

# IPV6

Auteur Pascal Petit

De nombreux éléments de ce cours sont tirés du cours IPV6 de France université numérique

<https://www.fun-mooc.fr/courses/MinesTelecom/04012S02/session02/info>

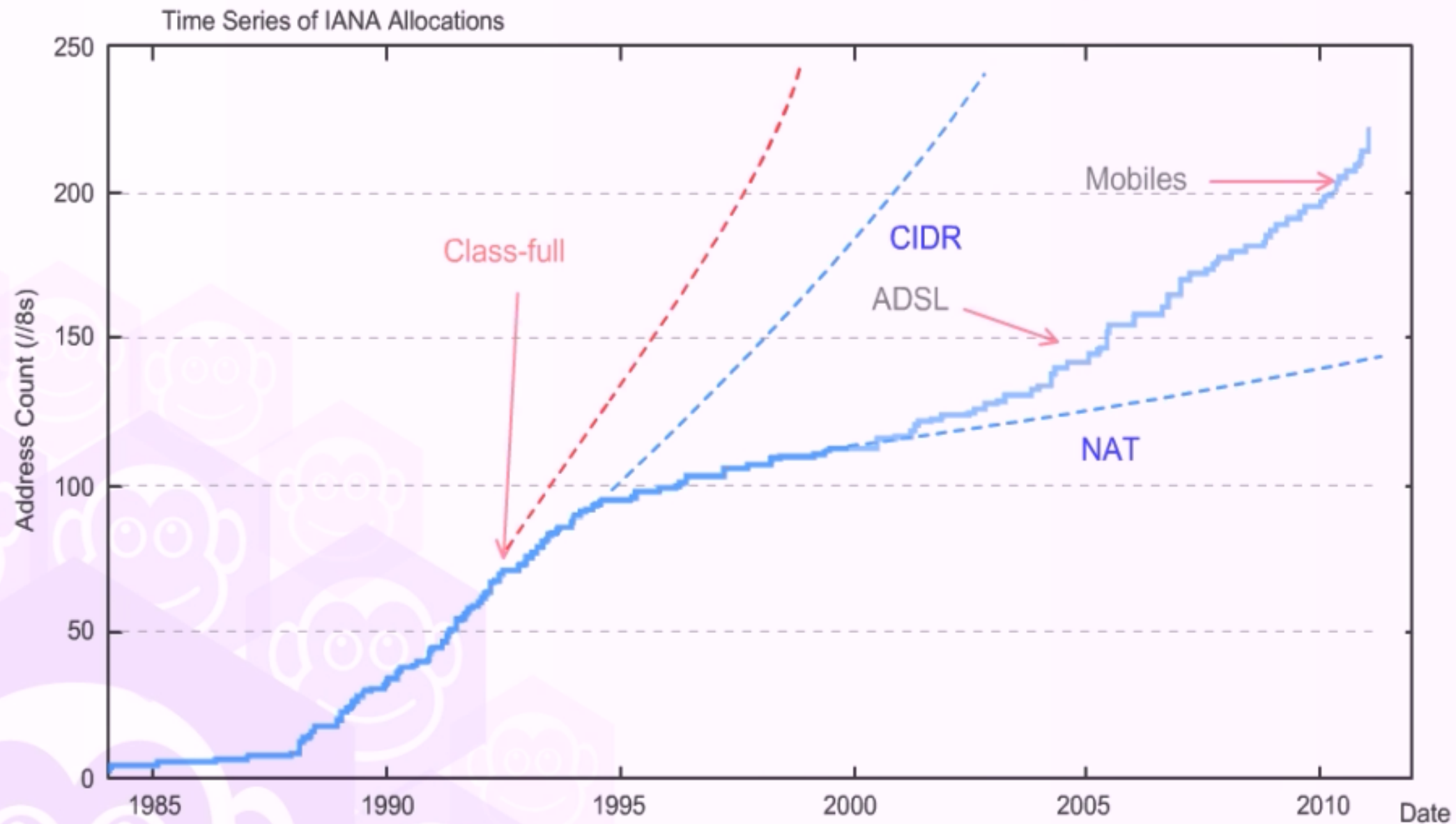
# Licence

- Ce document est publié sous la Creative Commons CC BY-SA 4.0 International
  - Vous pouvez le modifier à votre guise
  - Vous pouvez distribuer ce document et les versions modifiées, y compris pour utilisation commerciale ;
  - Vous devez attribuer le document initial à leurs auteurs et préciser les éventuelles modifications que vous aurez faites ;
  - Les versions que vous distribuez, modifiées ou non par vous doivent être distribuées sous les mêmes conditions.

