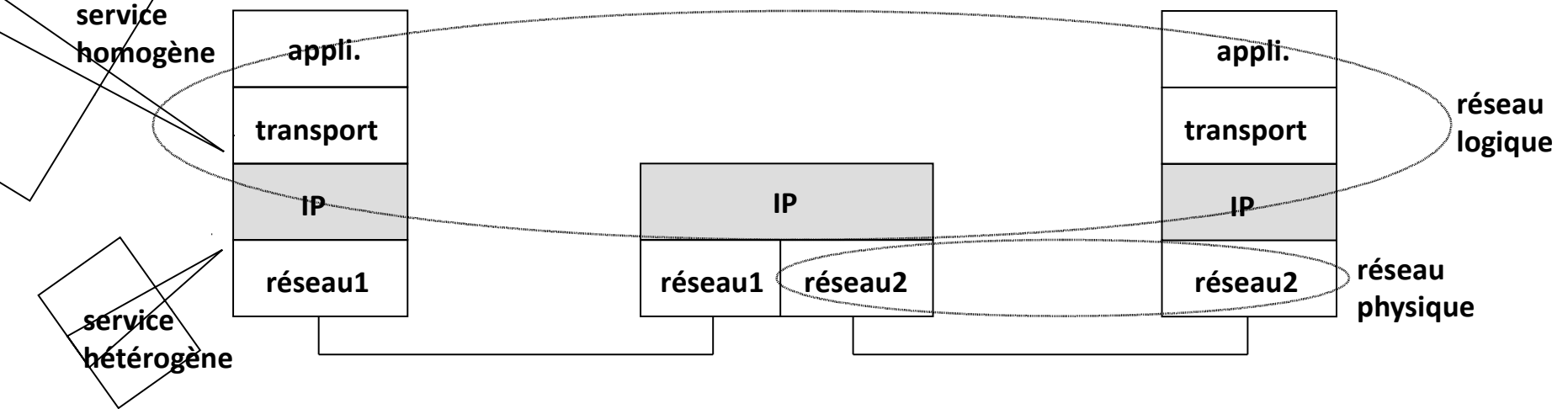
# Encapsulations successives

- cas application / TCP / IP / Ethernet



# Internet Protocol (IP)

- protocole de la couche d'interconnexion



- caractéristiques de l'interface IP
  - adresse **unique** et **homogène**
  - transfert par blocs (datagrammes)
  - service sans connexion
  - service *best effort* (les datagrammes ne sont pas éliminés sans raison)
- fonctions de IP
  - adressage (Donne des adresses)
  - routage (Cmt lordi sait par quels noeuds il faut passer pour arriver au terminal)
  - fragmentation (Recomposer le msg)

# Service IP

- utilise un service minimum
  - envoi d'une unité de transfert d'un point à ses voisins (Datagram/trame)
  - voisin : partage la même connexion physique
- rend un service minimum
  - service en mode non connecté
    - absence d'états dans les routeurs
    - transmission de datagrammes
  - remise *best effort*
    - service non fiable
  - service de connectivité
- avantages
  - ☺ robustesse
  - ☺ efficace pour les échanges brefs
  - ☺ simplicité d'utilisation

# Primitives de service IP

Pour spécifier et envoyer un msg, la fct sera définie de ce type :

**SEND** (

adresse source

adresse destination

TCP ou UDP: protocole supérieur

indicateurs de QOS: high ou normal

rapidité: de remise demandée

priorité: traitement d'abord

débit:

sécurité:

identificateur du paquet courant

indicateur de fragmentation: oui ou non

durée de vie

longueur des données

données optionnelles: sécurité (mot de passe pour une passerelle), routage/ source, enregistrement du chemin, estampillage (enregistrement de l'heure de chaque passage)

données )

) facultatif

# Primitives de service IP

## **DELIVER** (

adresse source

adresse destination

TCP ou UDP: protocole supérieur

indicateurs de QOS: high ou normal

longueur des données

données optionnelles et données )

# Couche réseau (IP) – Plan

1. Interconnexion IP
2. **Relayage et routage IP**
  - relayage IP
  - routage IP
3. Adressage et nommage
4. Adressage IP
5. Datagramme IP
6. Fragmentation
7. ICMP

# Relayage vs. routage

- **relayage** (*forwarding*) Transfert de la donnée sans la traiter
  - action de retransmettre sur une interface un datagramme reçu sur une autre interface et n'étant pas arrivé à destination
- **routage**
  - action de chercher (1) l'interface sur laquelle transmettre un datagramme, et (2) le destinataire immédiat
- interface (réseau) : point d'accès au réseau (physique)

# Qu'est ce qu'un routeur ?

Redirigé la donnée. Il lui faut donc un accès d'entrée et un accès de sortie pour pouvoir transférer les infos.

Le routeur est un matériel de couche 3 qui relie plusieurs réseaux.

Composé de 3 couches : physique , liaison de donnée, réseau

Il doit donc avoir une interface dans chacun des réseaux auquel il est connecté.

C'est donc tout simplement une machine qui a plusieurs interfaces (plusieurs cartes réseau), chacune reliée à un réseau. Son rôle va être d'aiguiller les paquets reçus entre les différents réseaux.

Un ordinateur ayant deux cartes réseau pourra être un routeur.

On essaye de prendre le chemin le plus court pour atteindre la source.



# Que fait un routeur ?

CRC = un type de checksum

Pour chaque datagramme IP qui traverse le routeur, IP :

- vérifie le checksum, si faux → destruction du datagramme
- détermine si ce sont des données *utilisateur* ou de *contrôle* destinées au routeur
- décrémente la durée de vie, si nulle → destruction du datagramme
- **décide du routage**
- **fragmente** le datagramme si nécessaire
- **reconstruit l'en-tête IP** avec les champs mis à jour
- transmet le(s) datagramme(s) au protocole d'accès de l'interface réseau de sortie avec **l'adresse de sous-réseau correspondante**

A réception dans l'hôte destinataire, IP :

- vérifie le checksum
- s'il y a eu fragmentation, mémorise puis **réassemble**
- **délivre au niveau supérieur** les données et les paramètres par la primitive DELIVER (De la couche réseau à la couche application)

# Routeage

Utilise une table de routage pour savoir quel chemin prendre, la façon dont il va pouvoir atteindre un noeud.

