

# Couche Réseau

# Fonction de La couche réseau

Le rôle principal de cette couche est de :

- Transporter des paquets de la **source** vers la **destination** via les différentes **nœuds de commutation** du réseaux traversés

Ce rôle est assuré par un ensemble de fonctions :

- Adressage et routage (**acheminement des paquets**)
- Fragmentation et réassemblage (**conversion de messages en paquets**)
- Régulation et répartition de la charge (**contrôle de flux**)

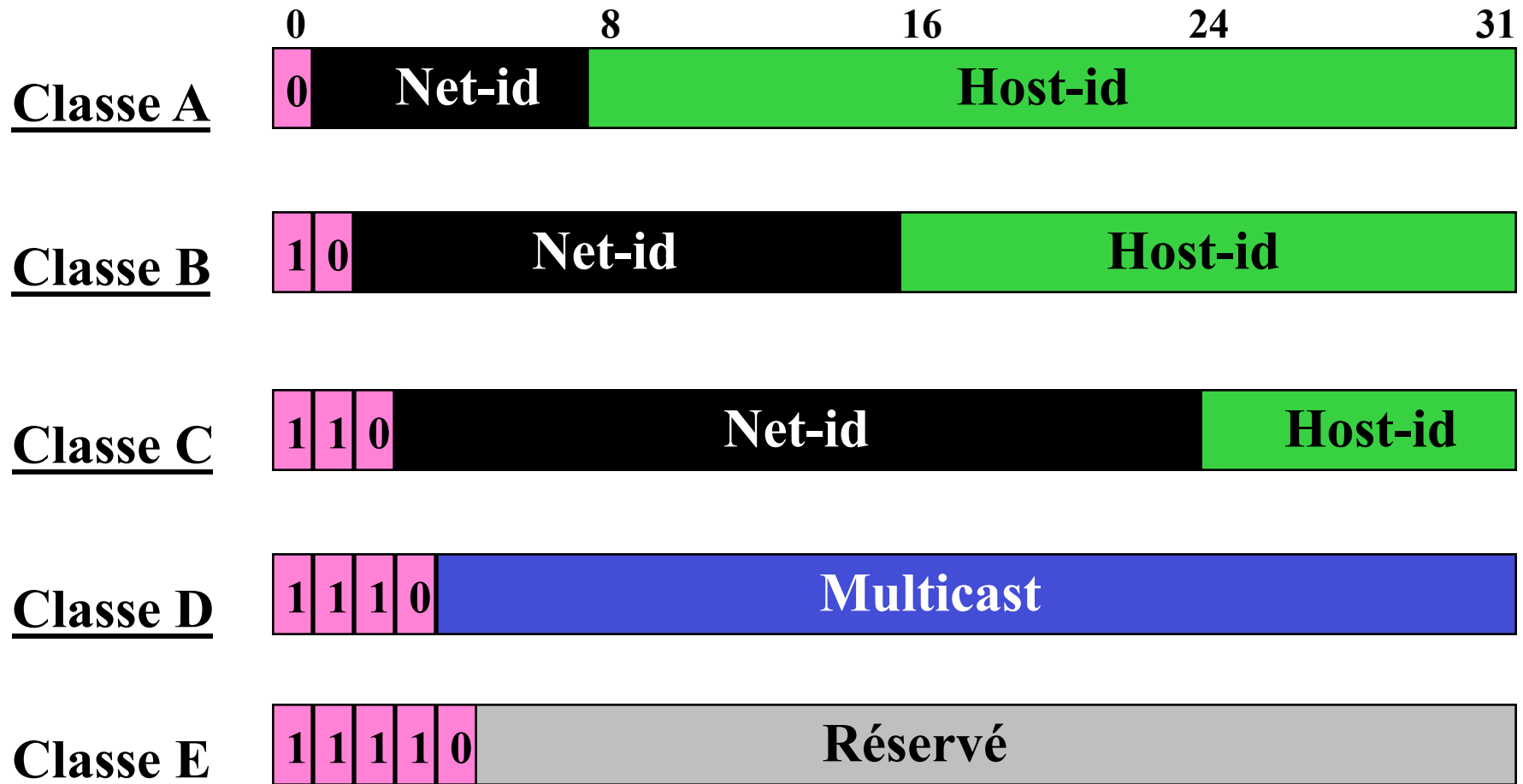
# Le protocole IP : L' adressage Internet

- **But** : fournir un service de communication universel permettant à toute machine de communiquer avec toute autre machine de l' interconnexion
- Une machine est identifiée par :
  - un **nom**,
  - une **adresse** qui doit être un identificateur universel de la machine,
  - une **route** précisant comment la machine peut être atteinte.

## Les classes d'adressage

- Une adresse = 32 bits dite "**Internet address**" ou "**IP address**" constituée d'une paire (**netid, hostid**) où **netid** identifie un réseau et **hostid** identifie une machine sur ce réseau.
- Cette paire est structurée de manière à définir **cinq classes d'adresse**

# Le protocole IP : L'adressage Internet



# Le protocole IP : L'adressage Internet

On dispose en théorie des plages d'adresses suivantes :

Classe	Plage	
A	0.0.0.0	127.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D	224.0.0.0	239.255.255.255
E	240.0.0.0	247.255.255.255

# Le protocole IP : L'adressage Internet

Classe A : 126 réseaux et 16777214 machines par réseaux

Classe B : 16382 réseaux et 65534 machines par réseaux

Classe C : 2097150 réseaux et 254 machines par réseaux

- Il existe des adresse dites non routables.
- Ces adresse sont réservées à usage interne, ou dans le cas de réseaux privées

Classe A : 10.0.0.0

Classe B : 172.16.0.0 à 172.31.0.0

Classe C : 192.168.0.0 à 192.168.255.0

# Le protocole IP : L'adressage Internet

- Notation décimale

utilisation de quatre entiers décimaux séparés par un point, chaque entier représente un octet de l'adresse IP : 128.10.2.30

10000000 00001010 00000010 00011110

- Adresses particulières

- **Adresses réseau** : adresse IP dont la partie **hostid** ne comprend que des **zéros**; => la valeur zéro ne peut être attribuée à une machine réelle : **172.20.0.0** désigne le réseau de classe **B 172.20**.
- **Adresse machine locale** : adresse IP dont le champ réseau (**netid**) ne contient que des zéros;
- **hostid = 0** (=> **tout à zéro**), l'adresse est utilisée au **démarrage** du système afin de connaître **l'adresse IP (RARP)**.



