# La couche réseau
# -
# Pile protocolaire TCP/IP

# Présentation

Objectif d'après OSI

- Interconnexion des réseaux
- Adressage universel des machines
- Gestion de la façon d'acheminer les informations à travers le réseau
- Gestion des problèmes de congestion réseau
- Gestion des erreurs non traitées par la couche Liaison de données

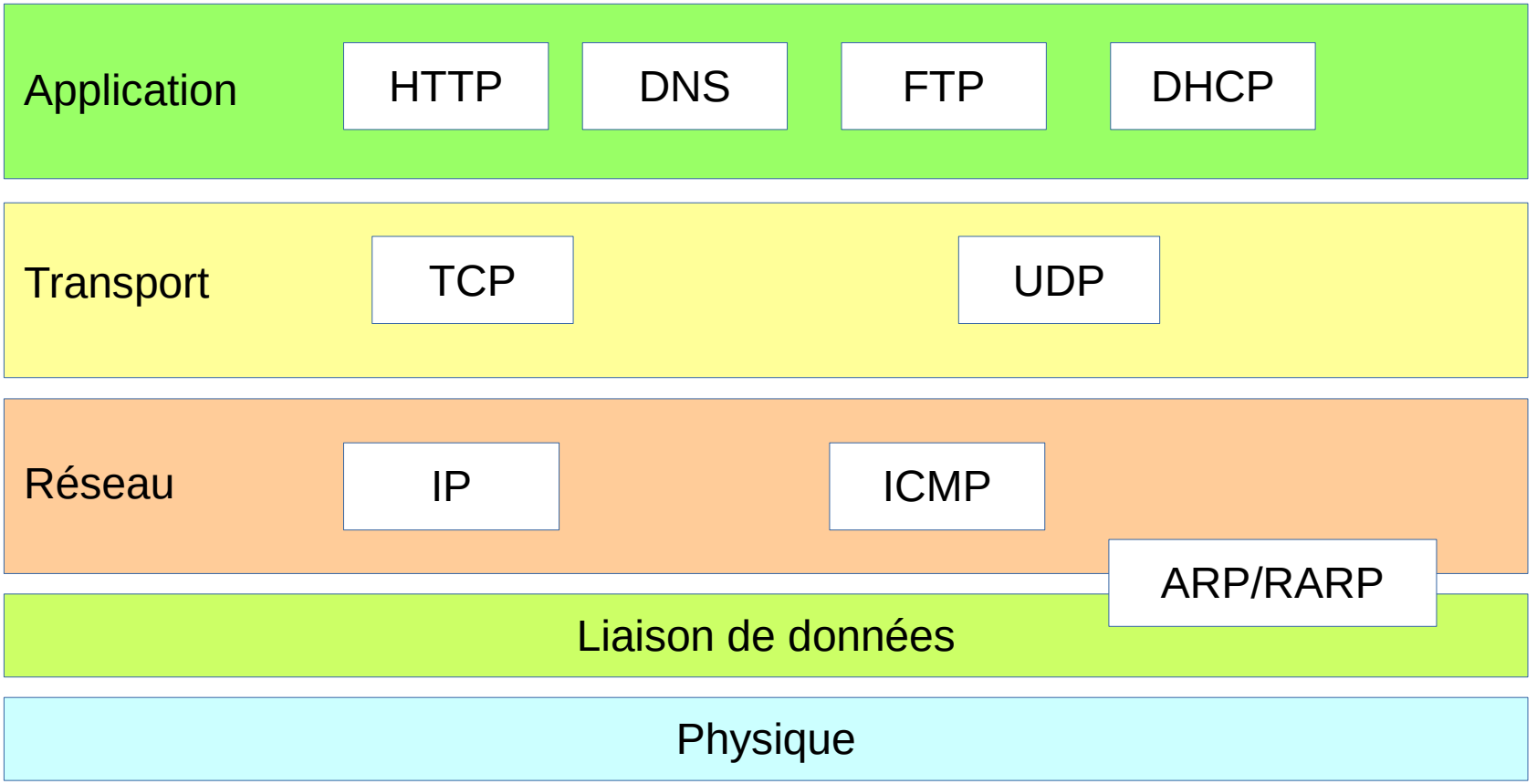
Deux visions s'affrontent

- Le réseau est non fiable
  - Les stations doivent assurer elles-mêmes la fiabilité (contrôle d'erreur et contrôle de flux)
    - *Mode non connecté basé sur l'échange de paquets*
- Le réseau doit être fiable
  - Établissement d'une connexion avant la communication

Exemples:

- Mode connecté: norme X25
- Mode sans connexion: protocole IP

# Architecture TCP/IP



# Le protocole IP

IP : *Internet Protocol*

- Issu des travaux d'ARPANET
- Standardisé dans le RFC 791 en 1981
- Contraintes de conception
  - La panne d'un équipement ne doit pas entraîner la paralysie du réseau
  - Privilégier la disponibilité du réseau
    - *Tant que l'émetteur et le destinataire sont disponibles la communication doit être assurée même si des routeurs intermédiaires subissent des pannes*
  - Définir une architecture souple permettant aussi bien l'échange de fichier que la transmission de voix en temps réel
- C'est un protocole qui opère par routage de paquets
- L'intelligence et le contrôle du réseau sont laissés au protocoles de transport
- Très faible qualité de service (*Best Effort*) et peu adapté aux usages multimédia

# Le protocole IP

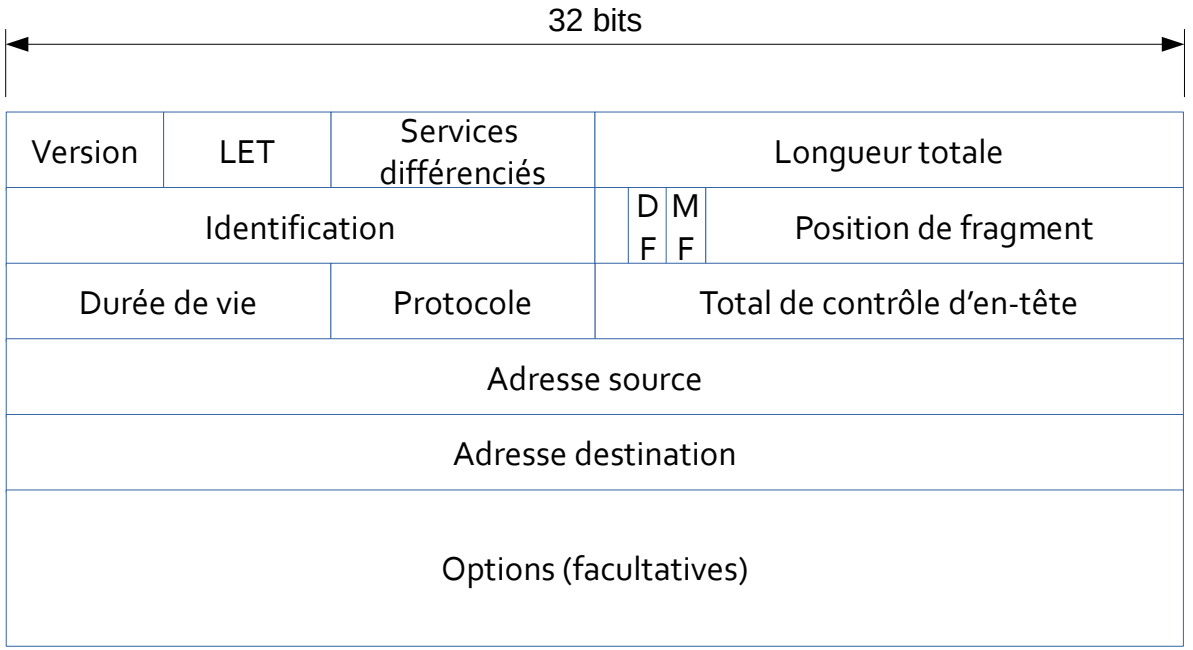
- Qu'est ce que le *Best Effort* ?
  - Mode non fiable et non connecté
  - Pertes de paquets/datagrammes possibles
  - Déséquencements possibles dus au routage

- IP est le protocole le plus largement déployé au niveau mondial avec deux versions qui cohabitent: IPv4, IPv6
  - Fonctionnalités d'IPv4
    - Adressage
    - Routage
    - Gestion de la fragmentation

  - Il s'appuie également sur ICMP pour vérifier le fonctionnement du réseau
  - Il définit un standard d'ordonnancement des données (*Network Byte Order*) qui consiste à envoyer l'octet de poids le plus fort d'un paquet en premier

