

# Cours de réseaux

**M1 Informatique**  
**Faculté Jean Perrin**



# Plan du cours

Partie 1 : Introduction

Partie 2 : Couche physique

Partie 3 : Couche liaison des données

Partie 4 : Couche Réseau / IPv4

Partie 5 : Couche Réseau / Routage

**Partie 6 : Couche Réseau / IPv4, IPv6**

Partie 7 : Couche transport : TCP et UDP

Partie 8 : Couche application

Partie 9 : Couche application / Etude de protocoles

Partie 10 : Notions d'attaques et de sécurité

# Partie 6

## IPv4 / IPv6

# Vous êtes ici

## Modèle OSI

7	Application
6	Présentation
5	Session
4	Transport
3	Réseau ●
2	Liaison
1	Physique

## TCP/IP

<i>Applications</i> <i>Services Internet</i>
<i>Transport (TCP)</i>
<i>Internet (IP)</i> ●
<i>Accès au Réseau</i>

# Dans ce chapitre....

Vous allez faire une terrible découverte.....



# Dans ce chapitre....

Mais tout va devenir plus clair et vous aurez les réponses à toutes vos questions....

Même Hugo !

# Types de trafic

A l'origine, la conception d'IP devait permettre de communiquer selon trois types de trafics.

**Unicast** : communication entre un hôte source unique et un hôte destinataire unique.

**Multicast** : Communication entre un hôte source et un groupe d'hôtes destinataires ayant choisi de recevoir les paquets de cette source.

**Broadcast** : Communication entre un hôte source vers **tous** les hôtes appartenant au même domaine de diffusion.

# Adresses IPv4 : rappels

Formées sur 4 octets

Composée à la fois de la partie réseau (**network id**) et de la partie identification de l'hôte (**host id**).

L'utilisation d'un masque de réseau permet de différencier les deux parties.

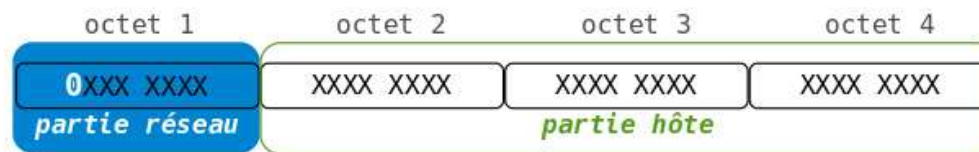
Chaque réseau possède une **adresse de diffusion** : les paquets envoyés à cette adresse sont traités par **tous** les hôtes du réseau.



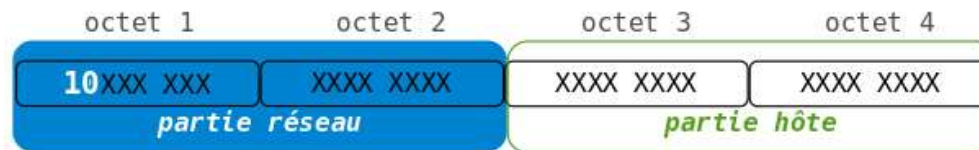
# Classes d'adresses : rappels

A l'origine, création de classes d'adresse

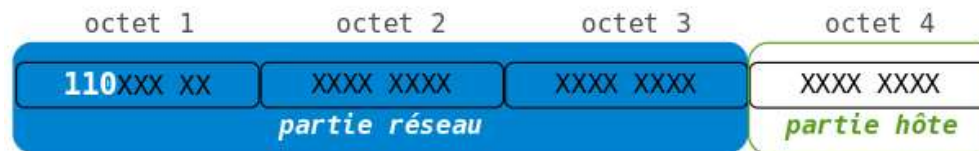
## Classe A



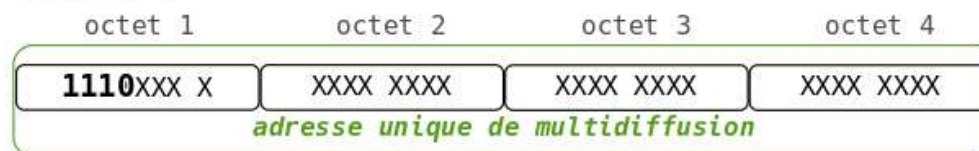
## Classe B



## Classe C



## Classe D



# Classes d'adresses : rappels

L'utilisation de classes d'adresses impliquait un masque réseau par défaut.

Les protocoles de routage de l'époque (RIPv1, IGRP ou Border Gateway Protocol) étaient dits *classful* (pas d'utilisation du masque)

# Classes d'adresses : problèmes

- Mauvaise répartition de l'espace d'adressage.

Exemple : la moitié des adresses disponibles est réservée au 126 réseaux de classe A.

- Gaspillage : à l'époque l'attribution sur simple demande était mal gérée et contrôlée.

Exemple : une grande entreprise avec plusieurs sous-réseaux obtenait plusieurs adresses publiques.

- Pénurie d'adresses
- Difficulté pour administrer les grands réseaux.