# Le protocole IP : L'adressage Internet

## Détermination du netmask

- Soit le **sous réseau** de **classe C** **192.168.16.x** ( $0 < x < 255$ ).
- Nous avons donc un réseau de **n=255** machines,
- **netmask = NON (n) = NON (0.0.0.255), soit (255.255.255.0)**
- Le **netmask** est donc nécessaire pour connaître le nombre de machines présente dans le sous-réseau.
- Le **netmask** permet de diviser la classe C en plusieurs sous réseaux.

### Exemple:

Adresse de sous réseau	Netmask	Nb de machines
192.168.16.0	255.255.255.128	128
192.168.16.128	255.255.255.224	64
192.168.16.192	255.255.255.224	32
192.168.16.224	255.255.255.224	32

# Le protocole IP : L' adressage Internet

## Calcul de l' adresse de diffusion

- L' adresse de **diffusion**, ou adresse de **broadcast**, permet d' adresser toutes les machines sur le même réseau que le votre d' une seule opération.
- Elle est **calculée à partir du netmask** et de **l' adresse de sous-réseau**.
- Si **R** est **l' adresse de sous-reseau** et **N** le **netmask** associé, on peut calculer l' adresse de **broadcast** par la formule suivante:

$$\mathbf{B = NON (N) OU (R)}$$

### Exemple:

$$N = 255.255.255.128$$

$$R = 192.168.16.0$$

$$B = \text{NON } (255.255.255.128) \text{ OU } 192.168.16.0$$

$$= (0.0.0.127) \text{ OU } 192.168.16.0$$

$$= 192.168.16.127$$

# Le protocole IP : L' adressage Internet

## L' adresse de passerelle

L' adresse de passerelle indique si nécessaire à quelle machine doit-on s' adresser lorsqu' une requête n' est pas destinée à une machine de notre réseaux

Pour une classe C 192.16.168.0, l' adresse 192.16.168.254 est réservée au routeur ou passerelle.

Comment déterminer si la machine émettrice se trouve dans le même réseau que la machine qu' elle souhaite contacter?

L' opération suivante est effectuée :  $V = IP_{dest} \text{ ET Netmask}$

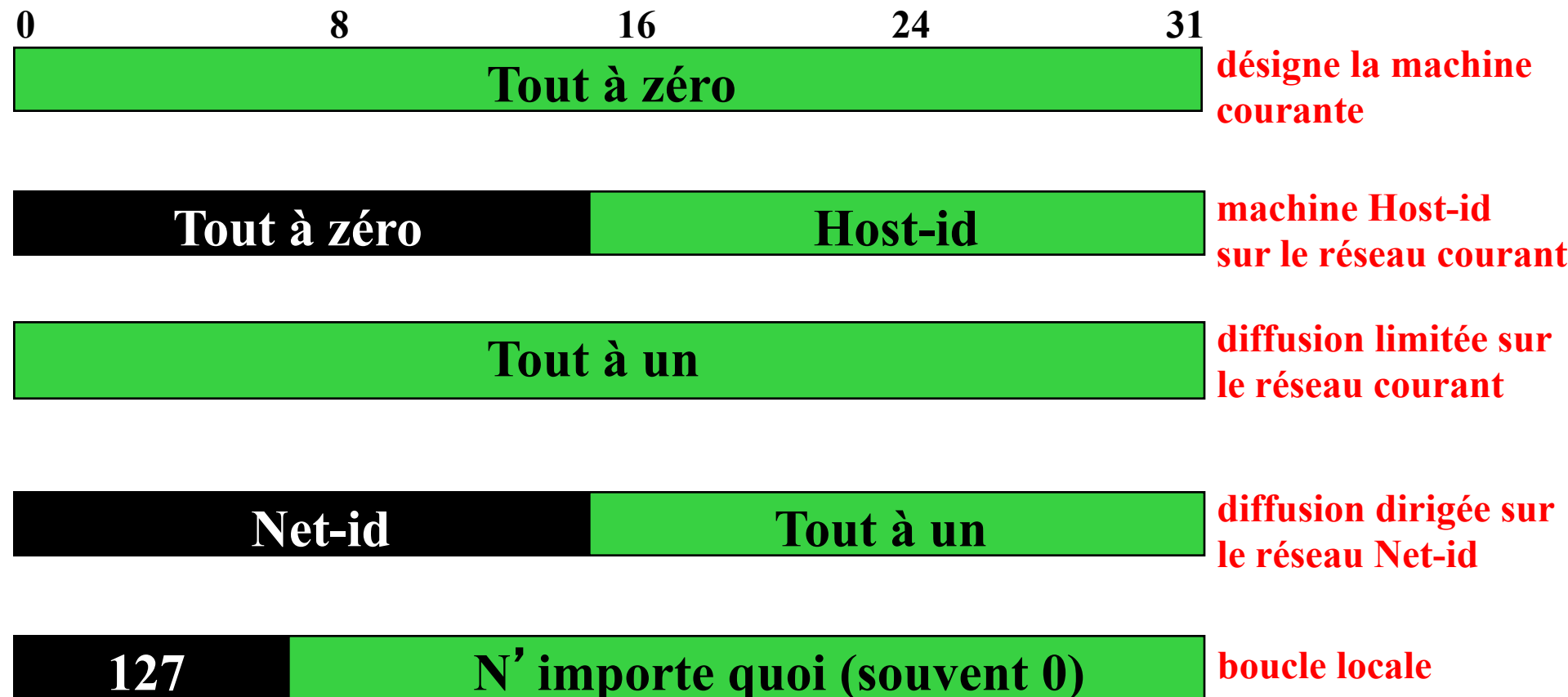
si  $V$  donne la même adresse de sous-réseau que la machine émettrice, alors la machine de destination se trouve dans le même réseau que la machine émettrice (on a pas besoin de la passerelle).