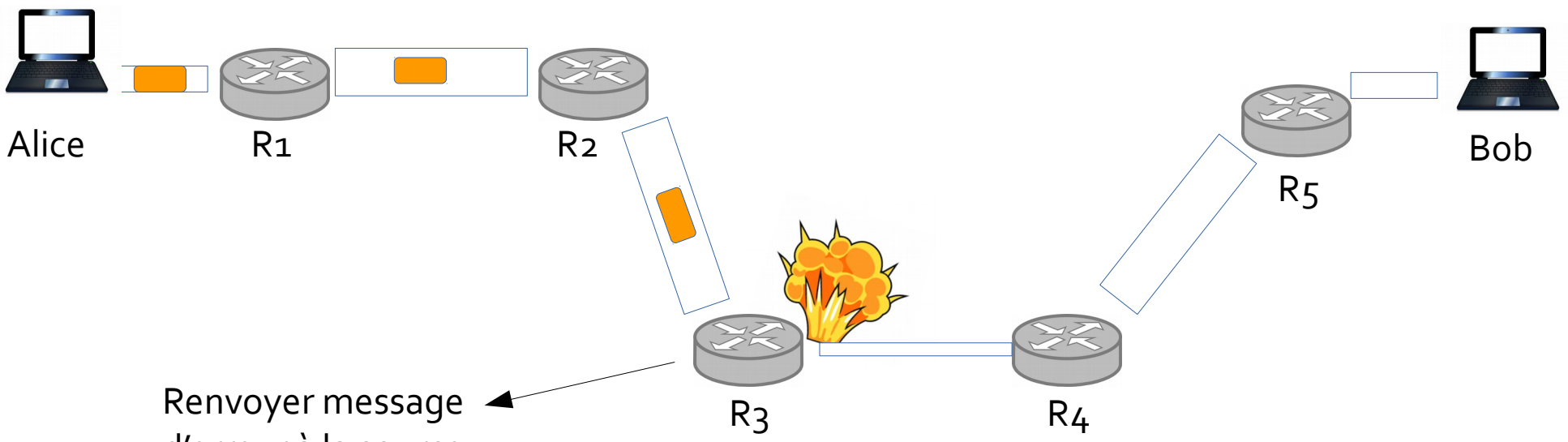
# Le protocole IP

- Un paquet IP = en-tête IP + charge utile
  - Taille maximale : 65 535 octets
  - Structure d'en-tête :




- Version
  - v4 vs v6
- Longueur en-tête
  - Mots de 32 bits
  - Min = ___ ; Max = 15
- Longueur totale
  - Octets
- Fragmentation
  - Fragments d'un même segment
  - *Don't Fragment* (DF)
  - *More Fragments* (MF)
  - Position de fragment
- Durée de vie
  - Nombre de routeurs traversés (sauts)
  - *Time to Live* ou TTL
- Protocole
  - TCP, UDP…
- *Checksum*
- Adresses

Mémo fragmentation




Renvoyer message d'erreur à la source

La source décide de fragmenter et recommencer ou non (découverte de MTU)

Fragmenter le paquet

- Surcharge pour le routeur sur action hors profil
- Droit d'ouvrir un paquet qui ne lui est pas destiné (*liability)*

# L'adressage IP v4

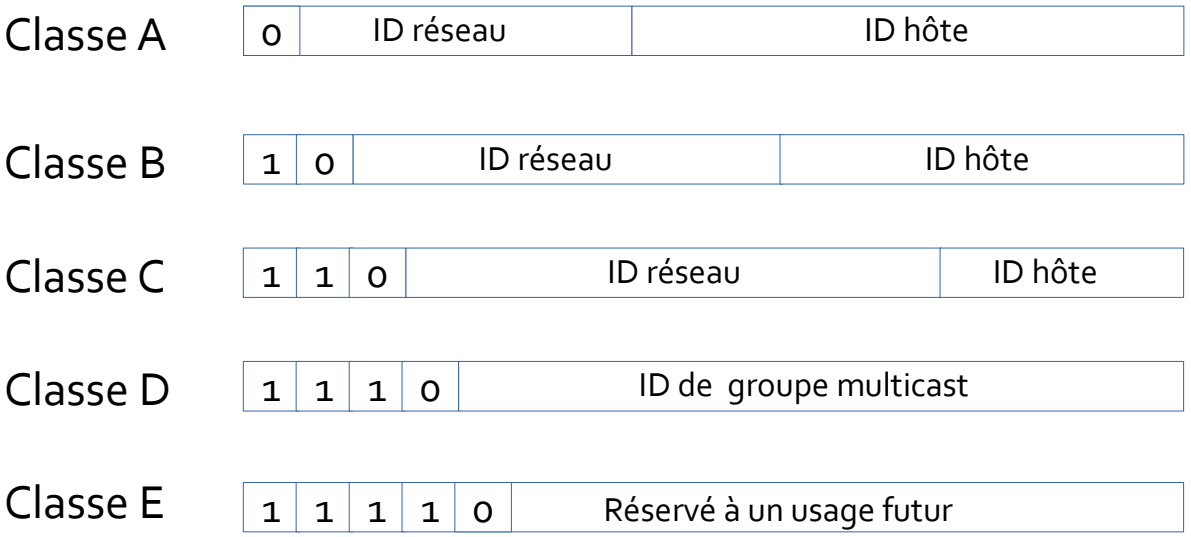
- Chaque interface du réseau dispose d'une adresse unique appelée adresse IP qui se compose
  - D'un identifiant réseau
  - D'un identifiant machine

- 1 adresse réseau pour une *Network Interface Card* (NIC)

- Codée sur 32bits
  - Représentée sous la forme de 4 entiers variant entre 0 et 255 séparés par des points

- Découpée en 5 classes d'adresses différentes
  - Les premiers bits du premier octet permettent de déterminer la classe
  - La classe va donner la taille de l'identifiant réseau et la taille de l'identifiant machine

# L'adressage IP v4 : les classes d'adresses




| Classe | | Plage d'adresses |
|---|---|---|
| A | | **0**.0.0.0 à **127**.255.255.255 |
| B | | **128**.0.0.0 à **191**.255.255.255 |
| C | | **192**.0.0.0 à **223**.255.255.255 |
| D | | **224**.0.0.0 à **239**.255.255.255 |
| E | | **240**.0.0.0 à **247**.255.255.255 |

# L'adressage IP v4

Certaines adresses IPv4 sont particulières

- Envoi de messages multi-destinataires
  - 255.255.255.255 désigne toutes les machines du réseau local (*limited broadcast*)
  - Lorsque la partie hôte a tous ses bits à 1: *directed broadcast,* vers un LAN à travers le routeur
  - Généralement, les routeurs sont configurés pour ne pas laisser passer le *broadcast*

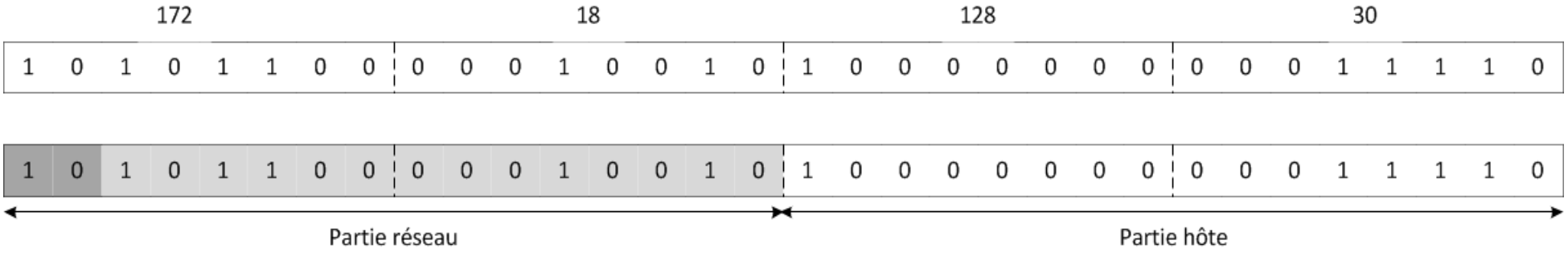
- Désigne la machine courante
  - 127.X.Y.Z pour effectuer des tests inter-processus et des tests réseaux en local
  - 0.0.0.0 désigne l'ordinateur lui-même (utilisé lors de la phase d'amorçage)

- Réseau courant
  - Partie hôte à 0
  - Adresse du réseau

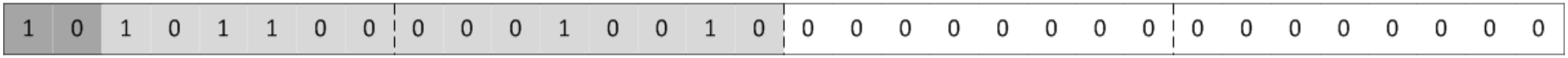
- Certaines adresses de classe A, B et C sont réservées à la constitution de réseaux privés
  - Classe A: 10.0.0.0 à 10.255.255.255
  - Classe B: 172.16.0.0 à 172.31.255.255
  - Classe C: 192.168.0.0 à 192.168.255.255