# Evolution et solutions

Pour remédier à ces problèmes, plusieurs solutions ont été proposées ces dernières décennies:

- Découpage d'une classe en sous-réseau
- Routage inter-domaine sans classe (CIDR)
- Utilisation de masques à longueur variable (VLSM)
- Réseaux privés et traduction d'adresses (NAT)

# Evolutions et solutions

Au milieu des années 90, l'utilisation des classes d'adresses est définitivement abandonnée au profit de certaines de ces solutions.

Il devient ainsi possible de gérer de manière uniforme la totalité de l'espace d'adressage.

Les protocoles de routage modernes intègrent désormais le masque réseau : ils sont dits ***classless***.

# Découpage d'une classe (ou d'un réseau) en sous-réseaux

# Découpage d'une classe en sous-réseaux

Formalisé en **1985** avec le document RFC 950.

Une classe d'adresses (A,B,C) est découpée en sous-réseaux.

Technique appelée **subnetting**.

On utilise une partie de l'adresse d'hôtes pour étendre l'identifiant réseau.

# Découpage d'une classe en sous-réseaux (subnetting)

## Exemple

Soit un réseau de classe C :

192.168.1.0 (masque 255.255.255.0) 254 hôtes max

Utilisation de 3 bits supplémentaires sur le dernier octet





## Découpage d'une classe en sous-réseaux (subnetting)

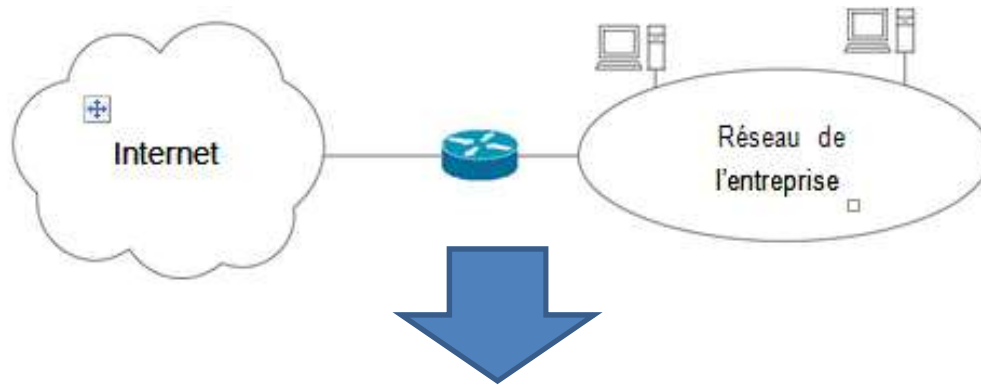
### Exemple (suite)

On dispose désormais de 8 sous-réseaux de 30 hôtes (+ 1 adresse de diffusion)

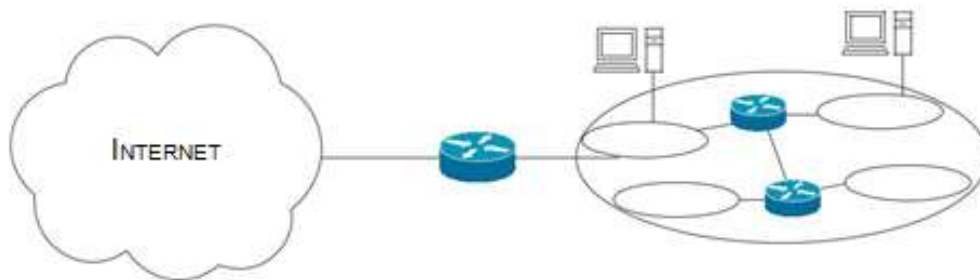
N° sous-réseau	Adresse	Plages d'adresses utilisables	Adresse de diffusion
0	192.168.1.0	192.168.1.1 à 192.168.1.30	192.168.1.31
1	192.168.1.32	192.168.1.33 à 192.168.1.62	192.168.1.63
2	192.168.1.64	192.168.1.65 à 192.168.1.94	192.168.1.95
3	192.168.1.96	192.168.1.97 à 192.168.1.126	192.168.1.127
4	192.168.1.128	192.168.1.129 à 192.168.1.158	192.168.1.159
...	...	.....	....

# Découpage d'une classe en sous-réseaux (subnetting)

Vue de l'extérieur, rien ne change



Les tables de routage d'internet ne changent pas.



Vue de l'intérieur, plein de sous-réseaux

Le routeur d'accès au réseau de l'entreprise doit router vers les « sous-réseaux ».

## Découpage d'une classe en sous-réseaux (subnetting)

### Remarque

Les protocoles de routage de première génération dits « classful » (ex. RIPv1) ne véhiculant pas l'information sur le masque, il pouvait y avoir confusion entre adresse de réseau et adresse de diffusion pour les sous-réseaux ayant leur bit de sous-masques tous à 0 ou à 1.

Exemple : l'adresse de diffusion 192.168.1.255 est la même pour 2 réseaux différents : 192.168.1.0/24 ou 192.168.1.224/27.