- fonction déterminant un chemin vers une adresse destinataire

## table de routage

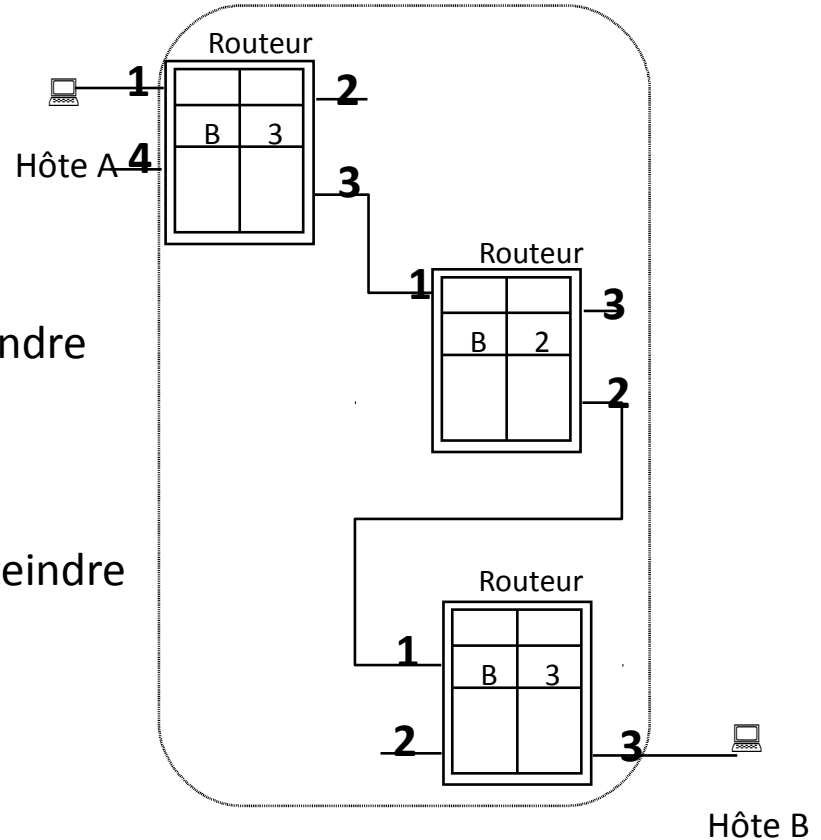
- informations nécessaires pour atteindre le prochain nœud

## algorithme de routage

- calcul d'un chemin optimal pour atteindre une adresse destinataire

## protocole de routage

- échange d'informations de routage
- dépend du domaine dans lequel se trouve le routeur
- ex : RIP, OSPF, ...



# Tables de routage

- La table de routage va donc lister les routeurs auxquels je peux envoyer mon datagramme pour joindre une destination donnée.
- La destination donnée ne va pas être une machine, mais un réseau. Si on devait indiquer un chemin pour chaque machine sur Internet, les tables de routage seraient énormes !
- Le principe est d'avoir d'un côté la liste des réseaux que l'on veut joindre, et de l'autre la liste des routeurs à qui nous devons envoyer le datagramme pour joindre les réseaux. On appelle aussi ce routeur une passerelle.

# Tables de routage

Exemple de 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

# Tables de routage

- contiennent 3 informations

- destination

- réseau
    - sous-réseau
    - machine
    - default

- chemin

- interface locale à la machine
    - un routeur intermédiaire

- coût ou *métrique*

- nombre de hops
    - débit ...

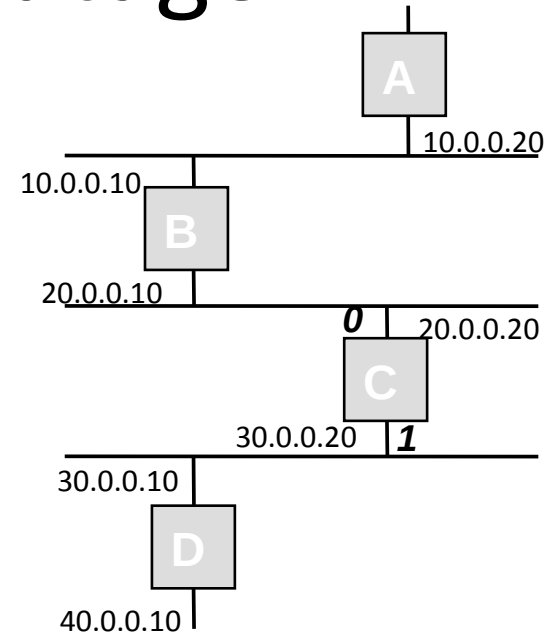
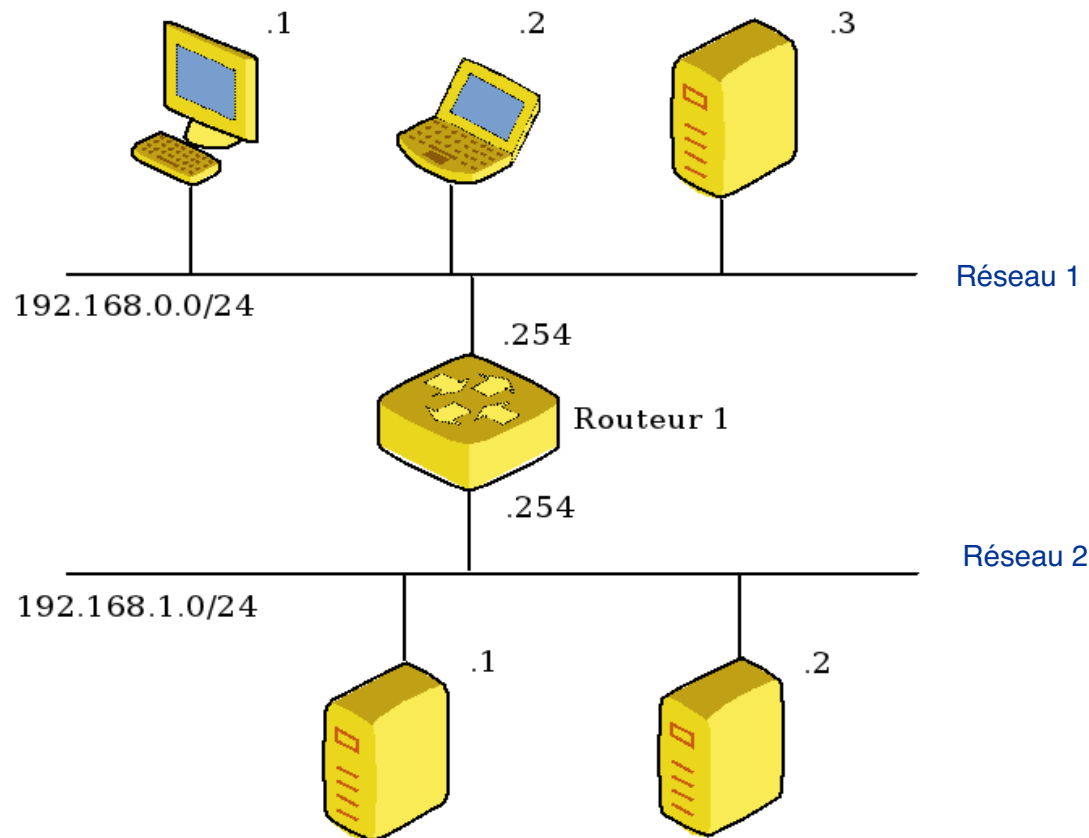


Table du nœud C

Préfixe @destinataire	@routeur par où passer
10.0.0	20.0.0.10
20.0.0	<i>eth0</i>
30.0.0	<i>eth1</i>
40.0.0	30.0.0.10