# L'adressage IP v4 : exemple

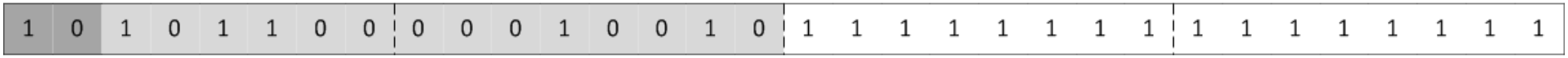
Exemple : 172.18.128.30




Adresse de réseau




Adresse de diffusion



# Correspondance adresse IP et adresse MAC

- La correspondance entre les adresses IP et les adresses MAC se fait à l'aide du protocole ARP (*Address Resolution Protocol*)
  - Il utilise la propriété de diffusion au niveau liaison de données

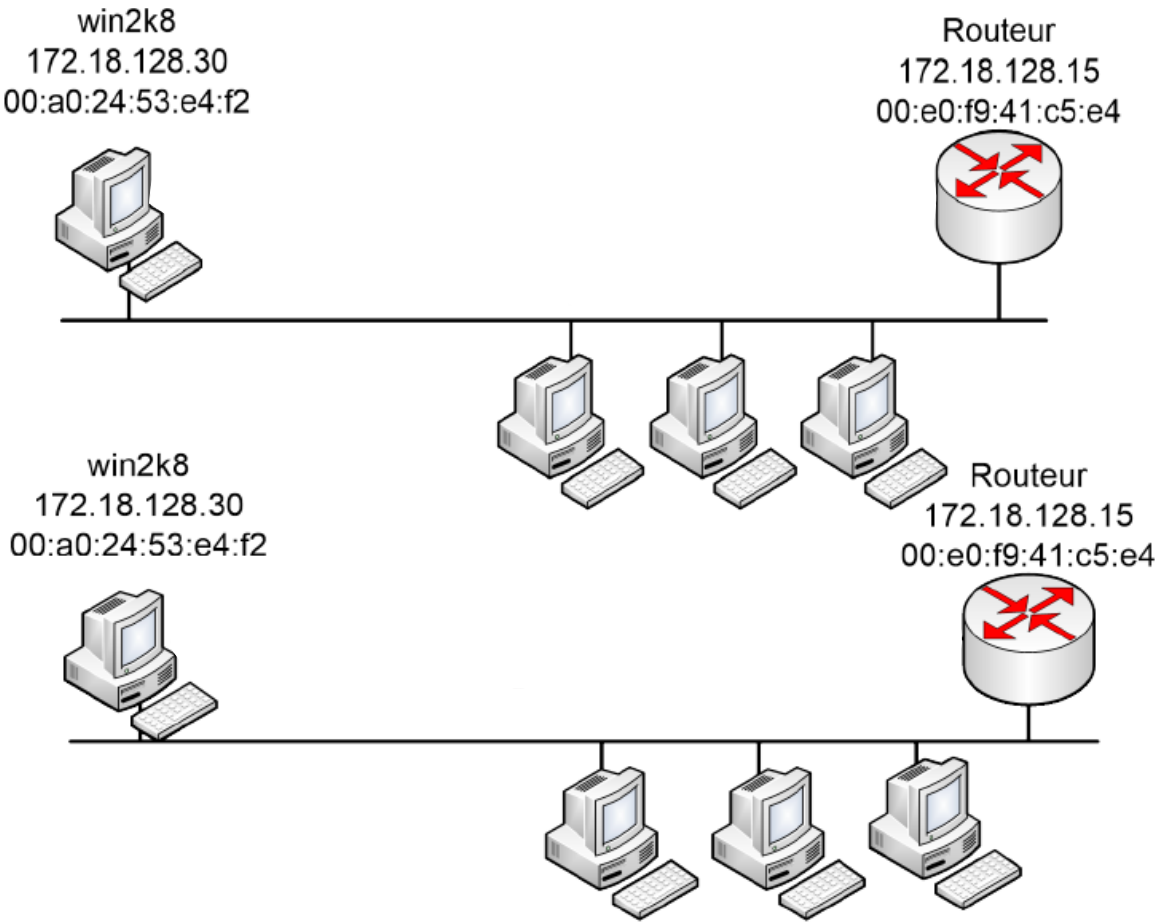
- Une table ARP est maintenue à l'interface entre les couches liaison de données et réseau de chaque machine
  - Lorsqu'un paquet est émis sur le réseau, la couche liaison consulte la table ARP
    - Si l'adresse MAC correspondant à l'adresse IP du destinataire est présente dans la table, le paquet IP est encapsulé dans la trame Ethernet qui sera acheminée au destinataire
    - Sinon, une requête ARP est construite et est diffusée sur le réseau

  - A la réception d'une requête ARP, la table locale est mise à jour avec la correspondance @MAC ↔ @IP de l' émetteur de la requête
    - Si la requête nous est destinée (i.e. si notre couche réseau possède bien l'@IP demandée), on répond à destination de l'émetteur de la requête
    - Sinon on ne fait rien

# Illustration du fonctionnement d'ARP

Requête

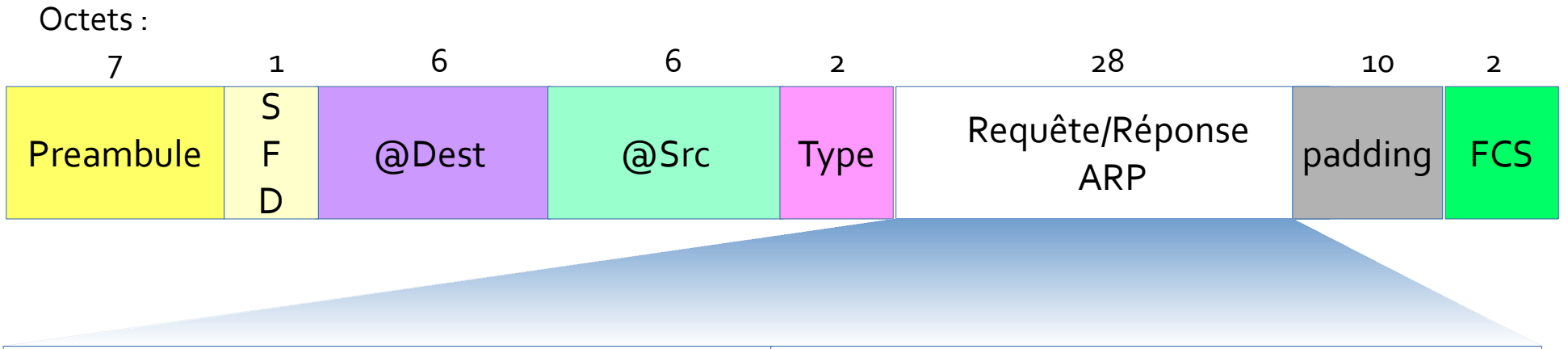
Réponse




Médium

- Hub => ?
- Switch => ?

# ARP : format de trame



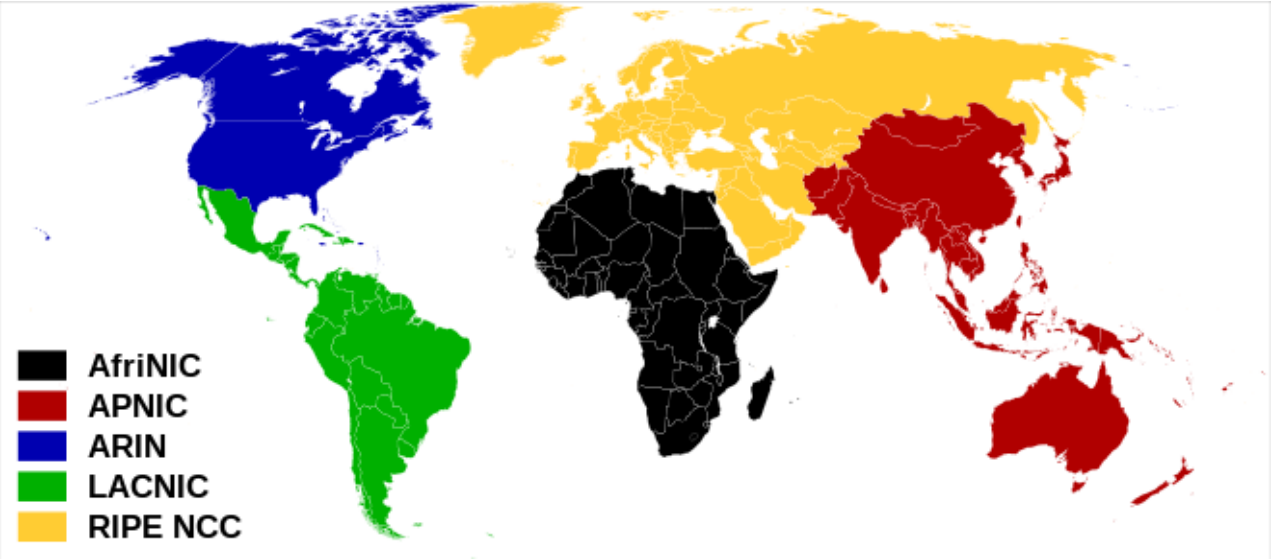

| | |
|---|---|
| *headerHardware type (2 bytes) (1 for Ethernet)* | *Protocol type (2 bytes) (0x800 for IP)* |
| *Hardware address length (1 byte)* / *Protocol address length (1 byte)* | *Operation code (2 bytes) (Request=1, Reply=2)* |
| *Source hardware address (6bytes for Ethernet)* | |
| *Source protocol address (4bytes for IPv4)* | |
| *Target hardware address (6bytes for Ethernet)* | |
| *Target protocol address (4bytes for IPv4)* | |