La RFC950 préconisait de ne pas utiliser ce type d'adresses de sous-réseaux.

# Découpage d'une classe en sous-réseaux (subnetting)

## Avantages du subnetting

**Répartition de l'administration** : il devient possible de déléguer l'administration des sous-réseaux à des administrateurs s'occupant alors de réseaux plus simples.

**Maintenance facilité** : il est plus facile de corriger ou analyser des problèmes sur des réseaux comportant un nombre restreint de machines.

**Performances accrues** : trafic moins important par sous-réseaux, bande passante mieux gérée (ex. cas du broadcast).

**Le subnetting a été une réponse intéressante pour la gestion des grands réseaux (classe A et B) mais n'a pas permis de « récupérer » des adresses perdues à cause du découpage en classes.**

## Découpage d'une classe en sous-réseaux (subnetting)

### 3 manières de « subnetter » un réseau

- En connaissant le nombre de sous-réseaux désiré.
- En connaissant le nombre de hosts par sous-réseaux.
- En mixant les deux, *i.e* en prenant en compte les deux paramètres.

## Découpage d'une classe en sous-réseaux (subnetting)

### « Subnetter » en connaissant le nombre de sous-réseaux

Il suffit de déterminer le nombre  $n$  de bits nécessaires pour créer  $s$  sous-réseaux ( $s = 2^n - 1$ ).

**Pourquoi « -1 » ?** Pour des raisons de compatibilité historique, on considère que l'octet définissant le sous-réseau ne peut être supérieur ou égal à l'octet utilisé dans le masque pour définir le sous-réseau.

**Exemple :** si le réseau est 179.168.0.0 et qu'on applique un masque personnalisé 255.255.240.0 : on ne définira pas de sous-réseau 179.168.240.0

## Découpage d'une classe en sous-réseaux (subnetting)

### « Subnetter » en connaissant le nombre de sous-réseaux

**Exemple** : soit le réseau 179.168.0.0/16. On désire créer 10 sous-réseaux ( $s = 10$ )

On sait que  $2^3 - 1 = 7$  et que  $2^4 - 1 = 15$ . On a donc besoin de 4 bits pour créer le sous-réseau. Le masque de sous-réseau sera 255.255.240.0 (rappel  $240 = 11110000b$ )

Num	Adresse sous-réseau	1 <sup>er</sup> adresse IP d'hôte	Dernière adresse IP d'hôte
1	179.168.0.0	179.168.0.1	179.168.15.254
2	179.168.16.0	179.168.16.1	179.168.31.254
10	179.168.144.0	179.168.144.1	179.168.159.254

## Découpage d'une classe en sous-réseaux (subnetting)

### « Subnetter » en connaissant le nombre d'hôtes par sous-réseaux

Il suffit de déterminer le nombre  $n$  de bits nécessaires pour avoir  $h$  hôtes avec ( $h = 2^n - 2$ ).

On retranche deux pour réserver deux adresses particulières :

- celle identifiant le sous-réseau
- celle affectée au broadcast



## Découpage d'une classe en sous-réseaux (subnetting)

### « Subnetter » en connaissant le nombre d'hôtes par sous-réseaux

**Exemple** : soit le réseau 179.168.0.0/16. On désire créer des sous-réseaux de 1030 hôtes ( $h=1030$ )

On sait que  $2^{10} - 2 = 1022$  et que  $2^{11} - 2 = 2046$ . On a donc besoin de 11 bits pour identifier les hôtes. Le masque de sous-réseau utilisera donc  $32 - 11 = 21$  bits. Il aura pour valeur 255.255.248.0 (rappel  $248 = 11111000b$ )

**Remarque** : dans chaque sous-réseau, un certain nombre d'adresses ( $2046 - 1030 = 1016$ ) ne sera pas utilisé. Elles permettront d'accueillir de nouvelles machines sans avoir à tout reconfigurer.

# Routage inter-domaine sans classe (CIDR)

# Routage inter-domaine sans classe (CIDR)

Au début des années 90, le modèle d'adressage par classes montre ses limites et il devient urgent de trouver un autre modèle avant une pénurie imminente d'adresses.

Le modèle **CIDR** : ***Classless Inter-Domain Routing*** est proposé et utilisé dès 1994.

**C'est encore ce modèle qui est utilisé aujourd'hui.**

Objectif : s'affranchir complètement de la notion de classes grâce à l'utilisation de masques de réseau moins contraints.

# Routage inter-domaine sans classe (CIDR)

## Les avantages de CIDR

- Permet d'économiser des adresses IP
- Facilite le routage : utilisé dans les tables de routage pour **agréger plusieurs routes possibles**.
- Définit un système d'adressage plus simple.

# Routage inter-domaine sans classe (CIDR)

## Principe

Puisque l'on s'affranchit des classes d'adresses, il devient possible de définir des identifiants de réseaux moins « contraints ». Ce modèle consiste à permettre de fusionner ensemble des (sous-)réseaux ayant des préfixes identiques.

On appelle cette fusion le ***supernetting***.