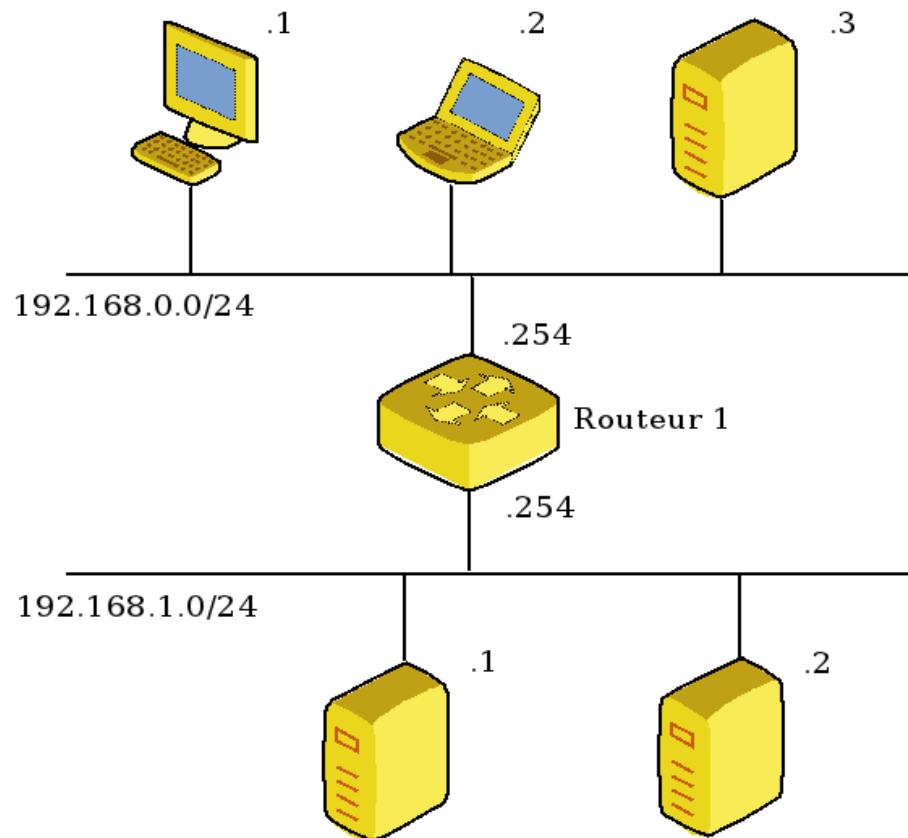
# Tables de routage (exercice)

Déterminer la table de routage du routeur 1 :



# Tables de routage (exercice)

Déterminer la table de routage de la machine 192.168.0.1 :



# Tables de routage :

Routeur 1 :

	Destination	Masque	(Pt d'interconnexion) Passerelle	Interface
	192.168.0.0	/24	192.168.0.254 <i>idem</i>	192.168.0.254
	192.168.1.0	/24	192.168.1.254 <i>idem</i>	192.168.1.254
<i>Routage direct</i>	192.168.0.0	/24	<i>idem</i>	192.168.0.1
<i>Routage indirect</i>	192.168.1.0	/24	192.168.0.254 (routeur)	192.168.0.1

Rq: si pas de passerelle, alors on a l'IP que l'interface.

Machine  
192.168.0.1



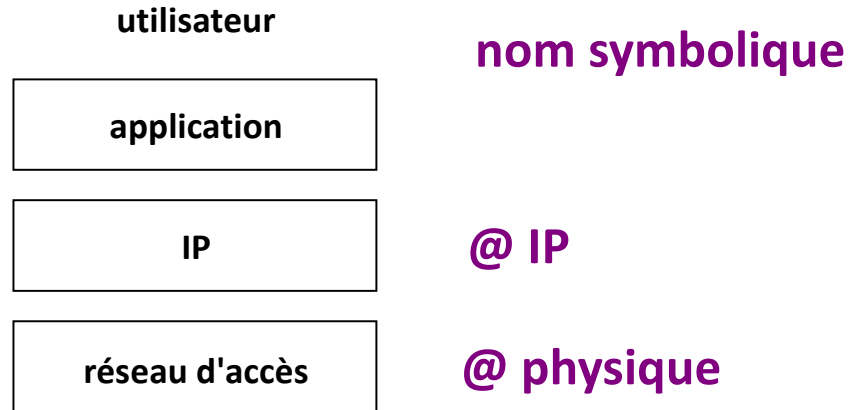
# Routage direct/indirect

- le module IP examine l'adresse de destination du datagramme et détermine si la destination est sur un même réseau physique
  - si oui : **routage direct**
    - la destination étant sur un même réseau physique, le datagramme est transmis directement
      - ↳ l'adresse physique suivante est celle de la destination
  - si non : **routage indirect**
    - la destination étant sur un réseau physique différent, le datagramme est transmis au routeur suivant
      - ↳ l'adresse physique suivante est celle du routeur suivant

# Couche réseau (IP) – Plan

1. Interconnexion IP
2. Relayage et routage IP
3. **Adressage et nommage**
  - noms et DNS
  - adresses physiques et ARP
4. Adressage IP
5. Datagramme IP
6. Fragmentation
7. ICMP

# Différentes "adresses"



↩ 2 niveaux de conversion Du nommage DNS :

- nom ⇔ adresse IP
- adresse IP ⇔ adresse physique

# Les domaines

- l'Internet est organisé en

## domaines

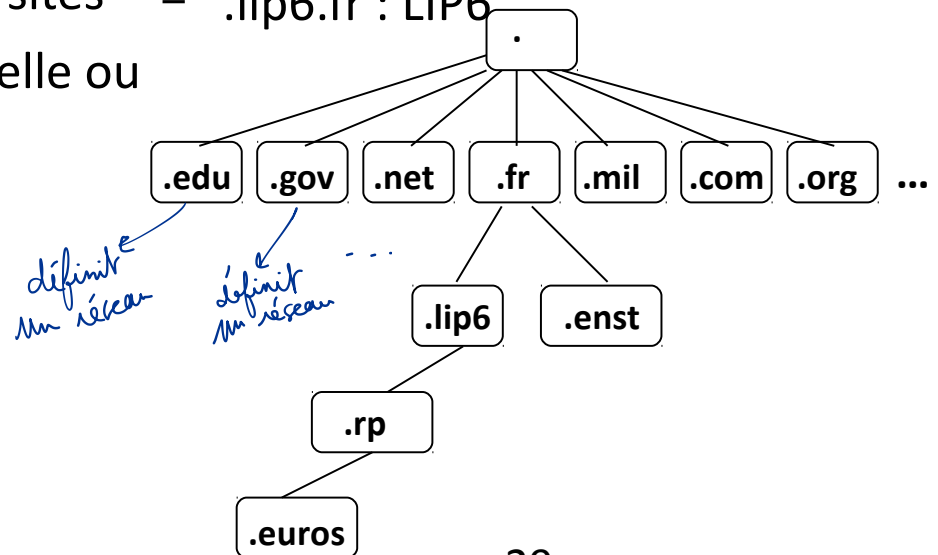
- Domaine : ensemble de réseaux administrée par une seule autorité, appelé *Administrative System*
- Domaine : regroupement de sites ayant une relation fonctionnelle ou géographique entre eux