# Protocole IPV6

- Successeur d'ip v4
- Finalisé dans la RFC 2460 (décembre 1998)
- Espace d'adressage important (adresses de 128 bits)
  - Résout l'épuisement des adresses ipv4
  - Élimine la nécessité d'utiliser du NAT (traduction d'adresses)
  - Permet plus de flexibilité dans l'attribution des adresses
  - Permet une meilleure agrégation des routes dans les tables de routage d'internet
- Mécanismes d'attributions automatiques d'adresses IP
- Intégration de la sécurité (IPSec)
- Simplification du format de l'entête

# Épuisement IPv4



Source : <http://livre.g6.asso.fr/images/7/7e/41-fig1-v1.png>

# Adressage IPv6

- 2 fonctions comme en IPV4
  - Identification d'une machine parmi l'ensemble des machines du réseau
  - Localisation :
    - Globale identifiant le réseau
    - Locale identifiant une machine sur un réseau donné
- Comme en IPV4 :
  - Routage
  - Partie réseau et partie hôte d'un adresse IP
  - 2 hôtes ayant la même partie réseau sont sur le même réseau
  - Notation CIDR /nn pour indiquer la taille de la partie réseau

# Adressage IPv6

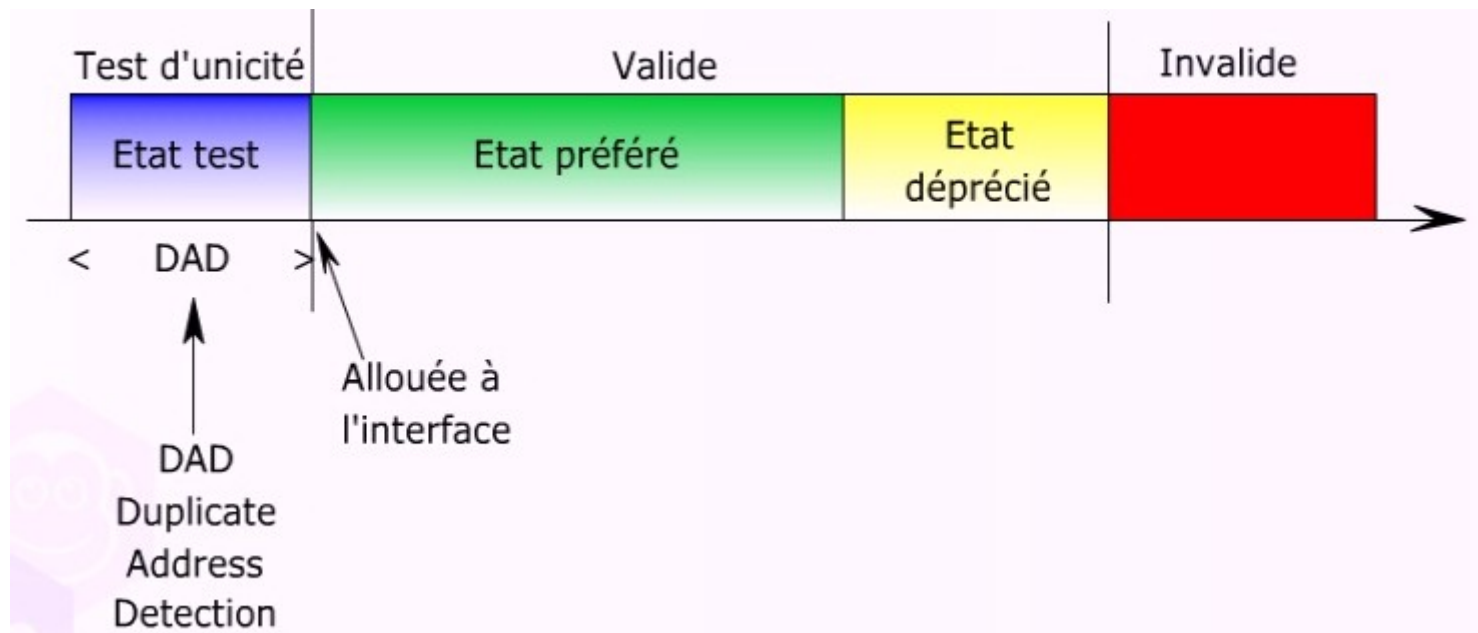
- Adressage hiérarchique
  - Opérateurs du coeur d'internet : décisions sur des préfixes courts
  - En périphérie, routage sur des préfixes longs
- l'adresse IPv6 fait 128 bits (contre 32 bits pour l'adresse IPv4)
- Les sites multidomiciliés ont autant d'adresses que de fournisseurs
- Des mécanismes de renumérotation automatique permettent de changer facilement de préfixes

# Durée de vie d'une adresse IPv6

- l'attribution d'une adresse à une interface est temporaire
- Durée de vie : temps pendant lequel l'interface est dépositaire de l'adresse
- Par défaut 30 jours
- Cas particulier : une adresse lien local a une durée de vie illimitée (fe80...)
- Renumérotation d'interface : passage d'une adresse à une autre
  - Transition via un mécanisme obsolescence ;
  - Plusieurs adresses valides à une même interface