# Routage inter-domaine sans classe (CIDR)

## Exemple

Soit l'adresse 192.224.11.0/23 une adresse IP.

En binaire, cette adresse s'écrit :

**110**00000.11100000.00001011.0000

Ce qui devait correspondre à une adresse de classe C.

Pourtant le masque (23 bits à 1) s'écrit 255.255.254.0 et ne correspond pas au masque classique des adresses de classe C (255.255.255.0)

**Et pourtant, ce type d'adresse réseau est possible en CIDR alors qu'il n'existait pas dans l'ancien modèle.**

# Routage inter-domaine sans classe (CIDR)

La plage des adresses IP des hosts sur le réseau 192.224.10.0/23 est :

Adresse début : 192.224.10.0 (11000000.11100000.00001010.00000000)

Adresse fin : 192.224.11.254 (11000000.11100000.00001011.11111110)

Elle peut donc être vue comme la fusion de deux sous-réseaux de classe C :

Sous-réseau 1 : 192.224.10.0/24

Sous-réseau2 : 192.224.11.0/24

Cette technique s'appelle ***supernetting*** ou encore « ***route summarization*** » (résumé de route).

# Routage inter-domaine sans classe (CIDR)

**En pratique, comment faire pour fusionner des sous-réseaux?**

- 1) Identifier les réseaux ayant un préfixe identique
- 2) Ecrire l'adresse de chacun de ces réseaux au format binaire
- 3) Retrouver le plus grand préfixe commun, ce qui donne le nombre de bits du masque.
- 4) Le *network id* (adresse réseau) est la plus petite adresse IP parmi l'ensemble des adresses de sous-réseaux



# Utilisation de masques à longueur variable (VLSM)

# Masques de longueur variable (VLSM)

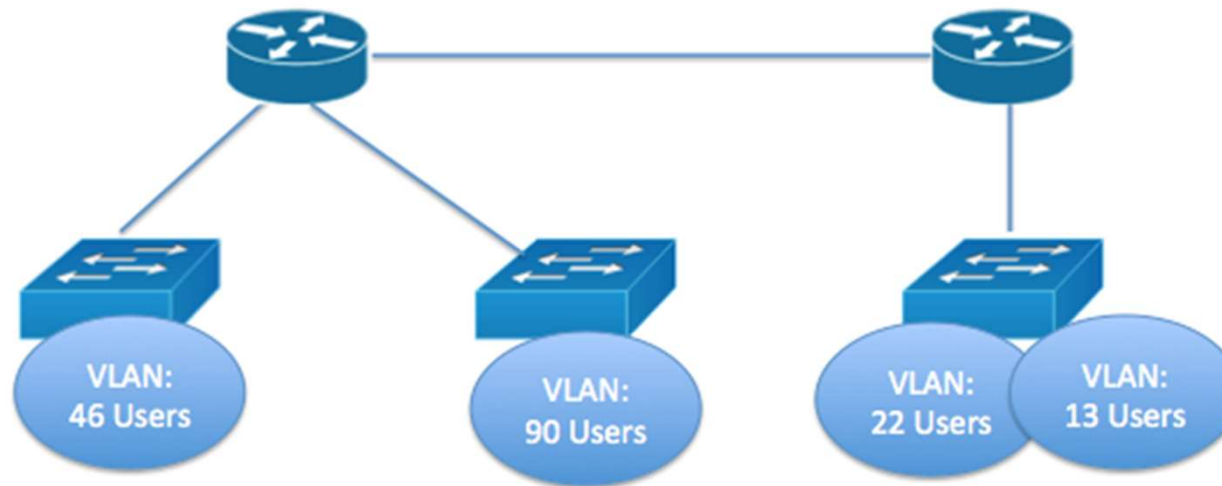
Une autre technique permet de mieux gérer les adresses IP et d'empêcher leur gaspillage : **VLSM** (Variable Length Subnet Mask).

Il s'agit d'une extension de CIDR permettant une meilleure gestion des adresses au **niveau local** (CIDR concerne plutôt internet).

Le VLSM permet à un « client » de n'acquérir qu'une partie d'un sous-réseau. Ainsi, les différents réseaux seront partagés en sous-réseaux dont les masques auront des longueurs différentes.

# Mise en œuvre VLSM

## Mise en place du plan d'adressage



Plage d'adresses : 192.168.0.0/24 (ancien classe C).  
254 adresses disponibles.

# Mise en œuvre VLSM

## Nombre d'adresses nécessaire pour ce réseau

171 utilisateurs total (46+90+22+13)

6 adresses pour les routeurs (1 sur chaque VLAN + 1 vers le routeur distant)

Soit 177 adresses réparties sur 5 sous-réseaux

# Mise en œuvre VLSM

On a besoin de 5 sous-réseaux.

Supposons que l'on fasse du subnetting standard : à partir de l'adresse initiale, nous pouvons de créer effectivement 5 sous-réseaux contenant chacun exactement 30 machines....

- 3 bits utilisés pour identifier les 5 sous-réseaux
- 5 bits restants => 30 hosts par sous-réseaux !!!