**Padding ???**

# *Internet Corporation for Assigned Names and Numbers (ICANN)*

- ICANN (anciennement l'IANA) est l'organisme en charge de la distribution des @IP
- Avec l'introduction de la norme CIDR, les adresses ont été découpées en blocs
- Les blocs sont composés d'@IP contiguës

- L'ICANN délègue la gestion et l'allocation à 5 Registres Internet Régionaux (RIR)



## *Internet Corporation for Assigned Names and Numbers*

- Les RIR assignent ensuite les blocs à des registres nationaux (NIR) qui les assignent eux-mêmes à des réseaux locaux (LIR) et des FAI En France
  - 2011 : 355 LIR

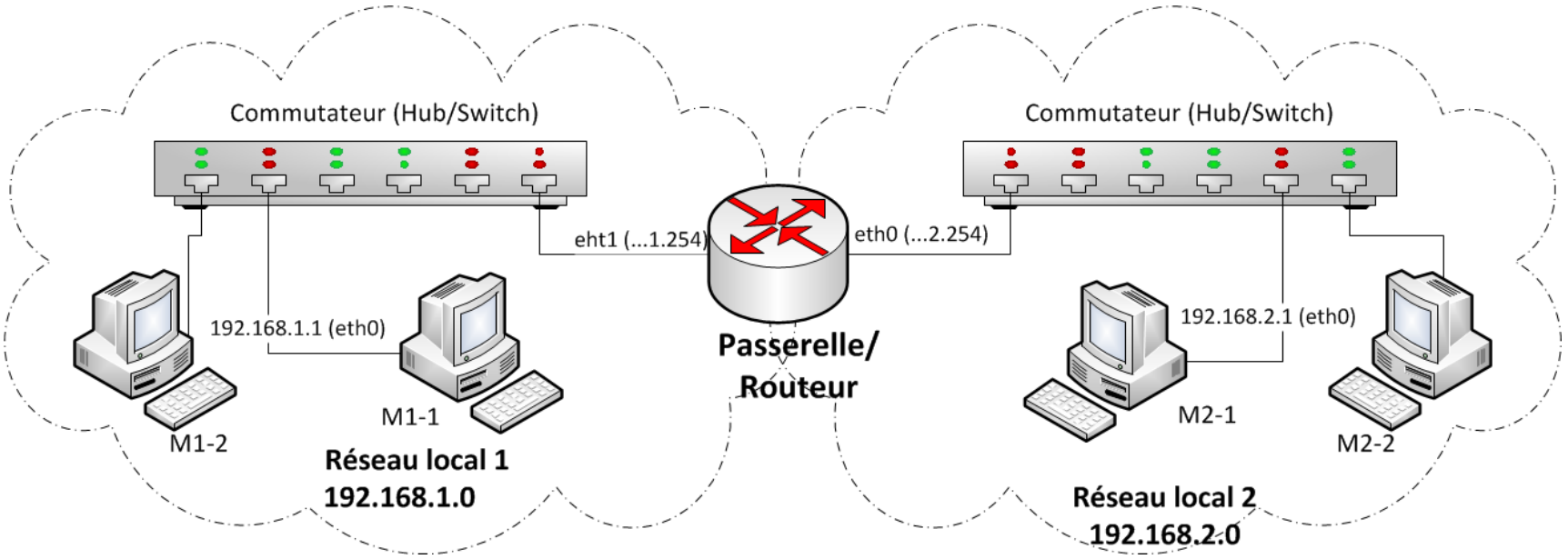
  - Le 3 Février 2011, l'ICANN a distribué les derniers blocs disponibles
  - Le 15 Avril 2011 l'APNIC (Asie) a annoncé qu'il ne disposait plus que d'un bloc /8
  - Le 14 Septembre 2012 idem pour le RIPE NCC
- Longtemps repoussée, la transition vers IPv6 devient inéluctable!
  - Épuisement du *pool* d'adresses IP
  - Arrivée de réseaux très large échelle
    - *Wireless Sensor Networks*
      - Réseaux de capteurs sans fil communicant à travers une *gateway*
    - *Internet of Things*
      - Réseaux d'objets directement adressables par Internet

# Routage et interconnexion de réseaux

# Routeur IP

Un routeur IP est constitué de plusieurs interfaces réseau

- En mode passerelle, le routeur a une adresse IP par interface physique, chacune appartenant à un réseau IP différent




- En mode pont, le routeur possède deux interfaces sur le même réseau IP
  - Permet l'interconnexion de plusieurs réseaux de technologies couches 1 et 2 différentes pour constituer le même réseau IP

# Les tables de routage

- Chaque station d'un réseau IP va maintenir des tables de routage lui permettant de joindre n'importe quelle destination
  - Si la destination est dans le même réseau, on parle de remise directe
    - Même réseau : <id.reseau> des adresses sources et destinations sont identiques
  - On parle de remise indirecte dans le cas contraire

- La table de routage se composait historiquement de deux informations
  - L'adresse du réseau destination
  - L'adresse du routeur suivant (adresse de prochain saut)
  - Dans le cas d'une remise directe, cette adresse est positionnée à 0.0.0.0

  - Exemple:

| Destination | Routeur |
|---|---|
| 172.18.0.0 | 0.0.0.0 |
| 138.5.0.0 | 172.18.128.254 |

    - La route par défaut est notée 0.0.0.0

# Les tables de routage

Exemple

Netstat -rn sur Ubuntu

Kernel IP routing table

| Destination | Gateway | Genmask | Flags | MSS | Window | irtt | Iface |
|---|---|---|---|---|---|---|---|
| 192.168.1.0 | 0.0.0.0 | 255.255.255.0 | U | 0 | 0 | 0 | eth0 |
| 169.254.0.0 | 0.0.0.0 | 255.255.0.0 | U | 0 | 0 | 0 | eth0 |
| 127.0.0.0 | 0.0.0.0 | 255.0.0.0 | U | 0 | 0 | 0 | lo |
| 0.0.0.0 | 192.168.1.1 | 0.0.0.0 | UG | 0 | 0 | 0 | eth0 |

Quelle est ~~la machine~~ le réseau à viser ?

Par quelle NIC dois-je passer ?

# Les tables de routage

Exemple :
Netstat -r sur Windows

Est-ce là que je veux aller ?

IPv4 Table de routage
===========================================================================
Itinéraires actifs :