# Le protocole IP : L'adressage Internet

- Adresses de diffusion : la partie **hostid** ne contient que **des 1**
- En conséquence, une adresse **IP** dont la valeur **hostid** ne comprend que **des 1** ne peut être **attribuée** à une **machine réelle**.
- Adresse de boucle locale : l'adresse réseau **127.0.0.0** est réservée pour la **désignation de la machine locale**, c'est à dire la communication **intra-machine**.

**Une adresse réseau 127 ne doit, en conséquence, jamais être véhiculée sur un réseau et un routeur ne doit jamais router un datagramme pour le réseau 127.**

# Le protocole IP : L'adressage Internet



# Le sous-adressage

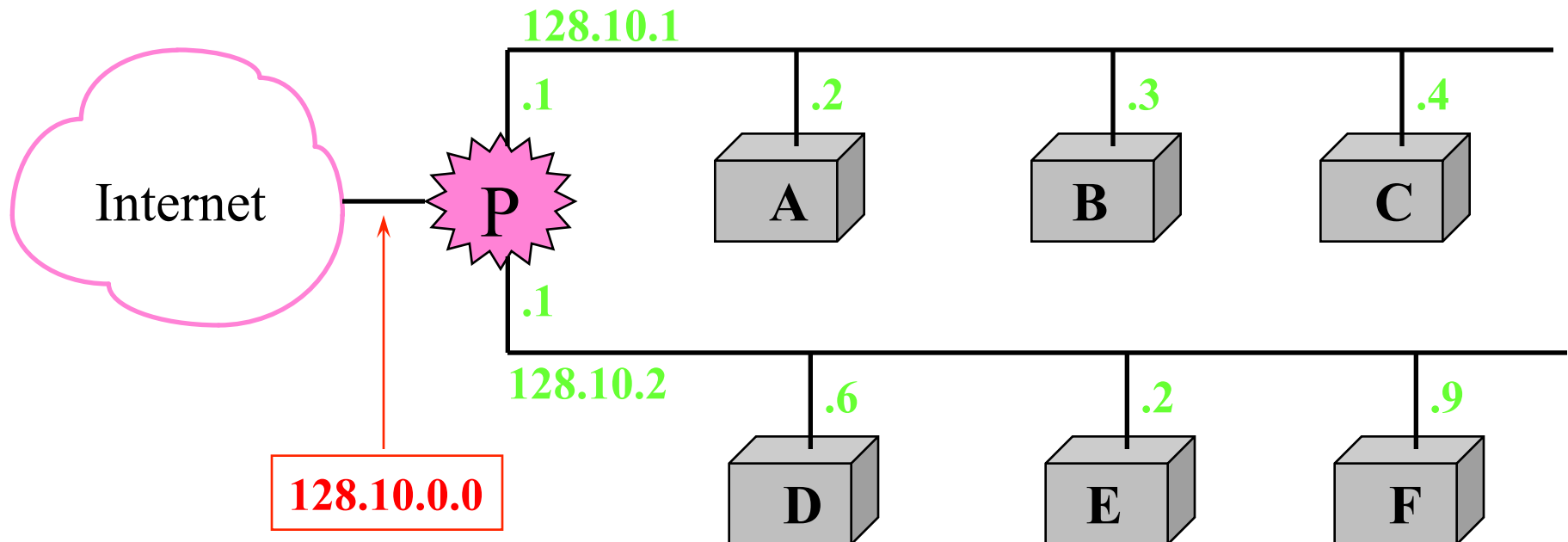
- Le sous-adressage est une extension du plan d'adressage initial
- Devant la croissance du nombre de réseaux de l'Internet, il a été introduit afin de limiter la consommation d'adresses IP, qui permet également de diminuer :
  - la gestion administrative des adresses IP,
  - la taille des tables de routage des passerelles,
  - la taille des informations de routage,
  - le traitement effectué au niveau des passerelles.

## Principes

- A l'intérieur d'une entité associée à une adresse IP de classe A, B ou C, plusieurs réseaux physiques partagent cette adresse IP.
- On dit alors que ces réseaux physiques sont des sous-réseaux (*subnet*) du réseau d'adresse IP.

# Le sous-adressage

Les sous-réseaux **128.10.1.0** et **128.10.2.0** sont notés seulement avec le **NetId**, les machines seulement avec le **Hostid** ; exemple IP(**F**) = **128.10.2.9**



La **passerelle P** accepte tout le trafic destiné au réseau **128.10.0.0** et sélectionne le **sous-réseau** en fonction du **troisième octet** de l'adresse destination.