**Système peu flexible et pas adapté à nos besoins**

**VLSM va nous permettre de régler le problème en adaptant la taille des sous-réseaux**

# Mise en œuvre VLSM

## Découpage en sous-réseaux

**Etape 1** : Commencer par le réseau comportant le plus grand nombre d'adresses IP (soit  $N$  ce nombre).

Déterminer le nombre de bits utiles pour définir les  $N$  adresses;  
Soit  $b$  ce nombre. En déduire la taille du masque de sous-réseau  
( $32 - b$ )

Ici  $N = 91$ , on a donc besoin de 7 bits pour identifier les machines en tenant compte des 2 adresses réservées ( $2^6 \Rightarrow 62$  mais  $2^7 \Rightarrow 126$ ). Taille du masque de sous-réseau = 25 bits ( $32 - 7$ )

**Il nous faut donc un masque de 25 bits pour couvrir les besoins du plus gros VLAN**

# Mise en œuvre VLSM

**Remarque :** en travaillant avec des masques de taille fixe (25 bits), on ne pourrait alors définir que 2 sous-réseaux possibles :

**192.168.0.0/25** et **192.168.0.128/25**

-> seul le premier bit du dernier octet permet de distinguer les 2 sous-réseaux (128 = 1000 0000b).

On va donc redécouper le second sous-réseau en réseaux plus petits avec un masque plus grand pour les autres VLAN.

# Mise en œuvre VLSM

**Etape 2 :** on recommence avec le deuxième sous-réseau (en terme de nombre d'adresses).

On a besoin de 47 adresses (46 machines + routeur) soit 6 bits.

Le masque du second sous-réseau aura donc une taille de 26 bits (32-6).

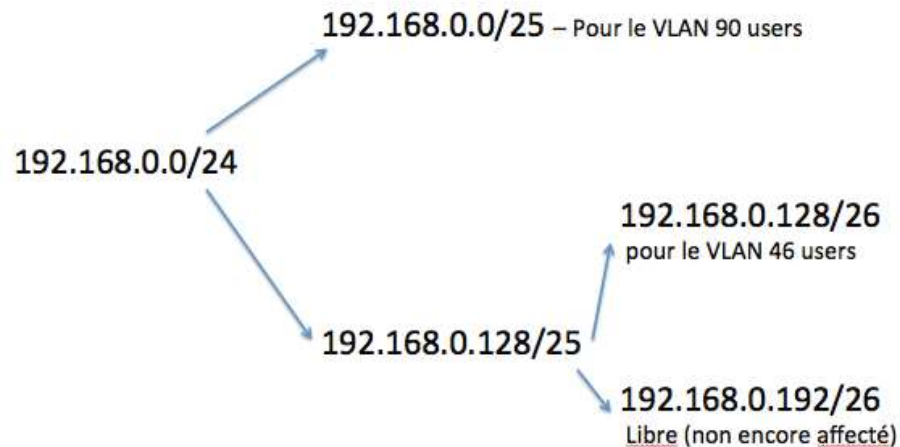
Le deuxième sous-réseau est créé à partir de l'adresse 192.168.0.128/25

**192.168.0.128/26**



# Mise en œuvre VLSM

A ce stade on a donc :




On continue ainsi de suite pour tous les sous-réseaux...

# Réseaux privé et traduction d'adresses (NAT)

# Réseaux privés et traduction d'adresses (NAT)

En réponse :

- À la pénurie imminente des adresses IP (malgré CIDR)
  -  on considère que la quasi-totalité des plages d'adresses IP a été utilisée en juin 2012.
- Aux besoins de sécurisation des réseaux (d'entreprises)

Mise en place de la notion de réseaux dits **privés** ayant peu ou pas de connexions vers l'extérieur (internet).

## Réseaux privés et traduction d'adresses (NAT)

Si le réseau (privé) n'est **jamais** connecté à d'autres réseaux (dont internet) on peut utiliser n'importe quelle adresse à l'intérieur de celui-ci.

Si le réseau privé peut être connecté à d'autre réseaux (via un routeur), on utilisera deux « espaces » d'adresses : les **adresses privées** et les **adresses publiques**.

# Réseaux privés et traduction d'adresses (NAT)

## Les adresses privées sont :

- propres aux réseaux internes et ne sont « connues » que dans celui-ci.
- uniques dans un même sous-réseau, mais pas uniques dans le monde (plusieurs sous-réseaux peuvent utiliser les mêmes adresses).

C'est pour cela que Gautier et Antoine ont des adresses IP égales !

- **non routables.**
- définies dans un document RFC1918
  - plage de 10.0.0.0 à 10.255.255.255 pour de grands réseaux
  - plage de 172.16.0.0 à 172.31.255.255 pour des réseaux de taille moyenne
  - plage de 192.168.0.0 à 192.168.255.255 pour des réseaux de petite taille (ex réseaux domestiques)

# Réseaux privés et traduction d'adresses (NAT)

## Principe

Toutes les machines d'un réseau interne, connectées à internet par l'intermédiaire d'un routeur et ne possédant pas d'adresse IP publique doivent utiliser une adresse contenue dans l'une de ces plages.