- hiérarchie de domaines

- ex : euros.rp.lip6.fr
- racine (ICANN)

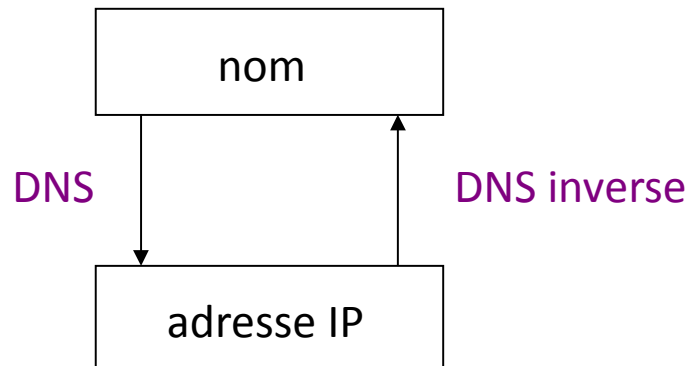
- exemples de domaines

- .gov : institutions gouvernementales américaines
- .edu : universités américaines
- .com : sites commerciaux
- .fr : sites français
- .lip6.fr : LIP6



# Nommage

- Principe
  - unicité des noms
  - gestion distribuée : nommage hiérarchique
  - plan de nommage indépendant du plan d'adressage
- besoin d'une glue : Domain Name System

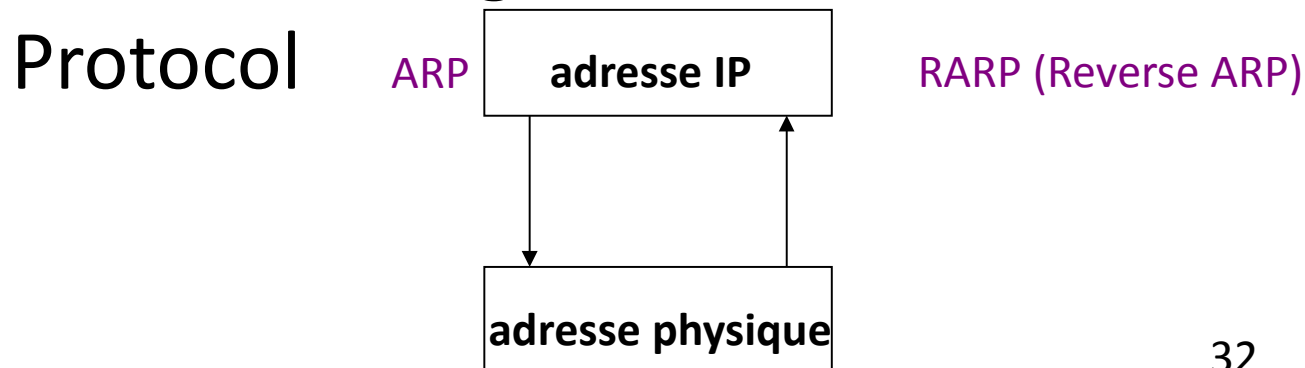


# Résolution des noms

- hiérarchie de serveurs DNS
  - découpage de la hiérarchie en zones
  - chaque zone est servie par au moins un serveur de noms
- redondance
  - un serveur primaire + des serveurs secondaires par zone
- un client effectue des requêtes à son serveur DNS
- le serveur DNS répond ou remonte la demande au serveur de niveau supérieur
- présence de caches aux différents niveaux

# Adressage physique

- Principe
  - adresse propre au système de transmission
  - identifie le périphérique
  - multitude d'espaces d'adressage
  - exemples
    - @ Ethernet (MAC-48) : sur 6 octets, adressage absolu
    - @ ATM : sur 20 octets, adressage hiérarchique
- besoin d'une glue : Address Resolution Protocol



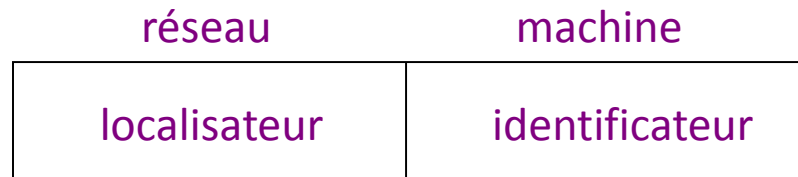
# Couche réseau (IP) – Plan

1. Interconnexion IP
2. Relayage et routage IP
3. Adressage et nommage
4. **Adressage IP**
  - classes d'adresses
  - subnetting
5. Datagramme IP
6. Fragmentation
7. ICMP



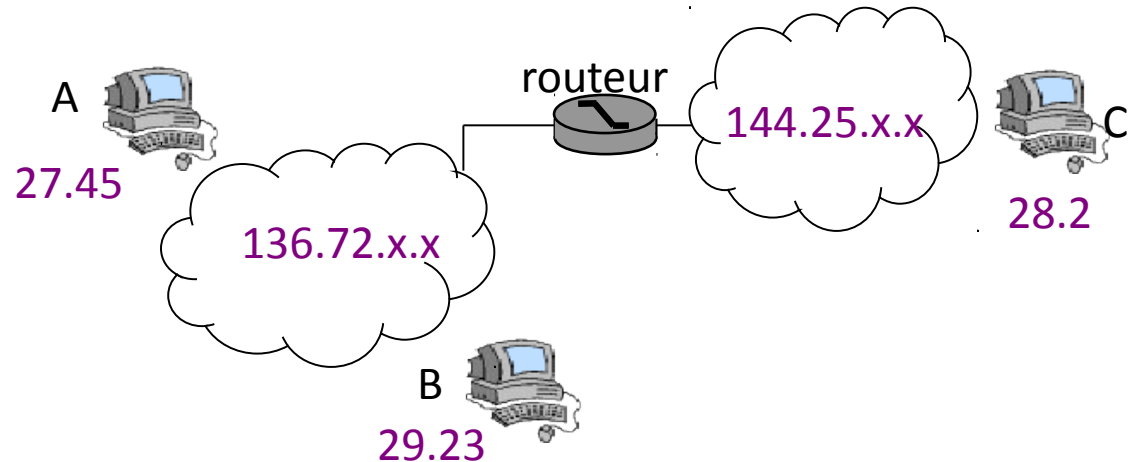
# Adressage IP

- adressage
  - pour l'identification d'un équipement réseau
  - pour le routage
- plan d'adressage homogène
  - format : 4 octets ➔ 4,3 milliards d'adresses ???
  - notation décimale pointée : x1.x2.x3.x4
- adresse globalement unique et hiérarchique
- format : <réseau> <machine>
  - localisateur ou préfixe réseau : identificateur de réseau
  - identificateur : identificateur de machine