# Vie d'une adresse IPv6 :

- État d'un IPv6 :
  - DAD : duplicate address detection
  - Préféré : pas de restriction
  - Déprécié : non utilisé pour nouvelles connexions





# Notation des adresses IPV6

- 16 octets
- adresse notée en hexadécimal (16 chiffres de 0 à 9 puis de A à F)
- En majuscules ou en minuscules ou en mixant les 2
- Exemple : 2001:0db8:0000:0000:0008:0800:200C:417A
- Par convention ; on peut supprimer les 0 de poids forts non significatifs : 2001:db8:0:0:8:800:200C:417A
- On peut abréger plusieurs champs nuls consécutifs par :: (mais une seule fois) : 2001:db8::8:800:200C:417A

# Exemples :

exemple	l'adresse	Peut s'écrire
Adr. unicast	2001:db8:0:0:8:800:200C:417A	2001:db8::800:200C:417A
Adr. multicast	ff01:0:0:0:0:0:0:101	ff01::101
bouclage	0:0:0:0:0:0:0:1	::1
Non spécifiée	0:0:0:0:0:0:0:0	::
Lien local	fe80:0:0:0:64:73ff:fe77:6160/64	fe80::64:73ff:fe77:6160/64

# IPv6 notation canonique

- RFC 5952
- Présentation destinée à l'affichage, aux sorties des programmes, log, ...
- En entrée, tout logiciel doit accepter toutes les formes valides
- Forme canonique
  - Supprimer les zéros initiaux
  - :: doit être utilisé sur la série la plus longue, la plus à gauche en cas d'égalité
  - Chiffres hexa en minuscules
  - Si un numéro de port doit être indiqué, l'adresse doit être entre crochets

# Examples

# Types d'adresses IPV6

- Adresses unicast
- Adresses multicast
- Adresses Anycast
- Types d'adresses généralement définis par leur préfixe dans le RFC3513

# Adresses unicast

- Adresses globales : unique sur tout l'Internet  
2000 ::/3 (adresses commençant par 2 ou 3)
- Adresses localement restreinte (obsolète) :  
équivalent des adresses privées d'IPv4
- Lien local : restreintes à un lien ou à un  
domaine de diffusion type VLAN. Ne sera pas  
routée : fe80 :....

# Adresses multicast/anycats

- Adresses multicast :
  - désigne un groupe d'interfaces appartenant à différents nœuds pouvant être situés n'importe où sur le réseau.
  - le paquet est remis à TOUTES les interfaces du groupe
  - Remplace en plus souple et en plus sélectif la diffusion (broadcast) d'IPv4 : ex. s'adresser à tous les serveurs DHCP, à tous les routeurs, ...
- Adresses anycast
  - Désigne un groupe d'interfaces
  - Le paquet est remis à UN des membres du groupe
  - Utilisation expérimental pour l'instant (2016)

# Adresses unicast globales : RFC3587

- RFC3587
  - définit la structure d'adressage IPv6 en précisant les tailles de chacun des blocs
  - géré hiérarchiquement
    - Topologie publique fournie par le FAI (48 bits)
    - Topologie de site (16 bits)
    - Identifiant d'interface des hôtes sur 64 bits
  - Un opérateur peut aussi fournir un /56 à ses clients





# Adresses unicast globales : RFC3587

- Préfixes réservés :
  - préfixe 2002::/16 qui est est réservé au mécanisme de transition 6to4;
  - préfixe 2001:db8::/32 est réservé pour la documentation (non routé)
  - préfixe 3ffe::/16 était le préfixe des adresses du réseau expérimental 6bone (arrêté le 6/6/2006, déprécié)
- Le plan 2000/3 a été découpé en plages affectées par l'IANA aux différents RIR