# Réseaux privés et traduction d'adresses (NAT)

## Les adresses publiques sont :

- « connues » dans le monde entier.
- uniques dans le monde entier
- **routables**
- Gérées par l'**IANA** : Internet Assigned Number Authority

# Réseaux privés et traduction d'adresses (NAT)

## Les exceptions :

**Réseau 127.0.0.0** : réservé pour des tests en boucle locale.

Exemple : 127.0.0.1 -> localhost

**Réseau 0.0.0.0** : réservé pour définir les routes par défaut sur les routeurs.



# Réseaux privés et traduction d'adresses (NAT)

Mise en place d'un système de traduction d'adresses, **NAT** (Network Address Translation), de manière à :

- Partager une interface unique du réseau internet entre tous les hôtes du réseau privé,
- Rendre un serveur du réseau privé accessible depuis internet (incluant un niveau de sécurité),  
Cela sert notamment à récupérer la réponse aux requêtes du réseau privé.

# Réseaux privés et traduction d'adresses (NAT)

**Partager une interface unique du réseau internet  
entre tous les hôtes du réseau privé**

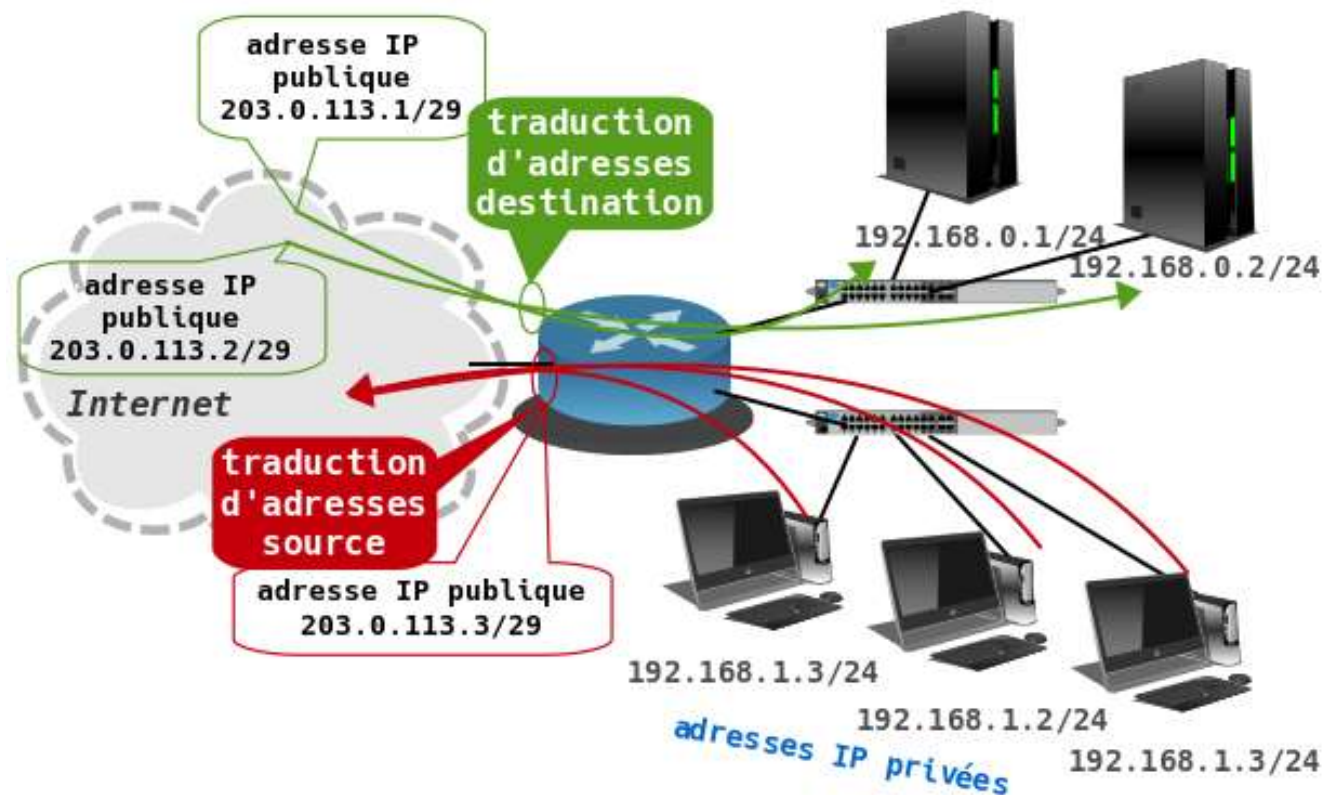
Traduction d'adresses sources (**S-NAT**). Les adresses sources des paquets IPv4 émis par les hôtes du réseau privé sont réécrites avec une adresse IPv4 publique.