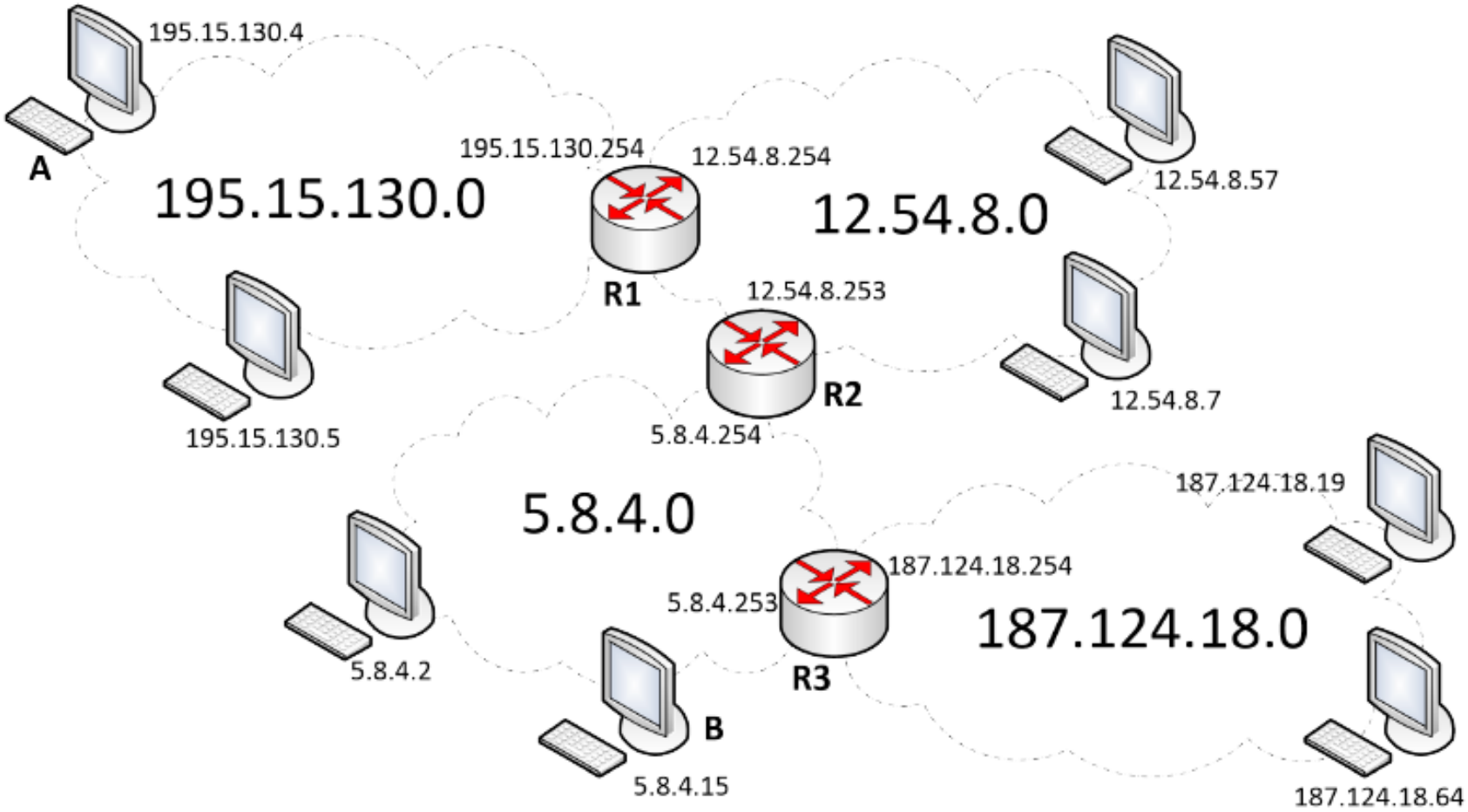
| Destination réseau | Masque réseau | Adr. passerelle | Adr. Interface | Métrique |
|---|---|---|---|---|
| 0.0.0.0 | 0.0.0.0 | 192.168.1.254 | 192.168.1.57 | 25 |
| 127.0.0.0 | 255.0.0.0 | On-link | 127.0.0.1 | 306 |
| 192.168.1.0 | 255.255.255.0 | On-link | 192.168.1.57 | 281 |

===========================================================================
Itinéraires persistants :
Aucun

Je passe par
quel intermédiaire ?

Je passe par quelle interface locale ?
IP => MAC addr

# Exemple de réseau interconnecté



# Les tables de routage

Host A

| Destination | Gateway |
|---|---|
| 195.15.130.0 | 0.0.0.0 |
| 12.54.8.0 | 195.15.130.254 |
| 5.8.4.0 | 195.15.130.254 |
| 187.124.18.0 | 195.15.130.254 |

R1

| Destination | Gateway |
|---|---|
| 195.15.130.0 | 0.0.0.0 |
| 12.54.8.0 | 0.0.0.0 |
| 5.8.4.0 | 12.54.8.253 |
| 187.124.18.0 | 12.54.8.253 |

Routes par défaut

R2

| Destination | Gateway |
|---|---|
| 195.15.130.0 | 12.54.8.254 |
| 12.54.8.0 | 0.0.0.0 |
| 5.8.4.0 | 0.0.0.0 |
| 187.124.18.0 | 5.8.4.253 |

R3

| Destination | Gateway |
|---|---|
| 195.15.130.0 | 5.8.4.254 |
| 12.54.8.0 | 5.8.4.254 |
| 5.8.4.0 | 0.0.0.0 |
| 187.124.18.0 | 0.0.0.0 |

# Les tables de routage

Host A

| Destination | Gateway |
|---|---|
| 195.15.130.0 | 0.0.0.0 |
| 0.0.0.0 | 195.15.130.254 |

R1

| Destination | Gateway |
|---|---|
| 195.15.130.0 | 0.0.0.0 |
| 12.54.8.0 | 0.0.0.0 |
| 0.0.0.0 | 12.54.8.253 |

R2

| Destination | Gateway |
|---|---|
| 195.15.130.0 | 12.54.8.254 |
| 12.54.8.0 | 0.0.0.0 |
| 5.8.4.0 | 0.0.0.0 |
| 187.124.18.0 | 5.8.4.253 |

Réduction de la taille

R3

| Destination | Gateway |
|---|---|
| 0.0.0.0 | 5.8.4.254 |
| 5.8.4.0 | 0.0.0.0 |
| 187.124.18.0 | 0.0.0.0 |

# Les tables de routage

- Problèmes de cette solution
  - Les tables de routage deviennent vite très importantes
    - Plus la taille augmente plus le temps pour traiter un paquet est long
  - Le nombre de réseaux qu'il est possible d'interconnecter est limité

- Supposons qu'un administrateur dispose de plusieurs réseaux physiques différents
  - Avec la solution basée sur les classes, il doit acheter une adresse par sous réseau physique s'il souhaite les différencier
  - Pbm: cela revient très cher et on gaspille de nombreuses adresses

# Sous-réseaux

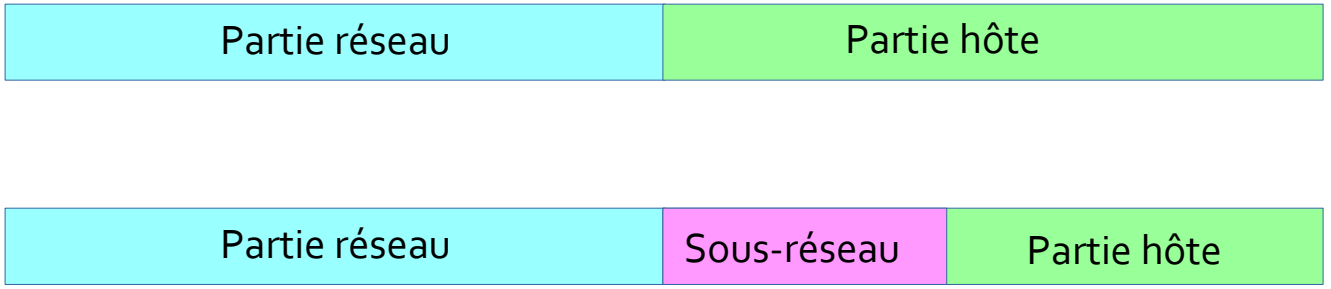
- Une solution consiste donc à partitionner un réseau en plusieurs sous-réseaux
  - De l'extérieur le réseau apparaît comme un seul réseau cohérent
  - En interne, le réseau est découpé en sous-réseaux permettant un contrôle plus fin des échanges

- L'organisation interne n'a pas à être connue de tout le monde
  - Cartographie
  - Identification des rôles des machines, de leurs failles…

- Les routeurs d'Internet ont juste besoin de connaître le routeur d'entrée sur le réseau
  - A charge pour ce dernier d'être capable de diriger les paquets vers le bon sous-réseau

# Sous-réseaux

- Mise en œuvre
  - On définit un sous réseau en découpant l'identificateur machine en deux parties:
    - Un identifiant réseau
    - Une partie locale




La taille de l'identifiant de sous-réseau va dépendre du nombre de sous-réseaux désirés

- Si l'on souhaite avoir k sous-réseaux, on doit réserver n bits de la partie locale tel que

$$k \leq (2^n)$$ †

# Sous-réseaux

- Le choix du découpage dépend des perspectives d'évolution du site
  - Exemple Classe B (139.54.0.0)
    - 8 bits pour les parties sous-réseau et 8 bits pour la partie machine donnent un potentiel de 256 sous-réseaux et 254 machines par sous-réseau
      - *De 139.54.0.0 à 139.54.255.0*
    - 3 bits pour la partie sous-réseau et 13 bits pour le champ machine permettent 8 sous-réseaux de 8190 machines
  - Exemple Classe C :
    - 4 bits pour la partie réseau et 4 bits pour le champ machine permettent ___ réseaux de ___ machines
- Lorsque le sous-adressage est défini, toutes les machines du réseau doivent s'y conformer sous peine de dysfonctionnement
  - Comment connaître le découpage effectué ?