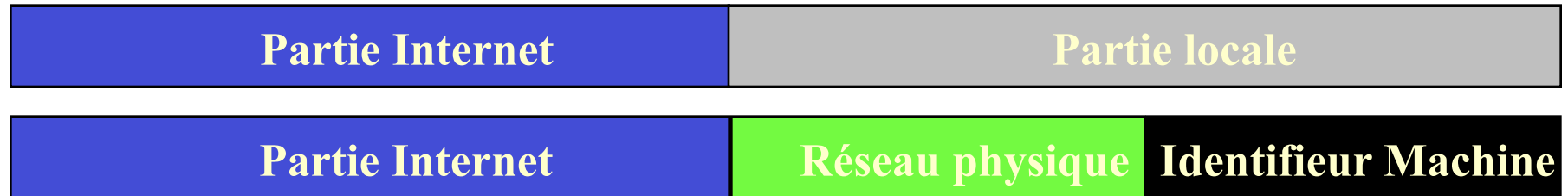
## Le sous-adressage

- La passerelle doit router vers l'un ou l'autre des sous-réseaux ; le découpage du site en sous-réseaux a été effectué sur la base du troisième octet de l'adresse :
  - les adresses des machines du premier sous-réseau sont de la forme 128.10.1.X,
  - les adresses des machines du second sous-réseau sont de la forme 128.10.2.X.
- Pour sélectionner l'un ou l'autre des sous-réseaux, P examine le troisième octet de l'adresse destination : si la valeur est 1, le datagramme est routé vers réseau 128.10.1.0, si la valeur est 2, il est routé vers le réseau 128.10.2.0.



# Le sous-adressage

- Conceptuellement, la **partie locale dans le plan d'adressage initial** est **subdivisée** en “**partie réseau physique**” + “**identification de machine (hostid) sur ce sous-réseau**” :



- ✦ «**Partie Internet**» correspond au **NetId** (plan d'adressage initial)
- ✦ «**Partie locale**» correspond au **hostid** (plan d'adressage initial)
- ✦ les champs «**Réseau physique**» et «**identifieur Machine**» sont de taille variable; la **longueur des 2 champs étant toujours égale à la longueur** de la «**Partie locale**».

## Le sous-adressage

- Le choix du découpage dépend des perspectives d'évolution du site:
  - Exemple Classe B : 8 bits pour les parties réseau et machine donnent un potentiel de 254 sous-réseaux et 254 machines par sous-réseau, tandis que 3 bits pour la partie réseau et 13 bits pour le champ machine permettent 6 réseaux de 8190 machines chacun.
  - Exemple Classe C : 4 bits pour la partie réseau et 4 bits pour le champ machine permettent 14 réseaux de 14 machines chacun.

# Le protocole ARP : Address Resolution Protocol

## Le besoin

- La communication entre machines ne peut s'effectuer qu'à travers l'interface physique
- Les applicatifs ne connaissent que des adresses IP, **comment établir le lien adresse IP/adresse physique?**

## La solution : ARP

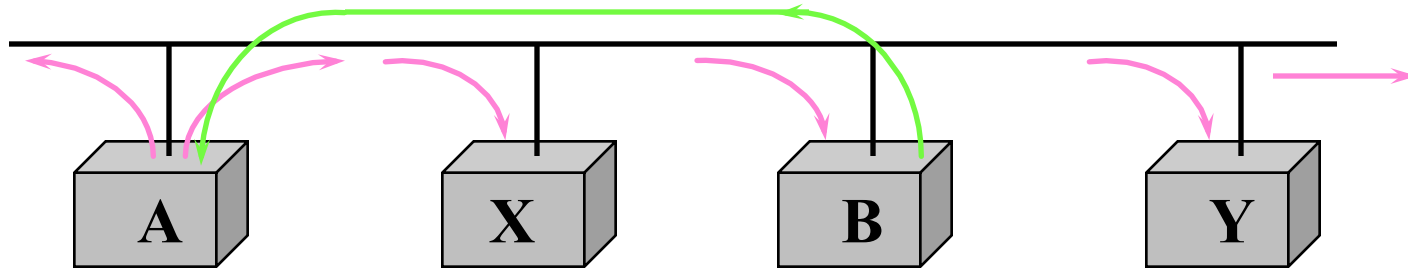
- Mise en place dans TCP/IP d'un protocole de bas niveau appelé **Address Resolution Protocol (ARP)**
- **Rôle de ARP** : fournir à une machine donnée **l'adresse physique** d'une autre machine située sur le même réseau à partir de **l'adresse IP** de la machine destinatrice

## La technique

- Diffusion d'adresse sur le réseau physique
- La machine d'adresse IP émet un message contenant son adresse physique
- Les machines non concernées ne répondent pas
- Gestion cache pour ne pas effectuer de requête ARP à chaque émission

# Le protocole ARP : Address Resolution Protocol

- L'association **adresse physique - adresse IP** de l'émetteur est incluse dans la **requête ARP** de manière à ce que les récepteurs **enregistrent** l'association dans leur propre **mémoire cache**



- Pour connaître l'adresse physique de **B, PB**, à partir de son adresse **IP, IB**, la machine **A diffuse une requête ARP** qui contient l'adresse **IB** vers toutes les machines; la machine **B répond avec un message ARP** qui contient la paire (**IB, PB**).

# Le protocole ARP : Address Resolution Protocol

## Format du message ARP

- La requête ARP est véhiculée dans un message protocolaire lui-même encapsulé dans la trame de liaison de données.
- Lorsque la trame arrive à destination, la couche liaison de données détermine l'entité responsable du message encapsulé;

**Exemple:** champ type de la trame Ethernet: **0806** pour ARP

# RARP: Reverse Address Resolution Protocol

## Le besoin

- L'adresse IP d'une machine est configurable (**elle dépend du réseau sur lequel elle se trouve**) et est souvent enregistrée sur la mémoire secondaire où le système d'exploitation l'accède au démarrage.
- Ce fonctionnement usuel n'est plus possible dans le cas où la machine est une station sans mémoire secondaire.