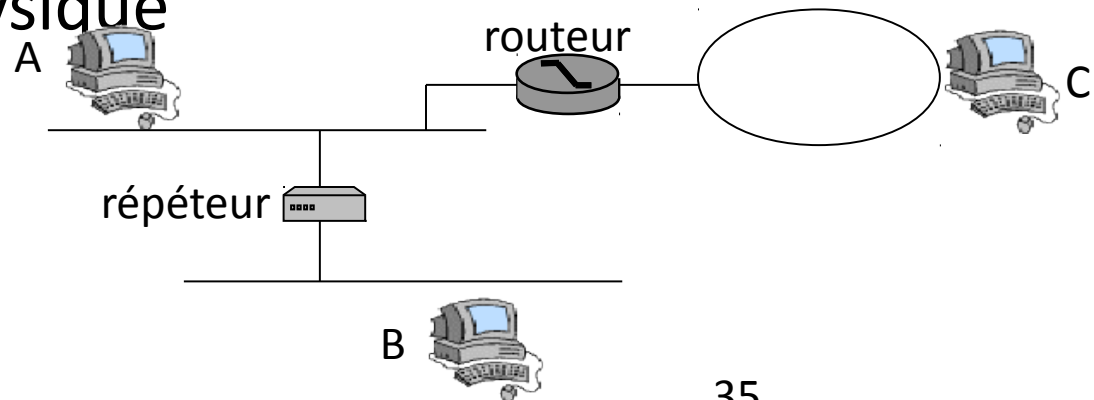


# Adressage réseau

- un préfixe réseau par réseau physique
  - au niveau IP



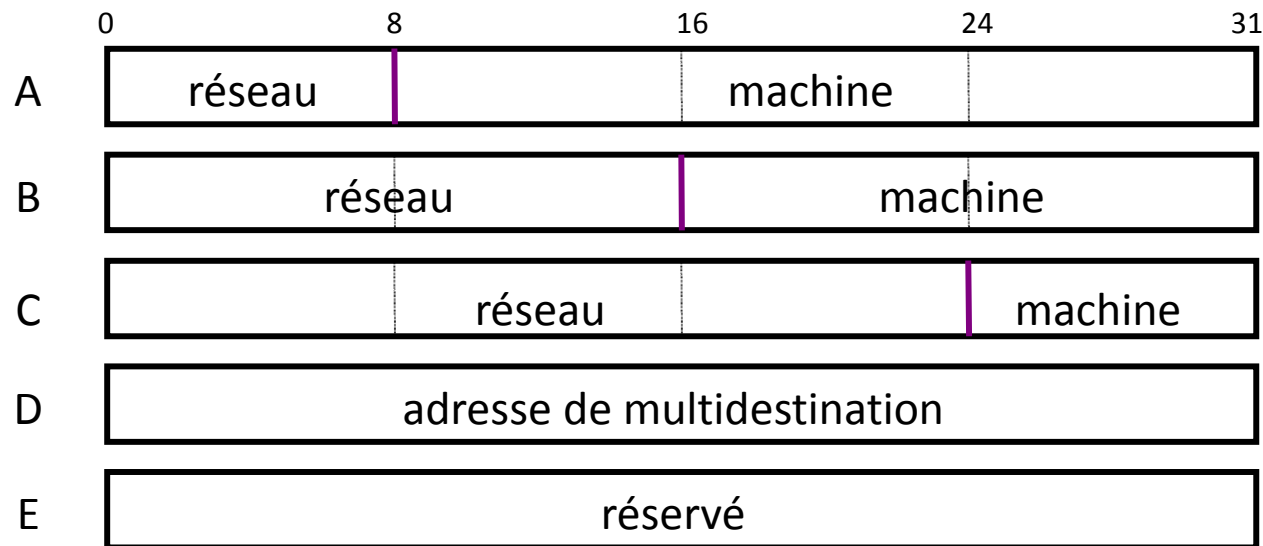
- au niveau physique



# Classes d'adresses

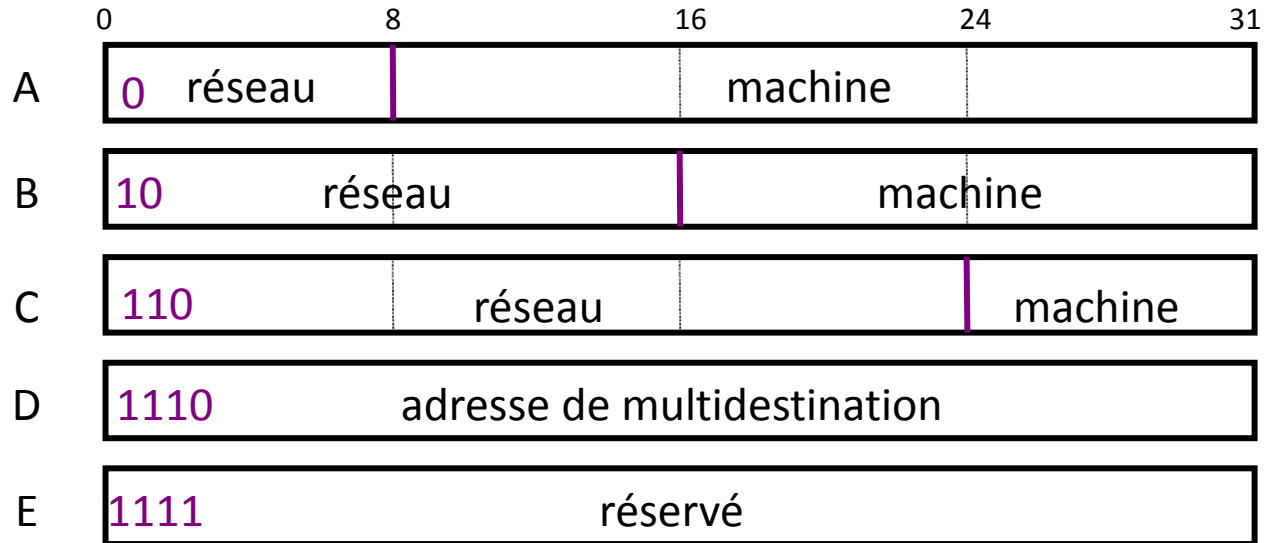
- le découpage <réseau> / <machine> n'est pas fixe

↪ 5 classes d'adresses



- comment reconnaître ces classes ?