# L'adressage IPv6

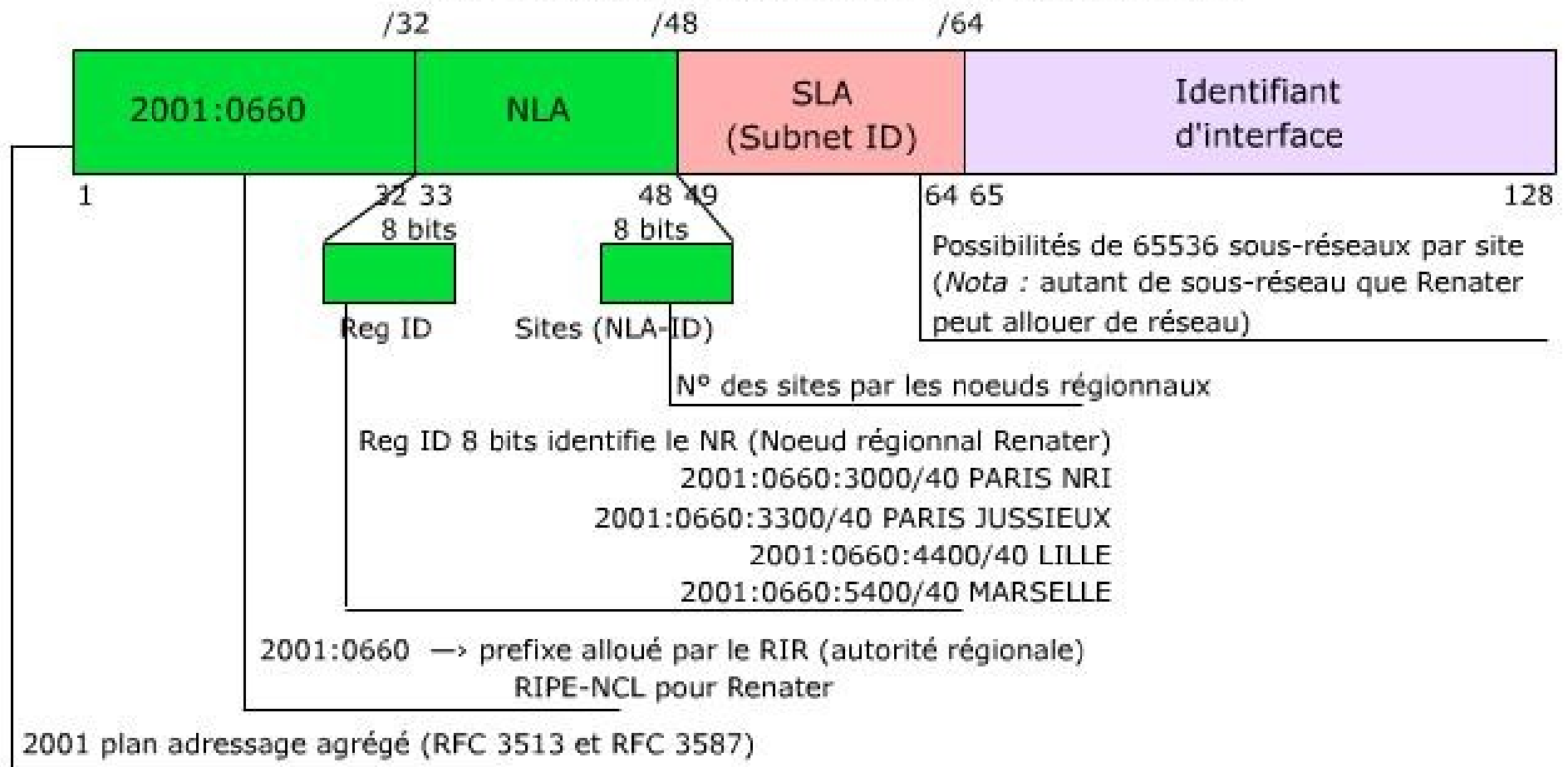
Allocation	Préfixe binaire	Préfixe Hexa
Réservé	0000 0000	0000 :: / 8
Non alloué	0000 0001	0100 :: / 8
Réservé pour allocation	0000 001	0200 :: / 8
Non alloué	0000 01	0400 :: / 6
Non alloué	0000 1	0800 :: / 5
Non alloué	0001	1000 :: / 4
<b>Global Unicast</b>	<b>001</b>	<b>2000 :: / 3</b>
Non alloué	010	4000 :: / 3
Non alloué	011	6000 :: / 3
Non alloué		100 8000 :: / 3
Non alloué		101 A000 :: / 3
Non alloué		110 C000 :: / 3
Non alloué		1110 E000 :: / 3
Non alloué	1111 0	F000 :: / 3
Non alloué	1111 10	F800 :: / 3
Non alloué		111110 FC00 :: / 7
Non alloué	1111 1110 0	FE00 :: / 9
<b>Link-Local Unicast Adresses</b>	<b>1111 1110 10</b>	<b>FE80 :: / 10</b>
<b>Site-Local Unicast Adresses</b>	<b>1111 1110 11</b>	<b>FEC0 :: / 10</b>
<b>MultiCast Adresses</b>	<b>11111111</b>	<b>FF00 :: / 8</b>

Allocation  
maintenant  
obsolète

# Adresses IPV6

## L'adressage Renater 3 est conçu de manière hiérarchisée

Chaque site se voit attribuer un NLA-ID soit 48 bits



# Mécanismes de gestion : découverte de voisins

- Remplace ARP
- Obtention de l'adresse Mac ou détection d'adresses IP dupliquées
- Via des messages ICMPv6
  - Sollicitation d'un voisin (Neighbor Sollicitation ou NS)
  - Annonce d'un voisin (NeighborAdvertisement ou NA).
- Adresse destination : adresse multicast en général (pas de broadcast)