# Sous-réseaux

- Utilisation d'un masque de 32 bits avec les bits du masque (*subnet mask*) :
  - positionnés à 1 : partie réseau et sous-réseau,
  - positionnés à 0 : partie machine

- Exemple de masques de sous réseau
  - Adresse de classe B avec 1 octet sous-réseau et 1 octet machine
  - Ex: 11111111 11111111 11111111 00000000
  - La notation classique utilisée est la notation décimale pointée 255.255.255.0

# Masques de sous-réseaux et routage

- Lorsqu'un message arrive de l'extérieur vers un réseau découpé en sous-réseaux, le routeur d'accès doit tenir compte des masques de sous-réseaux pour router correctement en interne
  - Le routeur doit déterminer l'appartenance d'une destination à un sous-réseau et non plus simplement à un réseau
  - Le routeur effectue un ET logique entre l'adresse de destination et le masque pour déterminer l'adresse de sous-réseau

- Quelques masques par défaut sont utilisables en fonction de la classe de l'adresse réseau
  - 255.0.0.0 pour les réseaux de classe A
  - 255.255.0.0 pour les réseaux de classe B
  - 255.255.255.0 pour les réseaux de classe C

# Masques de sous-réseaux et routage

- Avec l'introduction des masques, la structure des tables de routage maintenues par chaque nœud du réseau doit être modifiée
  - Ajout d'une troisième donnée: le masque
  - Exemple :
  - Table de routage du routeur d'accès à Internet d'une entreprise ayant reçu le bloc d'adresses 172.18.0.0/16

| Destination | Masque | Prochain saut |
|---|---|---|
| 172.18.0.0 | 255.255.128.0 | 0.0.0.0 |
| 172.18.128.0 | 255.255.192.0 | 0.0.0.0 |
| 172.18.192.0 | 255.255.192.0 | 0.0.0.0 |
| 0.0.0.0 | 0.0.0.0 | 138.5.1.254 |

*Exos time !!* 

# Notation CIDR

# Pénurie d'adresses IP v4

- Lors de la conception d'IP à l'époque d'ARPANET, les classes ont été pensées pour répondre à l'ensemble des besoins de l'époque
  - Connecter les universités à travers le monde
  - Relier quelques entreprises et quelques sites gouvernementaux
- Les adresses de classes B devaient suffire pour répondre à ces besoins
  - 16383 réseaux possibles soit bien plus que le nombre d'universités dans le monde
- La distribution d'adresses a été très généreuse au début
  - Des entreprises ou universités se sont vue allouées des adresses de classe A ou de classe B là où des classes C auraient été suffisantes
- Problèmes
  - Le réseau est devenu mondial et en 1996 il y avait déjà 100,000 réseaux connectés à Internet
  - La moitié des réseaux de classe B ne contenaient qu'une cinquantaine d'hôtes …

# Pénurie d'adresses IP v4

- Les adresses de classe B ont été épuisées en 1993
  - C'est pourtant la classe la plus utile pour une entreprise
    - La classe C est trop petite et limite l'évolutivité
    - La classe A est trop grande
- La solution serait donc de donner 256 adresses de classe C
  - Le problème c'est qu'il faut alors 256 entrées dans les routeurs
- Pour résoudre ce problème d'explosion du nombre d'entrées, la solution adoptée est le sur-adressage ou adressage hors classe (*Classless Inter-Domain Routing*, CIDR)
  - Les adresses attribuées doivent se suivre
  - Elles débutent à une puissance de 2 et forment un bloc de puissances de 2

# CIDR et routage

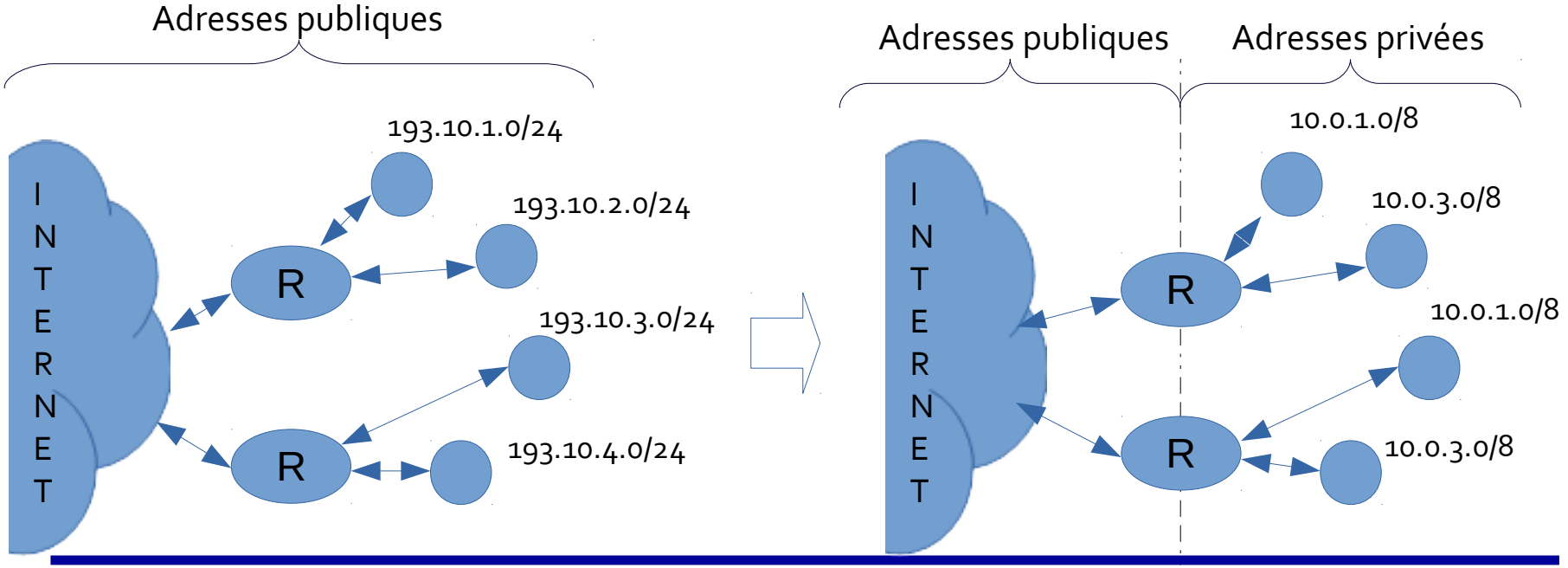
- Avec le sur-adressage, les routeurs ne peuvent plus décider à partir des classes d'adresse
- Le masque devient essentiel pour déterminer les routes
  - On adopte une nouvelle notation /n appelée préfixe
    - *n* représente le nombre de bits à 1 du masque (en partant de la gauche)
    - Les masques à trous sont interdits !

| Destination/préfixe | routeur |
|---|---|
| 172.18.128.0/24 | 0.0.0.0 |
| 138.5.0.0/16 | 172.18.128.254 |

- Pour une @dest, plusieurs entrées peuvent maintenant correspondre
  - On prend toujours l'entrée avec le préfixe le plus élevé
- La route par défaut peut s'écrire: 0.0.0.0/0

# *Network Address Translation* (NAT)

- Conjointement à l'utilisation du CIDR, le mécanisme NAT (RFC 2663) a été introduit pour essayer de limiter le nombre de demandes d'adresses IPv4
  - L'idée est de ne distribuer des adresses qu'aux machines qui ont besoin d'être contactées depuis Internet
  - Le reste peut être connecté à l'aide d'adresses privées et n'a pas besoin d'être visible
- L'objectif de NAT est de permettre à un ensemble de stations d'un même réseau privé de partager un même *pool* d'adresses IP publiques



# IP v6

# L'adressage IPv6

L'adressage IPv4 arrivant en fin de vie, une nouvelle solution est nécessaire pour permettre de connecter de nouvelles machines au réseau

IPv6 améliore IPv4 sur de nombreux points

- Augmentation de la taille des adresses
  - 128bits contre 32bits pour IPv4
- Prise en compte du trafic
  - Gestion de la QoS à travers des politiques de files d'attente notamment

- Réduction de la taille des tables de routage
- Simplification de l'entête pour accélérer le routage
- Amélioration de la sécurité (authentification et confientialité)
- Gestion de la mobilité
- Support du multicast de manière native
- …

# L'adressage IPv6

Les adresses sont codées sur 128 bits (16 octets)