# Classes d'adresses



# Classes d'adresses

- **classe A** : 27 réseaux (128)
  - réservé: 0.0.0.0 et 127.0.0.0
  - disponible: 1.0.0.0 à 126.0.0.0
    - 126 réseaux classe A et 16 777 214 machines/réseau
- **classe B** : 214 réseaux (16 384)
  - réservé: 128.0.0.0 et 191.255.0.0
  - disponible 128.1.0.0 à 191.254.0.0
    - 16 382 réseaux classe B et 65 534 machines/réseau
- **classe C** : 221 réseaux (2 097 152)
  - réservé 192.0.0.0 et 223.255.255.0
  - disponible 192.0.1.0 à 223.255.254.0
    - 2 097 150 réseaux classe C et 254 machines/réseau

# Classes d'adresses

- Classes obsolètes depuis le milieu des années 1990 :  
Classless Inter-Domain Routing (CIDR)
- Déterminer la partie machine et la partie réseau :
  - Utiliser le masque de sous-réseau
  - Exprimé comme une adresse IP
  - Les bits à 1 déterminent la partie réseau
  - Les bits à 0 déterminent la partie machine
- Exemple :
  - 255.255.0.0 -> 11111111.11111111.00000000.00000000
  - 192.168.0.1 -> 11000000.10101000.00000000.00000001
  - Partie réseau : 192.168
  - Partie machine : 0.1
  - Équivalent à : 192.168.0.1/16

# Classes d'adresses (exercice)

- Déterminer la partie réseau et machine des adresses suivantes :
  - 192.168.0.1 avec le masque 255.255.240.0
  - 192.168.0.1 avec le masque 255.255.173.0

# Plages d'adresses

- Déterminées en prenant toutes les combinaisons possibles de la partie machine
- Adresses particulières :
  - Adresse à 0 : adresse du réseau lui-même
  - Adresse avec que des 1 : adresse du broadcast
- Donc avec  $n$  bits pour la partie machine, on peut adresser  $2^n - 2$  machines (on enlève le réseau et le broadcast)

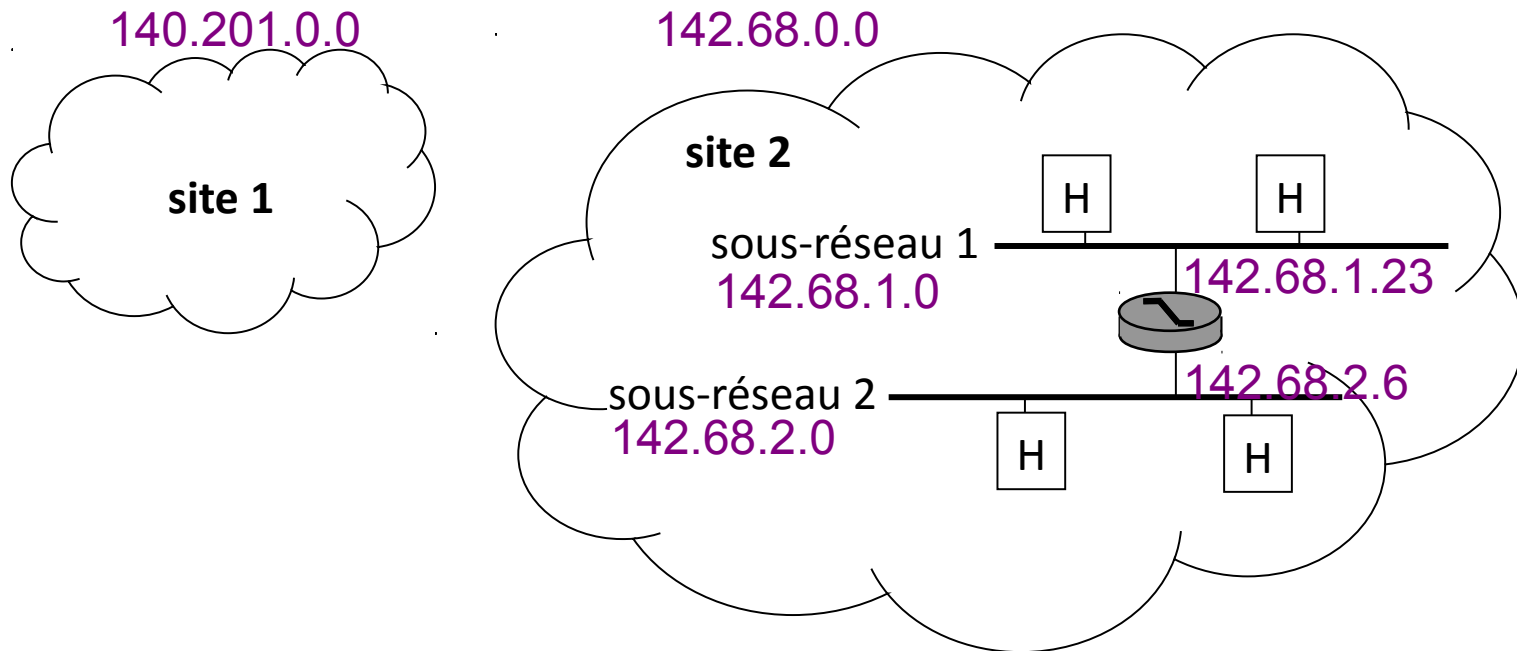


# Plages d'adresses (exercice)

- Pour les couples IP/masques suivants, donner la plage d'adresses du réseau et dire si l'adresse est l'adresse d'une machine, du réseau, ou du broadcast :
  - 192.168.0.15/255.255.255.0
  - 192.168.1.0/255.255.255.0
  - 10.8.65.31/255.255.255.224

# Subnetting

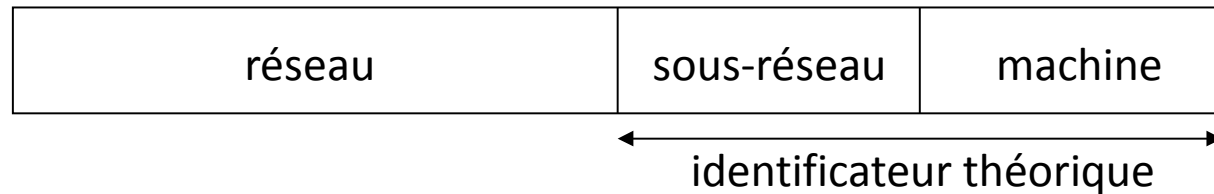
- Problème
  - distinction <réseau> / <hôte> insuffisante en pratique



# Sous-adressage

- Principe

- ajout d'un niveau hiérarchique dans l'adressage
  - adresse de sous-réseau
- subdivision de la partie <hôte>



- le sous-réseau

- est un réseau physique (i.e. un réseau IP connexe) du réseau de site
- a une visibilité purement interne

# Sous-adressage (exercice)

Une entreprise qui possède 10 succursales s'est vue affecter l'adresse IP 196.179.110.0 avec le masque de sous-réseau 255.255.255.0. Pour une gestion plus fine de ses sous-réseaux, le responsable informatique désire pouvoir affecter une adresse IP propre à chaque sous-réseau des 10 succursales.

.

1. Donner la nouvelle valeur du masque de sous-réseau correspondant à ce besoin.
2. Combien de machines chaque sous-réseau pourra-t-il comporter et pourquoi ?
3. Définir l'adresse de broadcast du sous-réseau 3.