# Mécanismes de gestion : autoconfiguration

- Étapes
  - Mise en place d'une adresse lien local (vérification d'unicité)
    - La machine peut communiquer avec les hôtes de son réseau
  - Obtention des informations de son routeur
    - Autoconfiguration sans état (choix de l'IP par l'hôte)
    - Autoconfiguration avec état (DHCPv6)
  - Routeur par défaut : adresse source du paquet d'annonce de routeur
  - Dns (cas sans état) : RFC3646
    - Via DHCP
    - Via les annonces de routeur (RA)
    - Via anycast (abandonnée)

# Transition vers ipv6

- IPV6 a de nombreux points communs avec IPV4
- MAIS IPV6 est incompatible avec IPV4
  - Un client IPV4 ne peut communiquer directement avec un serveur IPV6
  - Un client IPV6 ne peut communiquer directement avec un serveur IPV4
- Une machine peut communiquer simultanément en IPV4 et en IPV6 (dual stack)
- Comment faire communiquer des îlots IPV6 au travers d'un réseau IPV4 ?
- Pas de grand soir mais une intégration progressive d'IPV6 dans le réseau actuel (=> nécessite de mécanismes de compatibilité)

# IPV4/IPV6 : état des forces

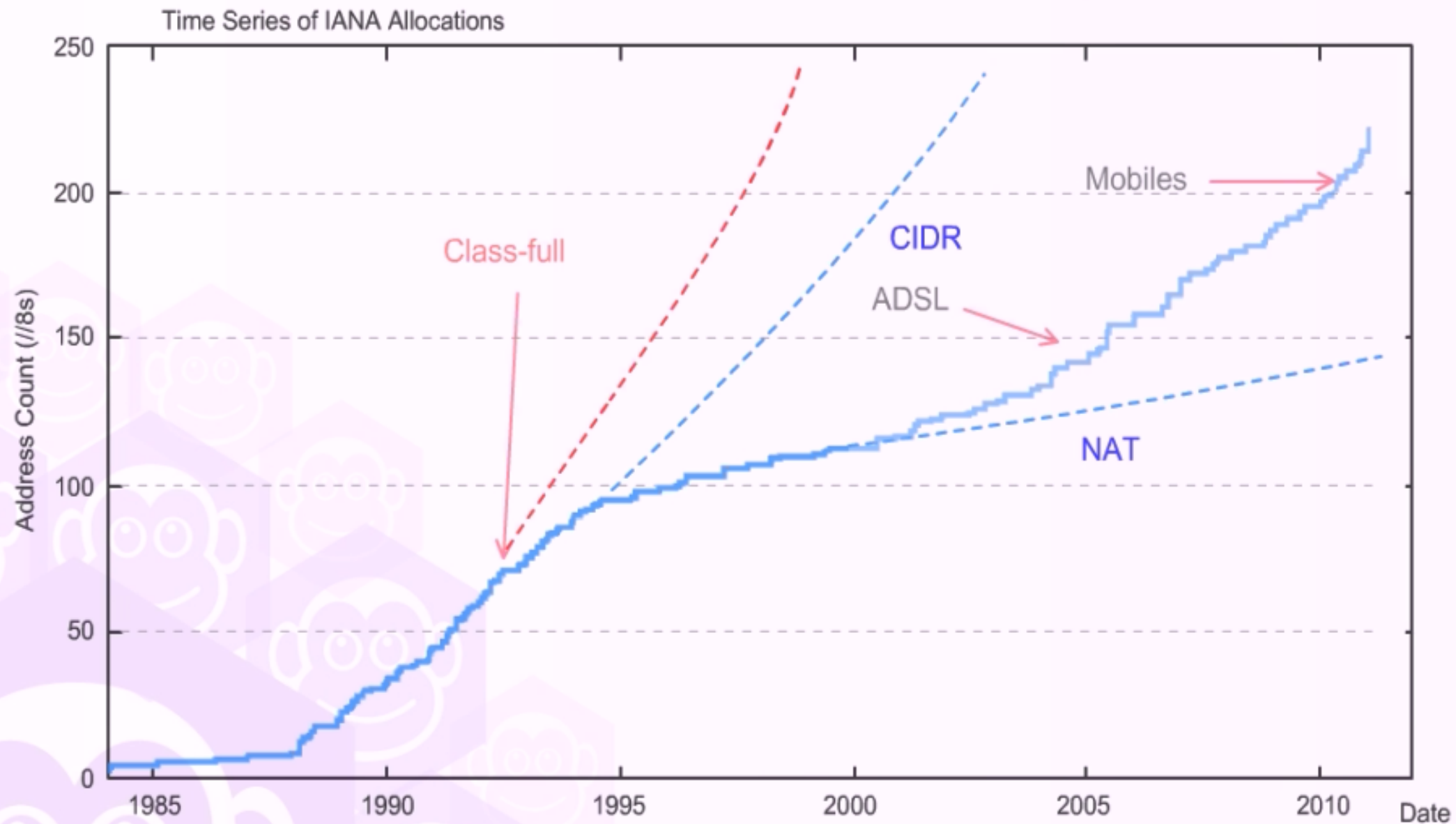
- IPV4 : épuisement des adresses
  - 3 février 2011 : plus aucun bloc libre (tous affectés à registre régional)
- Remèdes à l'épuisement
  - CIDR (1993, rfc1518, rfc1519 → rfc4632) ;
  - Mise en œuvre de Server Name Indication (SNI, rfc4366) permettant plusieurs noms de serveurs https sur une même IP ;
  - Adresses privées (rfc1918) et traduction d'adresse (NAT)
  - CGNAT (Carrier Grade NAT ou NAT444, rfc6888)