**Problème** : déterminer un mécanisme permettant à la station d'obtenir son adresse IP depuis le réseau.

## La solution

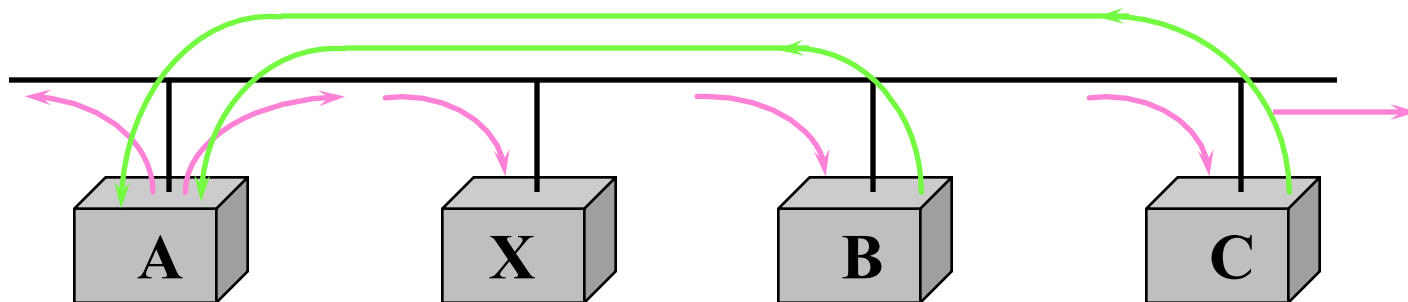
- Protocole de bas niveau appelé **Reverse Address Resolution Protocol**
- Permet d'obtenir son **adresse IP** à partir de l'adresse physique qui lui est associée.

## Fonctionnement

Serveur **RARP** sur le réseau physique; son **rôle**: fournir les **adresses IP** associées aux **adresses physiques** des stations du réseau;

# RARP: Reverse Address Resolution Protocol

- Le serveur possède une base de données contenant les couples adresse **physique/adresse IP**,
- les stations émettent une requête **RARP** sur le réseau, consistant à demander l'adresse IP qui est associée à leur adresse physique,
- Les requêtes RARP sont propagées vers le ou les serveur(s) RARP par mécanisme de diffusion. Le(s) serveur(s) RARP répond(ent) par un message de type RARP.



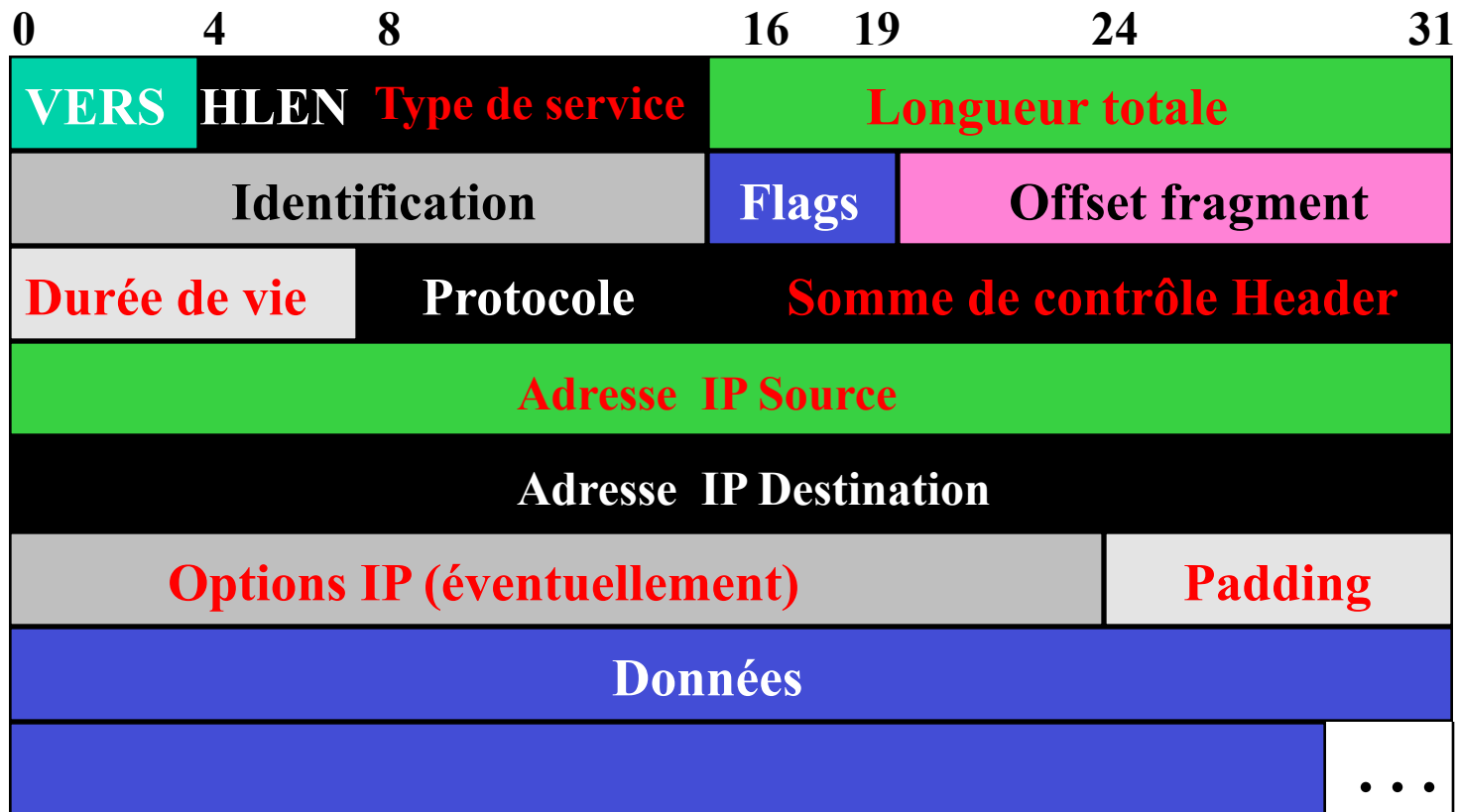
Pour connaître son adresse IP, **A** diffuse sur le réseau, une requête **RARP** qui la désigne comme destinataire