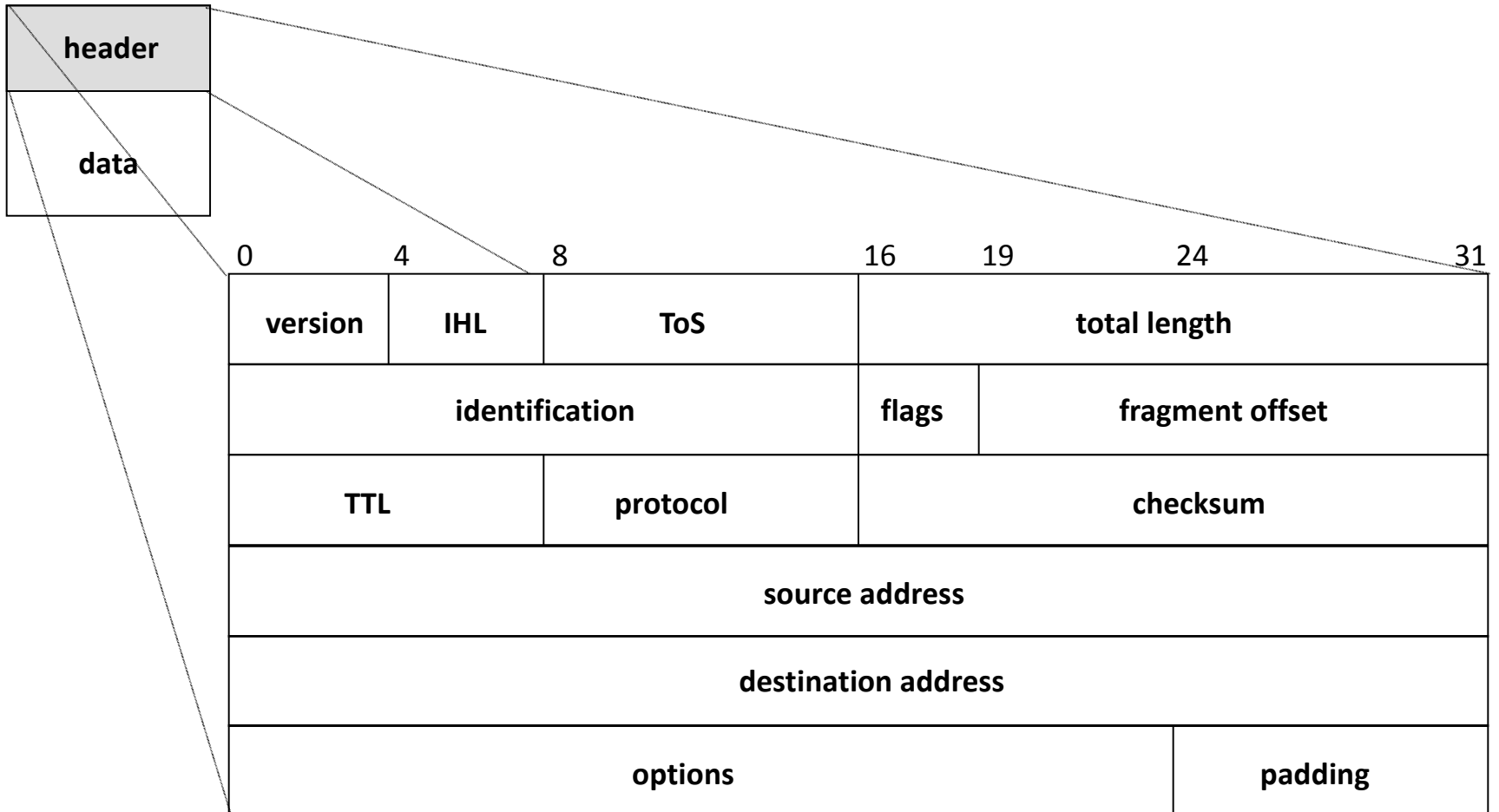
# IPv4

- IPv4 = Internet Protocol version 4
- $2^{32}$  adresses = 4 294 967 296
- Arrivé à saturation le 3 février 2011
- Solutions :
  - IPv6 : sur 128 bits, mais en 2015 seuls 6% des utilisateurs en IPv6 !
  - Network Address Translation (NAT)

# Couche réseau (IP) – Plan

1. Interconnexion IP
2. Relayage et routage IP
3. Adressage et nommage
4. Adressage IP
5. **Datagramme IP**
6. Fragmentation
7. ICMP

# Le datagramme IP



# Les champs de l'en-tête IP

- **version** : identification de la version courante du protocole (4 pour IPv4)
- **IHL** (*IP Header Length*) : longueur de l'en-tête IP (en mots de 32 bits)
- **TOS** (*Type Of Service*) : type de service à appliquer au paquet en fonction de certains paramètres comme le délai de transit, la sécurité
- **total length** : longueur totale du datagramme (en octets)
- **identification** : valeur fournie par la source aidant la destination au réassemblage des différents fragments du datagramme
- **flags** : utilisé par la fragmentation et composé de
  - DF (*Don't Fragment*)
  - MF (*More Fragment*)
  - réservé
- **offset** : déplacement par rapport au datagramme initial (en multiple de 8 octets)
- **TTL** (*Time To Live*) : limite supérieure du temps de vie d'un datagramme
- **protocol** : protocole utilisé pour le champ de données
  - 1 pour ICMP
  - 6 pour TCP
  - 17 pour UDP
- **checksum** : zone de contrôle d'erreur portant uniquement sur l'en-tête du datagramme
- **source address** : @ IP de la source du datagramme
- **destination address** : @ IP de la destination du datagramme
- **options** : fonctions de contrôle utiles dans certaines situations (estampillage temporel, sécurité, routage particulier, etc.)
- **padding** : pour aligner l'en-tête sur 32 bits



# IP - Plan

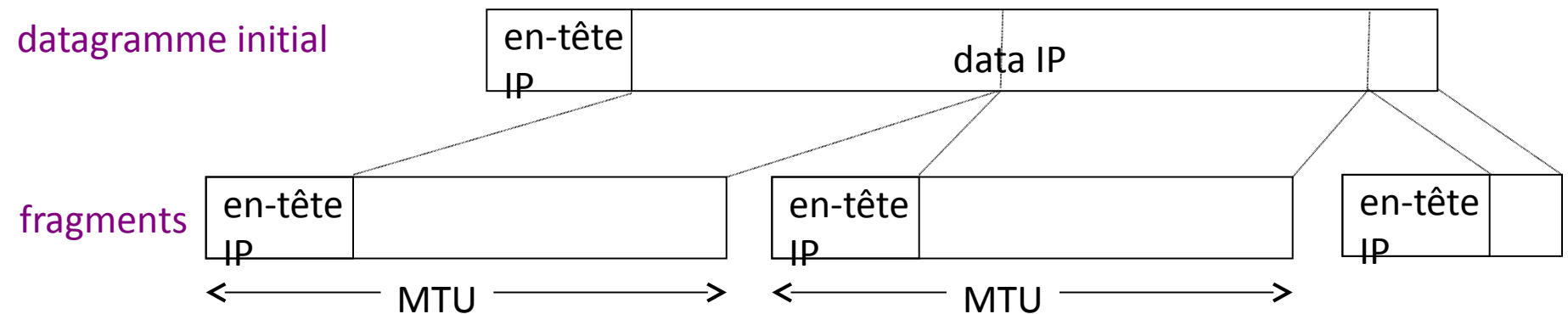
1. Interconnexion IP
2. Relayage et routage IP
3. Adressage et nommage
4. Adressage IP
5. Datagramme IP
6. **Fragmentation**
7. ICMP