# NAT, CGNAT : une mauvaise solution

- Ici, on décrit les problèmes posés par ces solutions



# Épuisement IPv4



Source : <http://livre.g6.asso.fr/images/7/7e/41-fig1-v1.png>

# IPV6

- l'adoption d'ipv6 progresse : le 30/01/2018, 18,14 % des requêtes google venaient de machine en ipv6 (cf <https://www.google.fr/ipv6/statistics.html>)
- Akamai a des statistiques évalentes (avec un taux de 47 % (2016) pour la Belgique, cf <https://www.akamai.com/fr/fr/our-thinking/state-of-the-internet-report/state-of-the-internet-ipv6-adoption-visualization.jsp>)
- De plus en plus de fournisseurs d'accès supportent ipv6
- Certains réseaux mobile sont en ipv6 seul cf <http://blog.g6.asso.fr/2013/11/07/t-mobile-usa-un-reseau-3g-ipv6-seul/>
- Les CDN (Content Delivery Networks) supportent souvent ipv6 (Akamai, Cloudflare, ...)
  - Mise à dispo en ipv6 (et ipv4) du contenu d'un serveur seulement ipv4

# IPv6 sur un réseau local via la double pile

- Les machines, ... du réseau peuvent avoir en même temps
  - Une adresse IPv4
  - Une adresse IPv6
- Avantages :
  - Compatibilité avec tous les types de serveurs (ipv6 ou ipv4)
- Défauts :
  - Temps de connexion sur un service non IPv6 sur serveur IPv6 et IPv4 (cf diapo suivantes)
  - Complexifie le debug réseau (quand un processus démarre en IPv6, passe par IPv4 puis IPv6, ...)
  - Si on le veut sur les serveurs, garantir le service en IPv6 et en IPv4 (utile si abandon progressif d'IPv4)
  - Performance IPv6 si utilisation d'un tunnel

# Double pile : performances

- Établissement de connexion : soucis si utilisation séquentielle IPV6 (échec) puis IPv4 (OK)
  - Cas d'un serveur avec IPv6 et IPv4 qui ne rend un service qu'en IPv4
- Solution (rfc6555) : établir les connexions en parallèle et garder la plus rapide
  - Soucis : pour les CGN IPv4, les connexions supposent la gestion d'un état (coûteux)
  - Solution : utiliser en priorité les protocoles qui ne génère pas d'état (IPv6)
- Solution : n'annoncer dans le dns que l'IP fournissant le service