**Les Serveurs RARP (B et C) répondent à la requête.**

# IP : Internet Protocol (le datagramme)

## Le datagramme IP

L'unité de transfert de base dans un réseau Internet est le datagramme qui est constituée d'un en-tête et d'un champ de données:





# IP : Internet Protocol (le datagramme)

## Signification des champs du datagramme IP :

- **VERS** : numéro de version de protocole IP, actuellement version 4,
- **HLEN** : longueur de l'en-tête en mots de 32 bits,
- **Longueur totale** : longueur totale du datagramme (en-tête + données)
- **Type de service** : indique comment le datagramme doit être géré :



- **PRECEDENCE (3 bits)** : définit la priorité du datagramme.
- **Bits D, T, R** : indiquent le **type d'acheminement** désiré du **datagramme**, permettant à une passerelle de choisir entre plusieurs **routes** (si elles existent) : **D** signifie délai court, **T** signifie débit élevé et **R** signifie grande fiabilité.

# IP : Internet Protocol (le datagramme)

## FRAGMENT OFFSET, FLAGS, IDENTIFICATION :

les champs de la fragmentation.

- Sur toute machine ou passerelle mettant en oeuvre TCP/IP, une unité maximale de transfert (*Maximum Transfert Unit* ou MTU) définit la taille maximale d'un datagramme véhiculé sur le réseau physique correspondant.
- Lorsque le datagramme est routé vers un réseau physique dont le MTU est plus petit que le MTU courant, la passerelle fragmente le datagramme en un certain nombre de fragments véhiculés par autant de trames sur le réseau physique correspondant.
- Lorsque le datagramme est routé vers un réseau physique dont le MTU est supérieur au MTU courant, la passerelle route les fragments tels quels (rappel : les datagrammes peuvent emprunter des chemins différents).
- Le destinataire final reconstitue le datagramme initial à partir de l'ensemble des fragments reçus; la taille de ces fragments correspond au plus petit MTU emprunté sur le réseau. Si un seul des fragments est perdu, le datagramme initial est considéré comme perdu : la probabilité de perte d'un datagramme augmente avec la fragmentation.

# IP : Internet Protocol (le datagramme)

- **FRAGMENT OFFSET** : indique le déplacement des données contenues dans le fragment par rapport au datagramme initial. C'est un **multiple de 8 octets**; la taille du fragment est donc également un multiple de 8 octets.
- Chaque **fragment a une structure identique à celle du datagramme initial**, seul les **champs FLAGS et FRAGMENT OFFSET** sont spécifiques.

# IP : Internet Protocol (le datagramme)

- **Longueur totale** : taille du fragment et non pas celle du datagramme initial, à partir du dernier fragment (**TOTAL LENGTH, FRAGMENT OFFSET et FLAGS**) on peut déterminer la taille du datagramme initial.
- **IDENTIFICATION (N° de datagramme)** : entier qui identifie le datagramme initial (**utilisé pour la reconstitution de datagramme à partir des fragments qui ont tous la même valeur**).
- **FLAGS** contient un bit appelé "*do not fragment*" (01X)
- un autre bit appelé "*More fragments*" (**FLAGS = 001** signifie d'autres **fragments à suivre**) permet au destinataire final de reconstituer le datagramme initial en identifiant les différents fragments (**milieu ou fin du datagramme initial**)
- les passerelles doivent accepter sans les fragmenter, les **datagrammes** de **longueur 576 octets**.

# IP : Internet Protocol (le datagramme)

- Durée de vie

- Ce champ indique en secondes, la durée maximale de transit du datagramme sur l'internet. La machine qui émet le datagramme définit sa durée de vie.
- Les passerelles qui traitent le datagramme doivent décrémenter sa durée de vie du nombre de secondes (1 au minimum) que le datagramme a passé pendant son séjour dans la passerelle; lorsque celle-ci expire le datagramme est détruit et un message d'erreur est renvoyé à l'émetteur.

- Protocole

Ce champ identifie le protocole de niveau supérieur dont le message est véhiculé dans le champ données du datagramme :

- 6 : TCP,
- 17 : UDP,
- 1 : ICMP.

## IP : Internet Protocol (le datagramme)

- Somme de contrôle de l'en-tête

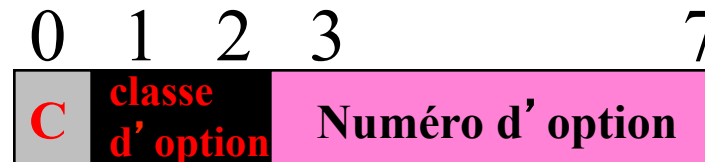
- Ce champ permet de détecter les erreurs survenant dans l'en-tête du datagramme, et par conséquent l'intégrité du datagramme.
- Le total de contrôle d'IP porte sur l'en-tête du datagramme et non sur les données véhiculées. Lors du calcul, le champ HEADER CHECKSUM est supposé contenir la valeur 0 :

- XXXX XXXX XXXX XXXX (VERS, HLEN, TYPE OF SERVICE)
- XXXX XXXX XXXX XXXX (TOTAL LENGTH)
- XXXX XXXX XXXX XXXX (ID. FLAGS, FRAGMENT OFFSET)
- XXXX XXXX XXXX XXXX (TIME TO LIVE, PROTOCOL)
- 0000 0000 0000 0000 (HEADER CHECKSUM)
- XXXX XXXX XXXX XXXX (IP SOURCE )
- XXXX XXXX XXXX XXXX (IP SOURCE)
- XXXX XXXX XXXX XXXX (IP DESTINATION)
- XXXX XXXX XXXX XXXX (IP DESTINATION)
- ... (OPTIONS éventuelles + PADDING)

# IP : Internet Protocol (le datagramme)

- OPTIONS

- Le champ **OPTIONS** est **facultatif** et de longueur variable. Les options concernent essentiellement des fonctionnalités de mise au point (**enregistrement de route, horodatage, bourrage ...**).
- Une option est définie par un champ octet :



- copie (C) indique que l'option doit être recopiée dans tous les fragments (**c=1**) ou bien uniquement dans le premier fragment (**c=0**).
- les bits classe d'option et numéro d'option indiquent le type de l'option.
- Une option particulière de ce type est:

# IP : Internet Protocol (le datagramme)

- Enregistrement de route (classe = 0, option = 7) :

permet à la source de créer une liste d'adresse IP vide et de demander à chaque passerelle d'ajouter son adresse dans la liste.