- Pour les adresses IPv6, la notation hexadécimale est privilégiée

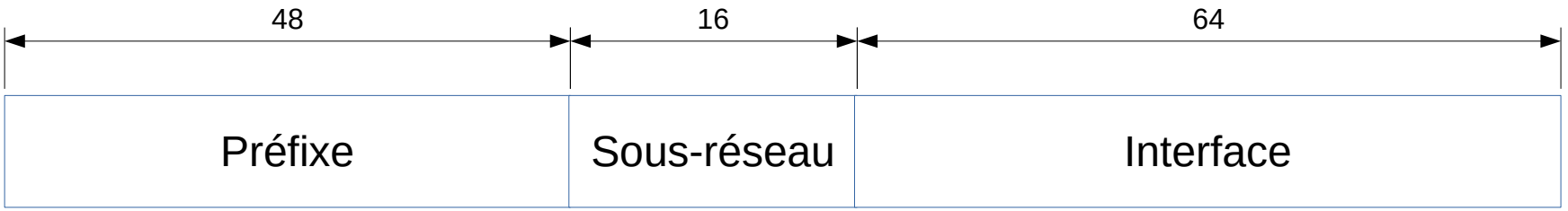
IPv6 supporte trois types d'adresses

- *Unicast*
- *Multicast*
  - Identifie un groupe de NIC
- *Anycast*
  - Le message va au membre du groupe *anycast* le plus proche de l'expéditeur [RFC 1546]
  - L'intérêt est de ne plus avoir besoin de connaître l'adresse IP d'un serveur
  - On effectue une requête anycast vers un service, et le serveur le plus proche répond
  - *Ex: transfert de fichiers hébergés sur plusieurs sites mirroirs*

L'adresse de *broadcast* est supprimée

- Emulée par un groupe *multicast* contenant toutes les stations du réseau

# Ecriture des adresses IPv6




Les adresses IPv6 se représentent sous la forme de groupes de 16bits séparés par le symbole « : » Ex: A8CF:54B3:0000:0000:0000:7CD1:04B1:0812

Quelques conventions supplémentaires

- Lorsqu'un groupe débute par « 0 », on peut omettre les « 0 »

Exemple : 04B1 peut s'écrire: 4B1

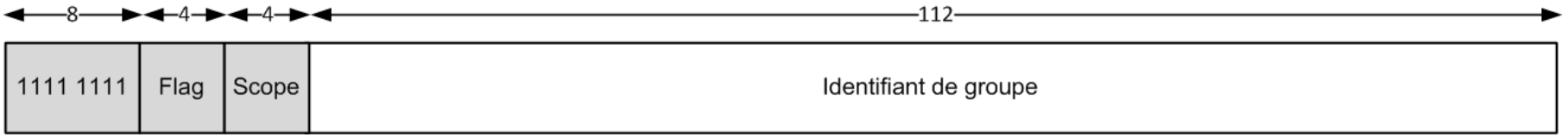
- Si un groupe de 16bits n'est composé que de « 0 », on peut le remplacer par « :: »

Une autre façon d'écrire l'adresse précédente serait A8CF:54B3::7CD1:4B1:812

Toute adresse IPv4 peut se représenter dans la notation IPv6

Ex: 172.18.128.61 en IPv4 peut s'écrire en IPv6 ::172.18.128.61 ou ::AC12:803D

# Adresses IPv6 particulières

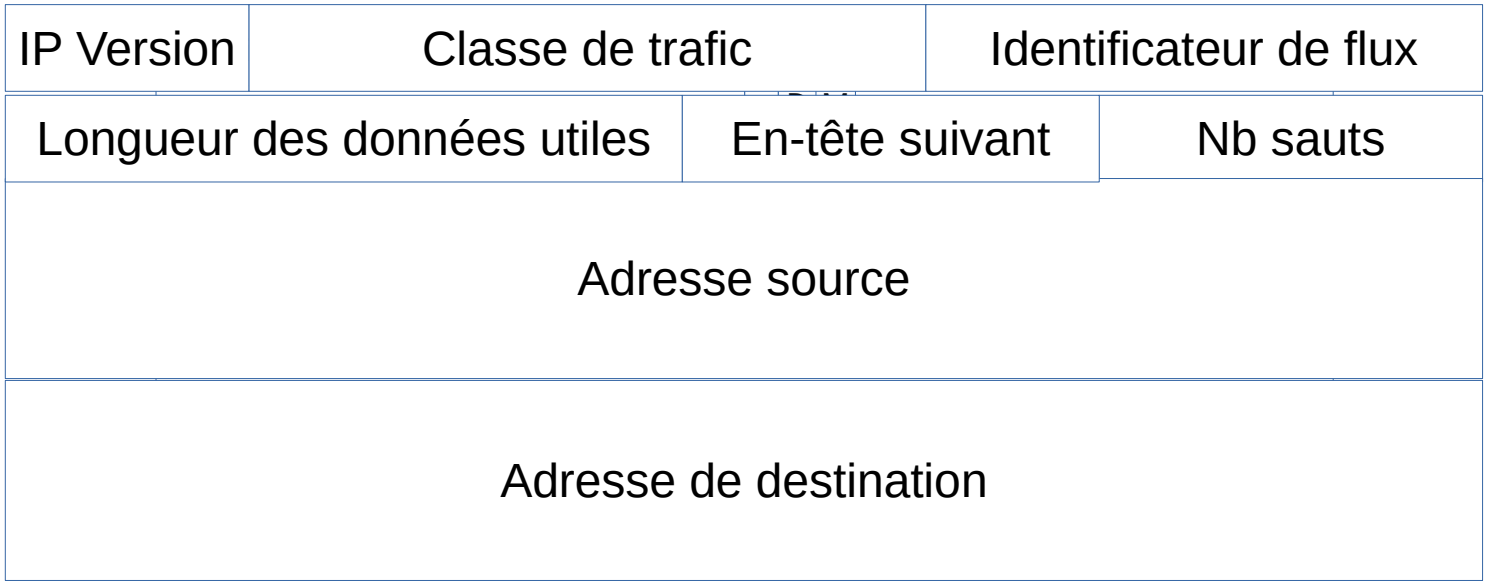



Adresse *multicast* démarre toujours par FF (ou en binaire 1111 1111)

- Le champ *Flag* permet de déterminer entre autre le type d'adresse *multicast* (adresse permanente attribuée par l'IANA ou adresse temporaire)
- Le champ *Scope* définit la portée d'utilisation de l'adresse

| Valeur du *scope* | Description |
|---|---|
| 0,3,F | Réservé |
| 1 | Limité à l'interface (au noeud) |
| 2 | Limité au lien |
| 4 | Limité Admin-local |
| 5 | Limité au site |
| 8 | Limité à l'organisation |
| E | Globale |
| 6,7,9,A,B,C,D | Non-assigné |

# Simplification de l'en-tête




- Bien que plus grand, l'en-tête IPv6 est plus simple à gérer que l'en-tête IPv4
  - Taille d'en-tête fixe (suppression du champ Longueur d'en-tête)
  - Pas de vérification de l'intégrité de l'en-tête
  - Obligation de le faire au niveau transport et au niveau liaison (redondance des contrôles)
- En-têtes d'extension optionnels pour permettre l'évolutivité

# Les en-têtes d'extension

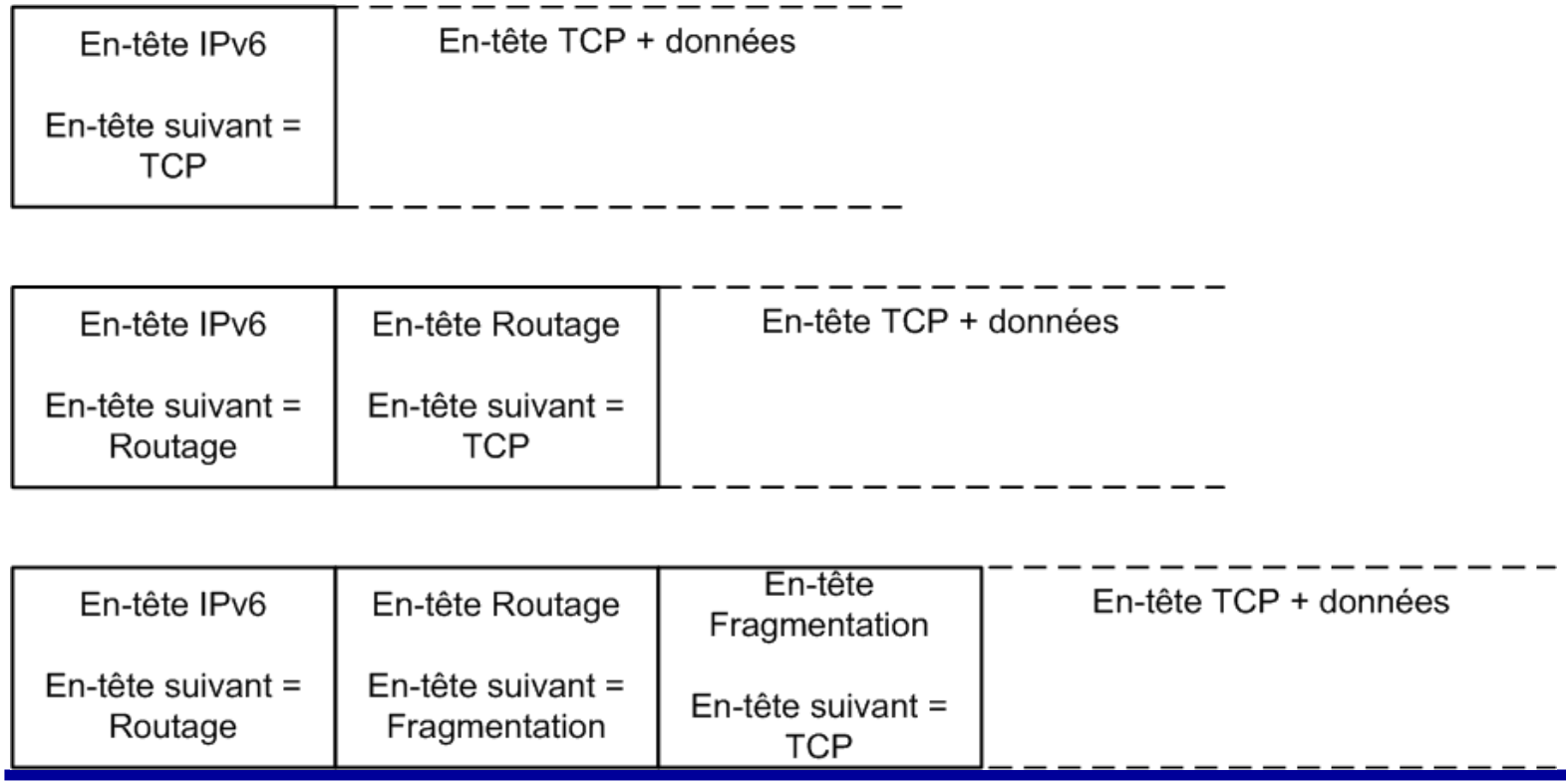
Ces en-têtes permettent d'ajouter des options au paquet classique