# Réseaux privés et traduction d'adresses (NAT)

## Rendre un serveur du réseau privé accessible depuis internet

Traduction d'adresses destination (**D-NAT**). Une adresse IPv4 publique est réécrite avec une adresse IPv4 privée en fonction du service internet demandé.

# Réseaux privés et traduction d'adresses (NAT)



# Réseaux privés et traduction d'adresses (NAT)

## Deux types de traduction

NAT statique

NAT dynamique

# Réseaux privés et traduction d'adresses (NAT)

## NAT Statique

On associe une adresse IP publique (routable) à une adresse privée (interne au réseau). La **passerelle** traduit dans les deux sens en passant de l'une à l'autre.

-> permet de connecter de manière transparente les machines du réseau interne à internet.

-> ne résout pas la pénurie d'adresses (une adresse privée est associée à une adresse publique)

-> utilisé pour donner accès à des serveurs (internes) depuis l'extérieur

**Ce type de traduction est très peu utilisée**

# Réseaux privés et traduction d'adresses (NAT)

## NAT Dynamique

Partage d'une adresse IP routable entre plusieurs machines ayant une adresse privée .

Vue de l'extérieur, toutes les machines du réseau interne possèdent la même adresse IP. On parle de **mascarade IP** (*IP masquerading*)

-> en général c'est l'adresse du routeur passerelle (la box) qui est utilisée.

L'utilisation d'un mécanisme de traduction de port (PAT : Port Address Translation) permet d'identifier les machines du réseau.

# Réseaux privés et traduction d'adresses (NAT)

## Critiques du NAT :

Casse la structure pair-à-pair d'internet :

« je peux me connecter à internet mais l'inverse n'est pas vrai »

-> problème pour certaines applications (ex. VoIP)

(Fausse) Impression de sécurité :

« ma station a une adresse privée non adressable derrière un NAT, donc on ne peut pas m'attaquer »

-> certaines attaques restent possibles dans cette configuration.



# Réseaux privés et traduction d'adresses (NAT)

## Attention !

Il s'agit bien d'un système de **traduction** d'adresses et non de translation d'adresses !

# Pour conclure sur IPv4

IPv4 a permis l'explosion d'internet.

Victime de son succès, le modèle d'adressage initial a du être abandonné.

Les techniques telles que CIDR, VLSM ainsi que NAT ont sauvé (temporairement) internet et ont permis d'optimiser l'utilisation de l'espace d'adressage.

Des problèmes de sécurité restent non résolus avec IPv4.

## Pour vraiment conclure

### **A-t-on répondu aux questions du chap 3 ?**

Qu'est-ce qu'une IP publique ? (IP Wan)

Qu'est-ce qu'une IP locale ou privée ? (IP Lan)

192.168.0.x

Qu'est-ce qu'une IP Statique/ IP Dynamique ?

# L'homme du jour



## **Robert Elliot Kahn, dit Bob Kahn**

né le 23 décembre 1938 à New-York. Ingénieur, chercheur et inventeur américain. Il est co-inventeur avec Vint Cerf du protocole TCP/IP et considéré comme l'un des pères fondateurs d'internet (Arpanet).

Il reçut le Prix Turing en 2004.

# Le futur d'internet : IPv6

Version 6 du protocole IP

Première version en 1995 : RFC 1883

Finalisation en 1998 : RFC 2460

# IPv6 pourquoi ?

Attribuer plus d'adresses

Réduire la taille des tables de routage

Router plus rapidement en simplifiant le protocole

Fournir une meilleure sécurité

Permettre la mobilité

# IPv6 : points clefs

Adresses et allocation des préfixes

Découverte des voisins

Format de datagrammes simplifié

# Adressage et espace de noms

Beaucoup plus d'adresses disponibles :  $2^{128}$   
soit

667 millions de milliards d'adresses disponibles  
par  $\text{mm}^2$  de la surface de la Terre !!!



# Adressage et espace de noms

## Notation

Plus de notation pointée, uniquement de l'hexadécimal  
: 8 groupes de 16 bits

Exemple :

2001:0DB8:0000:85A3:0000:0000:AC1F:8001

Possibilité de supprimer les 0 non significatifs (groupes de 1 à 3 ou blocs entiers)