# IP : Internet Protocol (le datagramme)

- Routage strict prédéfini par l'émetteur (classe = 0, option = 9):

Prédéfini le routage qui doit être utilisé dans l'interconnexion en indiquant la suite des adresses IP dans l'option.

Le chemin spécifié ne tolère aucun autre intermédiaire; une erreur est retournée à l'émetteur si une passerelle ne peut appliquer le routage spécifié.

## IP : Internet Protocol (le datagramme)

- Routage lâche prédéfini par l'émetteur (classe = 0, option = 3):

Cette option autorise, entre deux passages obligés, le transit par d'autres intermédiaires :

# IP : Internet Protocol (le datagramme)

- Horodatage (classe = 2, option = 4) :

Cette option permet d'obtenir les temps de passage (*timestamp*) des datagrammes dans les passerelles. Exprimé en heure et date universelle.

Une liste de couples (**adresse IP - horodatage**) est réservée par l'émetteur; les passerelles ont à charge de remplir un champ lors du passage du datagramme.

# IP : Internet Protocol (le datagramme)

- Les **horodatages**, bien qu'exprimés en temps universel, ne constituent qu'une **estimation sur le temps de passage** car **les horloges des machines situées sur les réseaux ne sont pas synchronisées**.
- ICMP

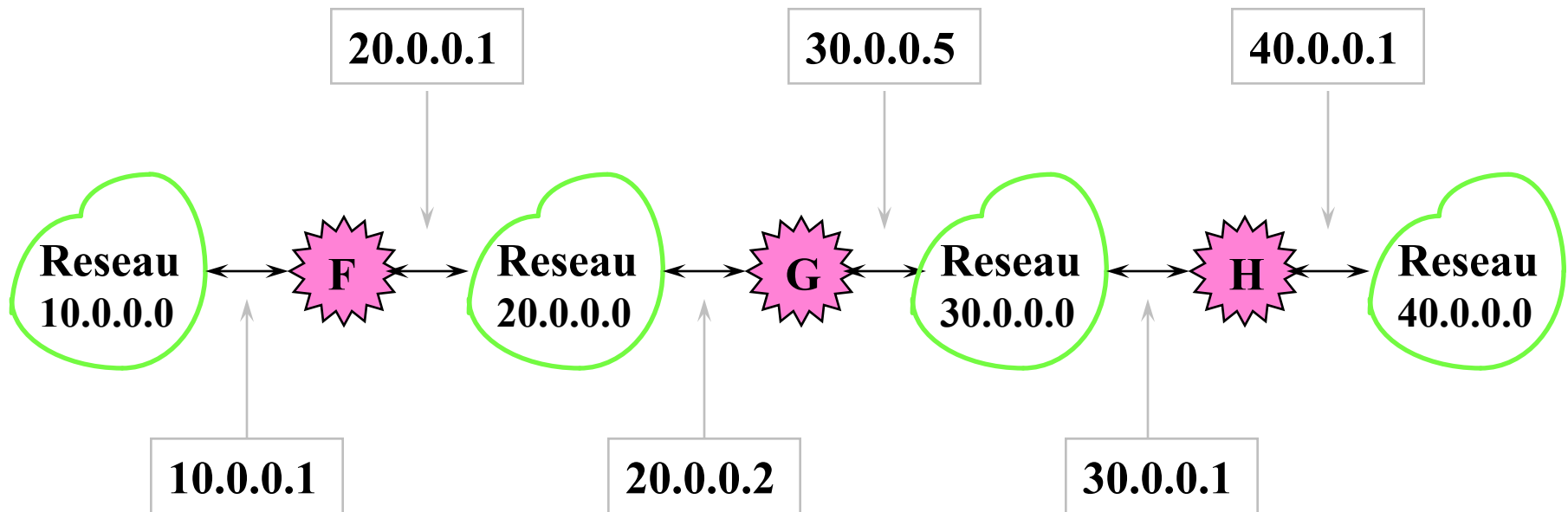
# Routage des datagrammes

- Le **routage** est le processus permettant à un **datagramme** d'être **acheminé** vers le **destinataire** lorsque celui-ci **n'est pas sur le même réseau physique** que l'émetteur.
- Le chemin parcouru est le **résultat du processus de routage** qui effectue les choix **nécessaires afin d'acheminer le datagramme**.
- Les **routeurs forment une structure coopérative** de telle manière qu'un **datagramme transite de passerelle en passerelle** jusqu'à ce que l'une d'entre elles **le délivre à son destinataire**.
- Un **routeur possède** deux ou plusieurs connexions réseaux tandis qu'une **machine possède généralement qu'une seule connexion**.
- **Machines et routeurs participent au routage** :
  - les machines doivent déterminer si le **datagramme doit être délivré sur le réseau physique sur lequel elles sont connectées** (routage direct) ou bien si le **datagramme doit être acheminé vers une passerelle**; dans ce cas (**routage indirect**), elle doit identifier la passerelle appropriée.
  - les **passerelles effectuent le choix de routage vers d'autres passerelles** afin d'acheminer le datagramme vers sa destination finale.