Ils n'ont en général pas à être interprétés par les éléments d'interconnexion

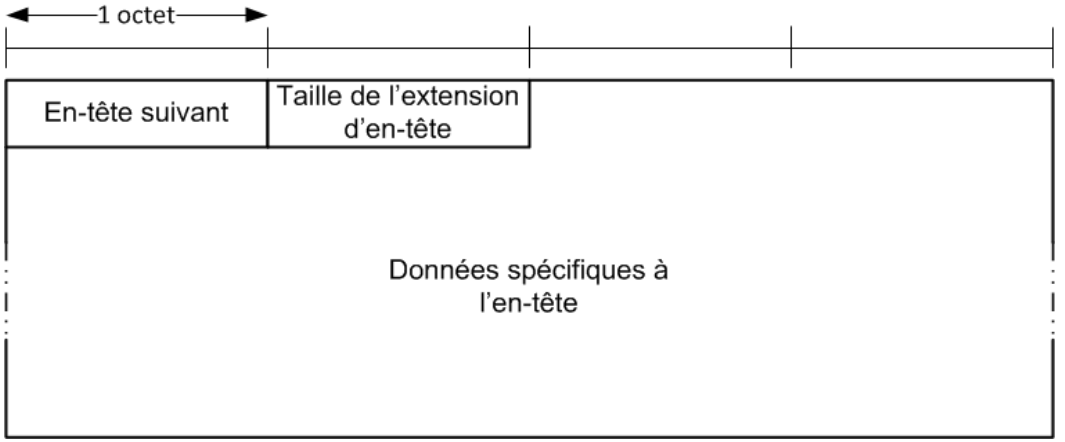
- Exception : l'en-tête *Hop-by-hop* qui contient des informations nécessaires au routage

Ils sont chaînés les uns à la suite des autres

- Chaque en-tête doit indiquer l'en-tête suivant



# Les en-têtes d'extention




Format

- En-tête suivant
- Longueur d'en-tête
- Données spécifiques

En-têtes prédéfinis

- Proche-en-proche (*Hop-by-hop*): examinés par chaque routeur intermédiaire
- Destination: examinés par la destination
- Routage: routage par la source
- Fragmentation
- Authentification [RFC 2402]
- Confidentialité [RFC 2406]

# Gestion de la fragmentation

Lorsqu'un routeur intermédiaire doit gérer la fragmentation (MTU trop faible)

- Coûteux en temps
- Sensible aux erreurs
  - Si un fragment est perdu, il est nécessaire de renvoyer tous les fragments

En IPv6, la fragmentation est gérée au niveau de l'émetteur

- Les routeurs intermédiaires se contentent d'informer l'émetteur que le paquet ne peut être transmis grâce à un message ICMPv6 « *Packet too big* »
- Le MTU (*Maximum Transmission Unit*) est fixé comme IPv4 à 65535 octets
  - Néanmoins il existe une extension pour permettre de supporter des tailles plus importantes jusqu'à $2^{32}$-1 octets ( environ 4,3 Go) [RFC 2675]
  - *On parle alors de jumbogram (jumbo datagram)*
  - *Nécessite d'adapter les protocoles de niveau transport*

# La découverte des voisins

Pour effectuer la découverte de voisins (*Neighbor discovery*), IPv6 s'appuie sur un protocole dédié NDP

Ce protocole reprend les fonctionnalités d'ARP en IPv4 et utilise des messages ICMPv6 pour la signalisation

- Offre des fonctionnalités plus avancées qu'ARP
  - Comme ARP, il effectue la traduction d'adresse entre IP et le niveau MAC
  - Utilise ICMP, ce qui évite de concevoir un nouveau protocole et simplifie son déploiement
  - Repose comme ARP sur des tables d'association au niveau de chaque noeud

- Ajout de la détection d'inaccessibilité d'un noeud: NUD (*Network unreachable detection*)
  - Permet d'effacer les entrées inaccessible de la table de correspondance

# La découverte de voisins

- Permet la configuration automatique des équipements
  - Découvertes des routeurs d'un même réseau physique
  - Découvertes des préfixes d'adresse
    - *A partir d'un préfixe diffusé par les routeurs, le nœud peut construire son adresse en ajoutant à ce préfixe son adresse matérielle (@MAC)*
  - Détection d'adresses dupliquées
  - Découverte des paramètres du réseau
    - *Taille des MTU*
    - *Nbre de saut maximum autorisé*
    - *Utilisation de DHCP ?*
    - *…*

- Permet de notifier des indications de redirections
  - Peut survenir lorsqu'un routeur est informé d'une meilleur route disponible que celle utilisée jusqu'à présent

# La découverte de voisins

Il existe 5 types de messages ICMP pour NDP

- Sollicitation de routeur (RS)
  - Émis en *multicast* (FF02::2) par un équipement au démarrage pour obtenir des informations d'un routeur
  - *Reçu par tous les routeurs du lien physique*

- Annonce de routeur (RA)
  - Émis périodiquement ou en réponse à un RS
  - Donne le préfixe du réseau, le MTU, le nbre de sauts tolérés etc

- Sollicitation de voisins (NS)
  - Pour détecter la duplication d'adresse
  - Pour déterminer l'adresse d'un voisin
  - Pour vérifier qu'un voisin est toujours accessible

- Annonce de voisin (NA)
  - Réponse au NS

- Redirection

# Auto-configuration IPv6

Le processus d'auto-configuration d'IPv6 se découpe en trois étapes

- Établissement d'une adresse lien locale
- Vérification de l'unicité de l'adresse
- Création de l'adresse *unicast* globale

Pour construire cette adresse *unicast* globale, deux solutions sont possibles

- Auto-configuration sans état (*stateless*)
  - Génération de l'adresse à partir du préfixe réseau fourni par un message RA et à partir de l'adresse matérielle de l'interface
  - Ce mode n'existe pas en IPv4

- Auto-configuration avec état (*stateful*)
  - Adresse fournie par un serveur ou un relais DHCPv6

# Transition IPv4 → IPv6

Même si IPv6 est stable depuis plusieurs années et même si les adresses IPv4 sont épuisées, IPv6 ne va pas supplanter IPv4 du jour au lendemain

Il faut prévoir un temps de cohabitation entre les deux protocoles

- Pour permettre la mise à jour des hôtes IPv4 vers IPv6
- Pour modifier/étendre/remplacer les équipements d'interconnexion IPv4 actuels
- Pour permettre aux hôtes IPv4 d'accéder à des services offerts par des hôtes IPv6

Il existe trois techniques de transition

- Cohabitation de deux piles protocolaires
  - Un équipement est capable de communiquer en utilisant aussi bien IPv4 qu'IPv6
  - Nécessité de déterminer les capacités de l'interlocuteur et des nœuds intermédiaires (routeurs)
    - Par exemple, le DNS pourrait retourner l'une ou l'autre adresse pour un serveur supportant les deux piles, en fonction des capacités du nœud appelant
- Encapsulation d'IPv6 dans IPv4
  - Technique dite de tunnel
- Mise en place de traducteur d'en-têtes NAT