# Compatibilité des applications

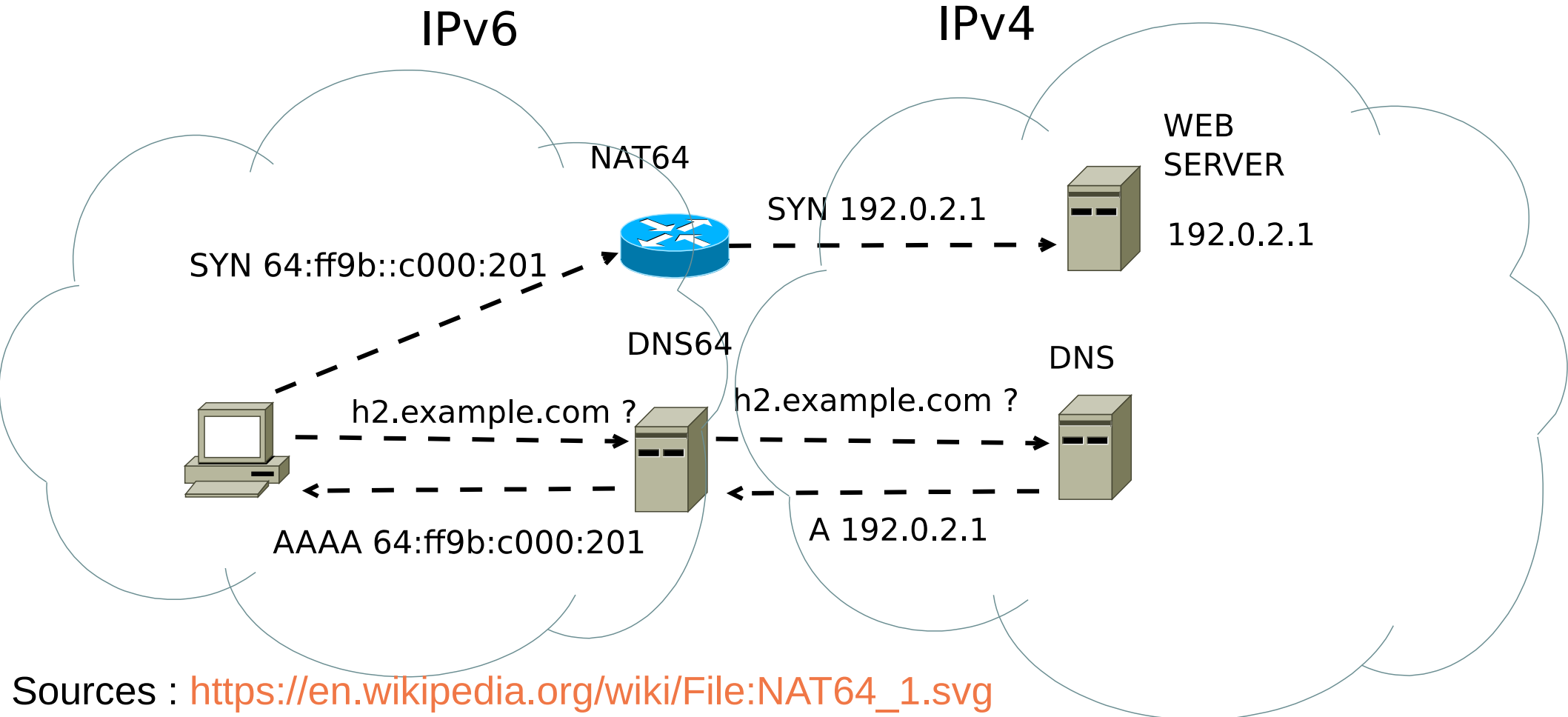
- Adresse sur 128 bits
- IPv4 mapped IPv6 adresse : stocker une IPv4 dans une IPv6 : `::FFFF:192.0.2.1`
- Rendre une application IPv6 compatible
  - Langage de haut niveau : utiliser des types d'une bibliothèque intégrant IPv6 : standard et automatique de nos jours
  - Sinon, pour le bas niveau, tenir compte de la taille de l'adresse et du remplacement de la diffusion par le multicast

# Interopérer les applications par traduction

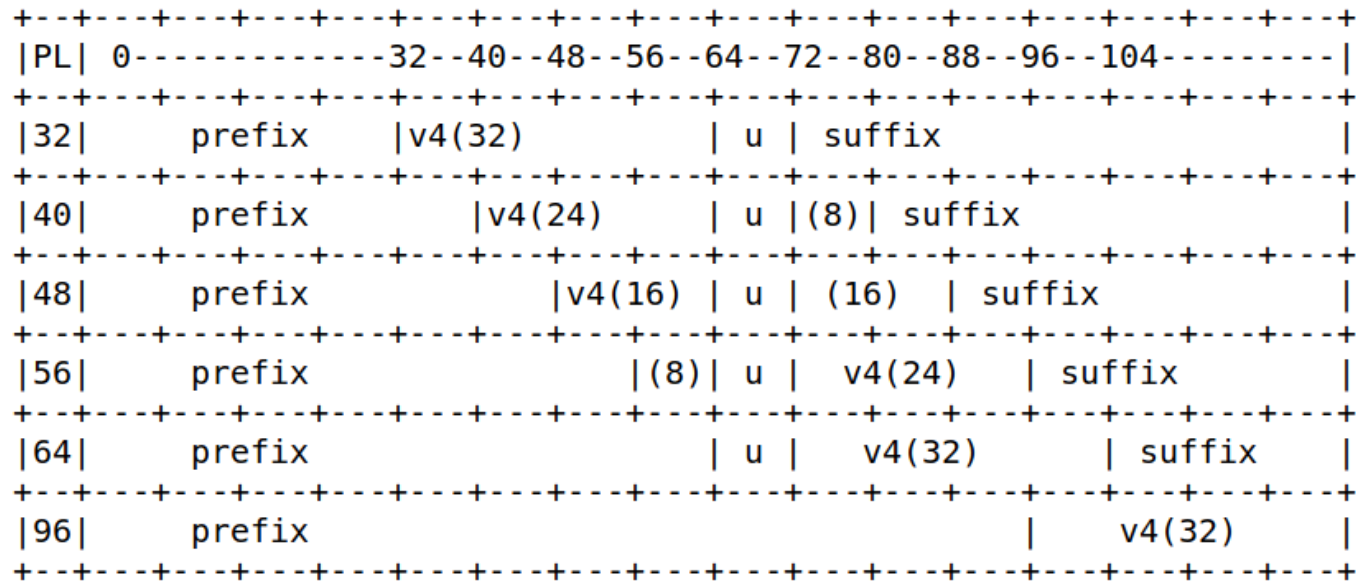
- Cette partie du document est très largement inspirée du travail réalisé par des membres du G6 dans le cadre du MOOC IPv6 de la plateforme FUN.
- Cf [http://livre.g6.asso.fr/index.php/MOOC:Compagnon\\_Act44](http://livre.g6.asso.fr/index.php/MOOC:Compagnon_Act44)