# Routage des datagrammes

- Les **tables de routage IP**, pour des raisons évidentes d'**encombrement**, renseignent seulement **les adresses réseaux** et **non pas les adresses machines**.
- Typiquement, **une table de routage** contient des couples **(R, P)** où **R** est l'**adresse IP** d'un **réseau destination** et **P** est l'**adresse IP** de la **passerelle correspondant au prochain saut dans** le cheminement vers le réseau destinataire.
- La passerelle ne connaît pas le chemin complet pour atteindre la destination.
- Pour une table de routage contenant des couples **(R, P)** et appartenant à la **machine M**, **P** et **M** sont **connectés** sur le même **réseau physique** dont l'**adresse de niveau réseau** (partie **Netid** de l'**adresse IP**) est **R**.

# Routage des datagrammes



<b>Pour atteindre les machines du réseau</b>	10.0.0.0	20.0.0.0	30.0.0.0	40.0.0.0
<b>Router vers</b>	20.0.0.1	direct	direct	30.0.0.1

**Table de routage de G**

# Routage des datagrammes

**Route\_Datagramme\_IP(datagramme, table\_de\_routage)**

- Extraire l'adresse IP destination, ID, du datagramme,
- Calculer l'adresse du réseau destination, IN.
- Si IN correspondant à une adresse de réseau directement accessible,  
envoyer le datagramme vers sa destination, sur ce réseau.
- sinon si dans la table de routage, il existe une route vers ID  
router le datagramme selon les informations contenues dans la table de routage.
- sinon si IN apparaît dans la table de routage,  
router le datagramme selon les informations contenues dans la table de routage.
- sinon s'il existe une route par défaut  
router le datagramme vers la passerelle par défaut.
- sinon déclarer une erreur de routage.



# Routage des datagrammes

- Après exécution de l' algorithme de routage, IP transmet le datagramme ainsi que l' adresse IP déterminée, à l' interface réseau vers lequel le datagramme doit être acheminé.
- L' interface physique détermine alors l' adresse physique associée à l' adresse IP et achemine le datagramme sans l' avoir modifié.
- Si le datagramme est acheminé vers une autre passerelle, il est à nouveau géré de la même manière, et ainsi de suite jusqu' à sa destination finale.

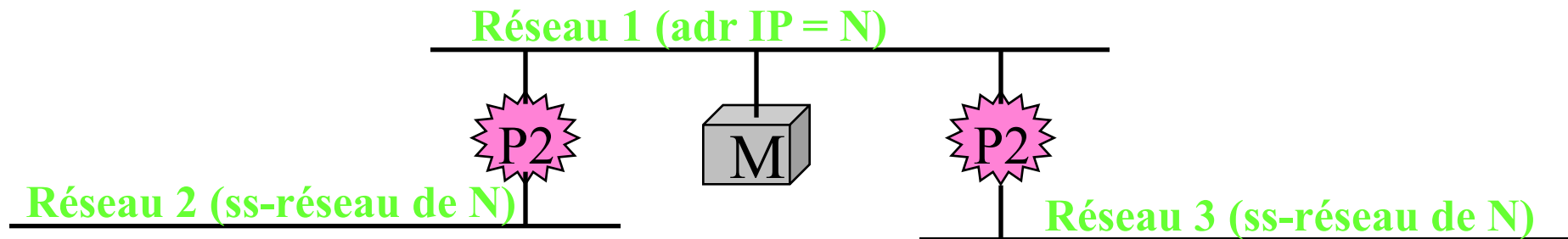
# Routage des datagrammes

- Les **datagrammes entrants** sont traités différemment selon qu’ il sont reçus par une **machine** ou une **passerelle** :
- **Machine** : le logiciel IP examine l’ adresse destination à l’ intérieur du datagramme
  - si cette **adresse IP** est **identique** à celle de la machine, IP **accepte** le datagramme et **transmet** son contenu à la **couche supérieure**.
  - sinon, le **datagramme est rejeté**; une **machine recevant** un **datagramme destiné** à une **autre machine** ne doit pas router le datagramme.
- **Passerelle** : IP détermine si le **datagramme** est **arrivé à destination** et dans ce cas le **délivre à la couche supérieure**. Si le **datagramme** n’ a pas atteint sa destination finale, il est **routé selon l’ algorithme de routage précédemment décrit**.

# Le sous-adressage

## Routage avec sous-réseaux

- Le routage IP initial a été étendu à l'adressage en sous-réseaux;
- l'algorithme de routage obtenu doit être présent dans les machines ayant une adresse de sous-réseau, mais également dans les autres machines et passerelles du site qui doivent acheminer les datagrammes vers ces sous-réseaux.



M doit utiliser le routage de sous-réseaux pour décider si elle route vers les passerelles P1 ou P2 bien qu'elle même soit connectée à un réseau (Réseau 1) n'ayant pas de sous-adressage

# Le sous-adressage

Le routage unifié : Une entrée dans la table de routage =  
(masque de sous-réseau, adresse sous-réseau, adresse de la passerelle)

Algorithme de routage unifié :

- `Route_IP_Datagram(datagram, routing_table)`
- Extraire l'adresse ID de destination du datagramme,
- Calculer l'adresse IN du réseau destination,
- Si IN correspond à une adresse réseau directement accessible  
envoyer le datagramme sur le réseau physique correspondant,
- Sinon
  - Pour chaque entrée dans la table de routage,
    - $N = (ID \ \& \ \text{masque de sous-réseau de l'entrée})$
    - Si N est égal au champ adresse réseau de l'entrée  
router le datagramme vers la passerelle correspondante,
  - Fin\_Pour
- Si aucune entrée ne correspond, déclarer une erreur de routage.