2001:DB8:0:85A3::AC1F:8001

# Adressage et espace de noms

## Notation CIDR conservée

Adresse / taille : préfixe de l'ensemble d'adresses considérées

Exemple : 2001:DB8:1F89::/48

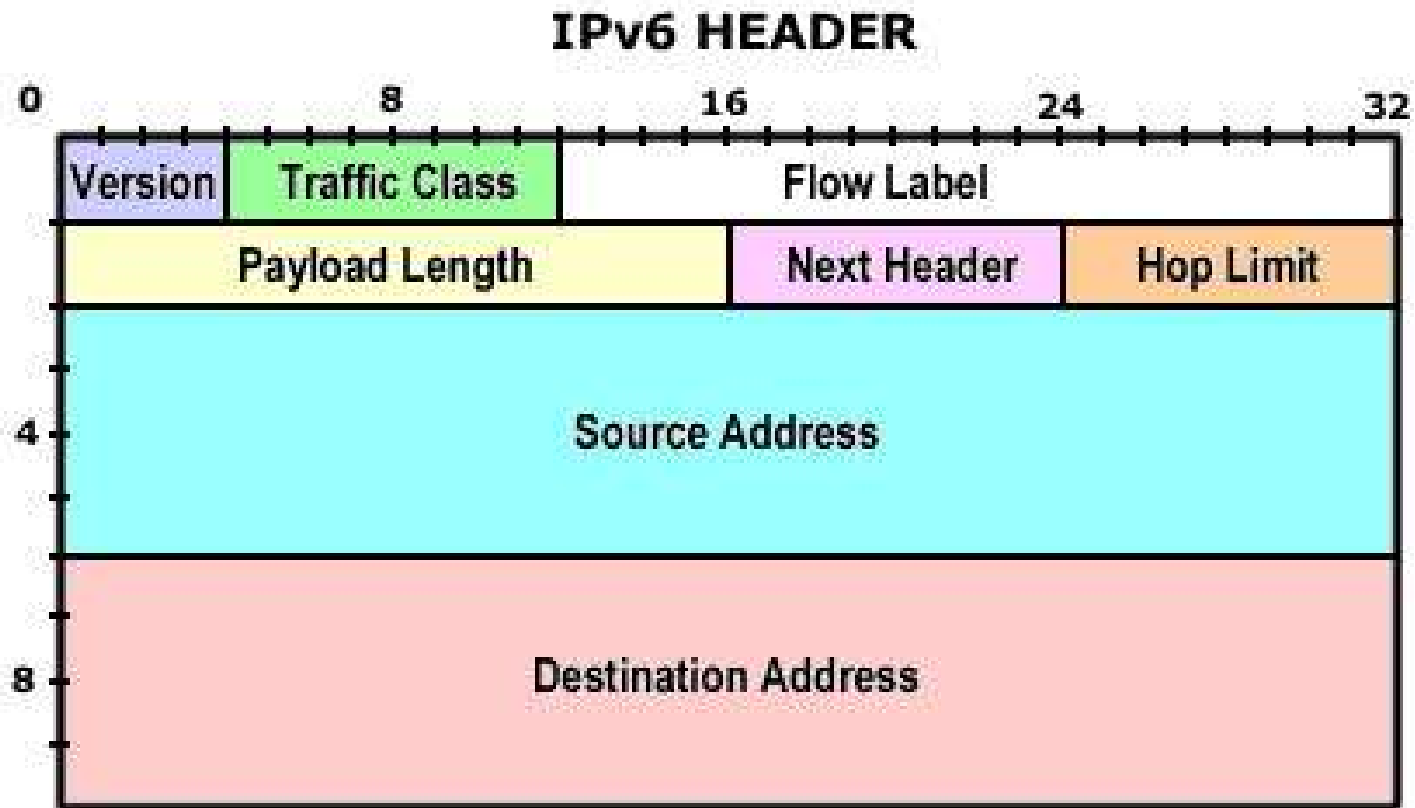
## Adresses réservées

Exemple : FF00::/8 -> multicast

## Cohabitation avec URL

http://[2002:400:2A41:378::34A2:36]:8080

# Entête IPv6



# Entête IPv6 : les champs

**Version** (4 bits) : numéro du protocole (6)

**Traffic Class** (8 bits) : indique si la donnée peut être ralentie en cas de congestion

**Flow Label** (20 bits) : permet d'identifier un flux pour un traitement spécifique dans le réseau

**Payload length** (16 bits) : taille des données utiles

**Next Header** (8 bits) : identifie le protocole de niveau supérieur

**Hop limit** (8 bits) : remplace TTL

## Et la fragmentation ?

Pas de fragmentation en IPv6 :

trop coûteuse pour les routeurs et  
génératrice de problèmes

Si problème de taille, le routeur renvoie un  
paquet *ICMPv6 Packet too big*.

C'est l'émetteur qui doit fragmenter.

# NDP : Neighbor Discovery Protocol

Protocole IPv6 permettant de découvrir :

- Les adresses MAC des hôtes voisins
- Les routeurs voisins pour une route donnée
- Des données utiles comme le MTU

Agrège les propriétés de ARP et ICMP de IPv4

# ICMPv6

## Internet Control Message Protocol v6

Définit des paquets contenant des messages spécifiques.

### Exemple :

1 : Destination Unreachable

2 : Packet too big

...

136 : Neighbor solicitation

# Transition entre IPv4 et IPv6

Les adresses ne sont pas compatibles

**Protocole de transition 6to4** : permet à un hôte IPv6 de communiquer avec un hôte IPv4 via le nuage IPv4

Correspondance entre les adresses

IPv4 : 192.0.2.4 -> IPv6 : 2002:C000:0204::/48



# Conclusion

IPv6 est réellement le futur d'internet et apporte de vraies améliorations

Son déploiement reste difficile.

Toute transition technologique est délicate, la transition complète d'IPv4 à IPv6 sera **très délicate**.