# Fragmentation et réassemblage

- Motivations
  - l'Internet est par nature hétérogène
  - le MTU (Maximum Transmission Unit) varie selon la technologie
  - certains protocoles de niveau supérieur génèrent des datagrammes de longueur supérieure au MTU
- ↳ adaptation de la taille du datagramme au MTU

# Fragmentation et réassemblage

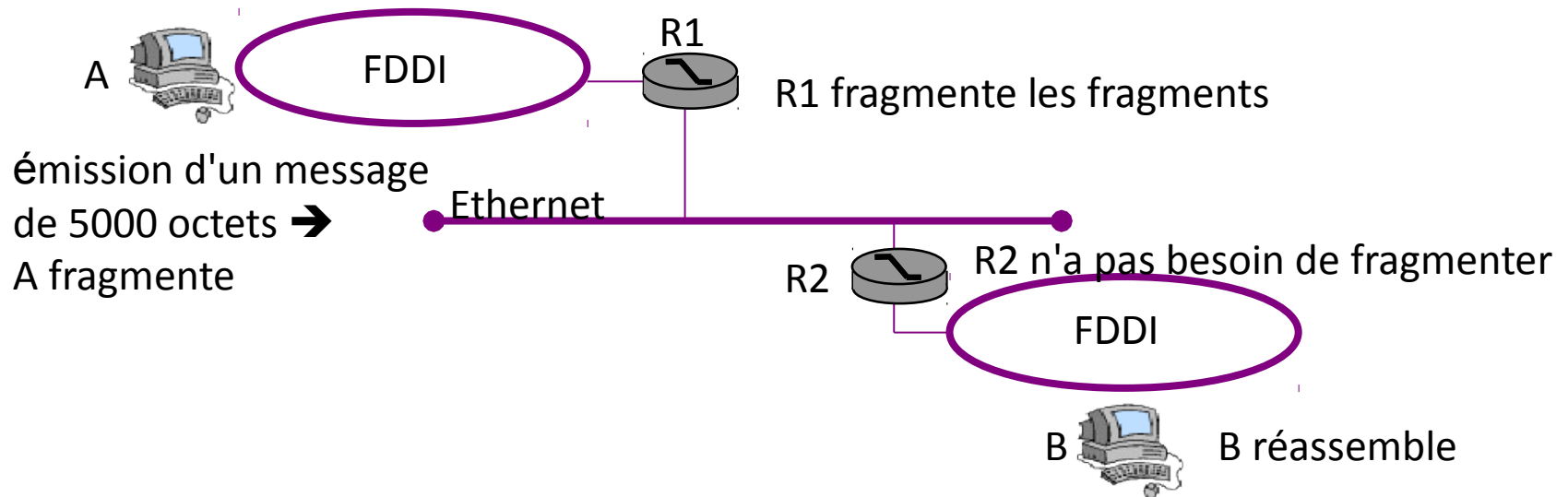
- Principe
  - découpage de la charge utile et duplication de l'en-tête



- réassemblage par le destinataire final

# Fragmentation et réassemblage

- exemple
  - MTU Ethernet = 1500 octets
  - MTU FDDI = 4500 octets

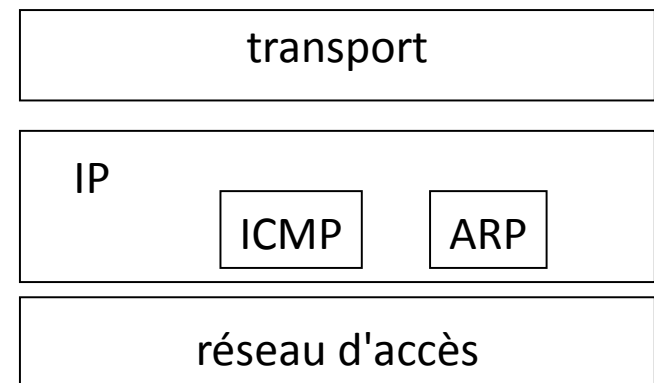


# IP - Plan

1. Interconnexion IP
2. Relayage et routage IP
3. Adressage et nommage
4. Adressage IP
5. Datagramme IP
6. Fragmentation
7. **ICMP**

# Protocole de contrôle ICMP

- Motivation
  - pas de signalisation dans IP
    - pas de retour d'information
    - pas de messages d'anomalies
- **ICMP** (Internet Control Message Protocol)
  - instrumentation et test
  - signalisation d'anomalies
  - mise en œuvre obligatoire
  - messages ICMP encapsulés dans des datagrammes IP
  - même si :



# Messages ICMP

- format
  - les messages ICMP ont tous le même format pour le premier mot de 32 bits

type	code	checksum
------	------	----------

- champ type

Type	Message	Objet
0	Echo Reply	Réponse en écho.
3	Destination Unreachable	Destination inaccessible.
4	Source Quench	Interruption de la source.
5	Redirect	Redirection, changement de route.
8	Echo	Demande d'écho.
11	Time Exceeded	Temps de vie d'un datagramme dépassé.
12	Parameter Problem	Datagramme mal formé.
13	Timestamp	Demande de date d'estampillage.
14	Timestamp Reply	Réponse à une demande d'estampillage.
15	Information Request	Demande d'information.
16	Information Reply	Réponse à une demande d'information.
17	Address Mask Request	Demande de masque d'adresse.
18	Address Mask Reply	Réponse à une demande de masque d'adresse.

# Outil ping

- Principe
  - exploite la fonction d'écho de ICMP
  - un routeur ou un hôte recevant un "echo request" retourne un "echo reply"
- ↳ permet de
  - tester l'accessibilité d'une machine
  - obtenir des statistiques sur la qualité de la route
- exemple

```
$ ping castor.univ-reunion.fr
PING castor.univ-reunion.fr (194.199.73.51): 56 data bytes
64 bytes from 194.199.73.51: icmp_seq=0 ttl=246 time=570.800 ms
64 bytes from 194.199.73.51: icmp_seq=1 ttl=246 time=581.364 ms
64 bytes from 194.199.73.51: icmp_seq=2 ttl=246 time=571.022 ms
64 bytes from 194.199.73.51: icmp_seq=3 ttl=246 time=572.722 ms
64 bytes from 194.199.73.51: icmp_seq=4 ttl=246 time=579.121 ms
64 bytes from 194.199.73.51: icmp_seq=5 ttl=246 time=571.619 ms
^C
----castor.univ-reunion.fr PING Statistics----
6 packets transmitted, 6 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 570.800/574.441/581.364/4.598 ms
```