# NAT 64

- Cas typique : Permettre à un client IPv6 d'accéder à un serveur IPv4



# RFC 6052 : IPv4-Embedded IPv6 Address Format



- Suffix et les bit 64 à 71 sont à zéro (usage futur)
- « Well known prefix » : 64:ff9b ::/96 (neutre pour le calcul de checksum de couche 4)



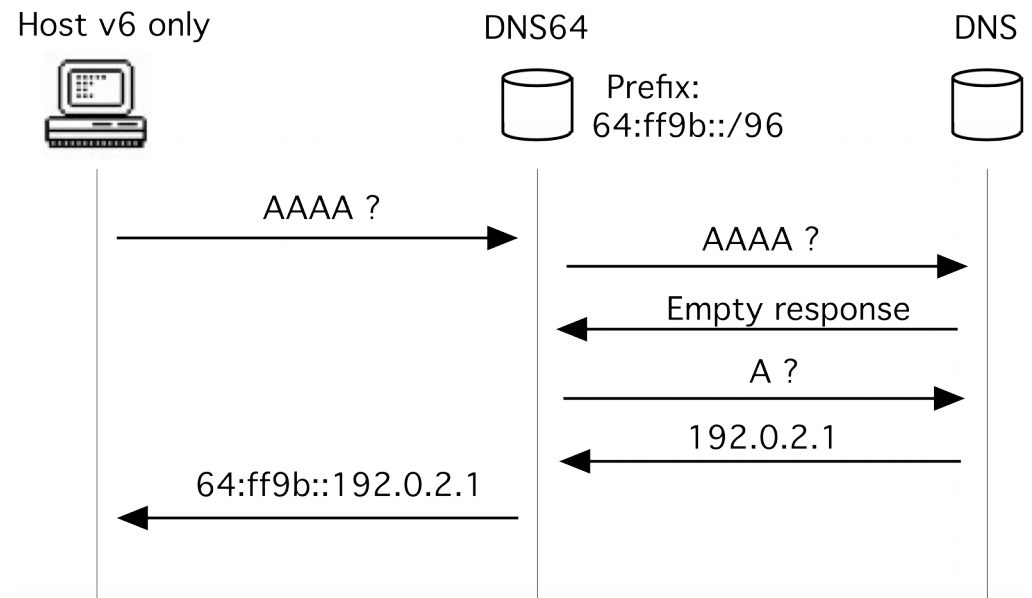
# RFC 6052 : IPv4-Embedded IPv6 Address Format

- Exemples (tirés de la RFC 6052) :

Network-Specific Prefix	IPv4 Address	IPv4-embedded IPv6 address
2001:db8::/32	192.0.2.33	2001:db8:c000:221::
2001:db8:100::/40	192.0.2.33	2001:db8:1c0:2:21::
2001:db8:122::/48	192.0.2.33	2001:db8:122:c000:2:2100::
2001:db8:122:300::/56	192.0.2.33	2001:db8:122:3c0:0:221::
2001:db8:122:344::/64	192.0.2.33	2001:db8:122:344:c0:2:2100::
2001:db8:122:344::/96	192.0.2.33	2001:db8:122:344::192.0.2.33

# DNS 64 : dns menteur (rfc 6147)

- Les services sont localisés par le dns :
  - Si ressource avec RR AAAA → IPv6
  - Si ressource sans RR AAAA, on essaie RR A et IPv4 et on renvoie un RR AAAA adapté du RR A
- Pb avec dnssec



# NAT 64 : sans état (RFC 7915)

- Problématiques :
  - comment associer une IPv6 à une connexion retour IPv4
  - Comment convertir un paquet IPv6 en paquet IPv4 et réciproquement
- Solution : toute IPv6 initiant une connexion correspond à une IPv4
- Défaut : il faut des IPv4
- Avantage : modèle sans état donc pas de table, performances indépendantes du nombre de connexions
- Usage typique : devant un serveur purement IPv4 devant être utilisé par des clients IPv6

# SIIT : Stateless IP/ICMP Translation (RFC 7915)

- Transposition protocolaire des champs de l'entête
  - De nombreux champs des paquets IPv4 et IPv6 ont des significations proches (TTL/Hop limit, DiffServ, payload Length)

# RFC 7915 : entête IPv6 à partir d'IPv4

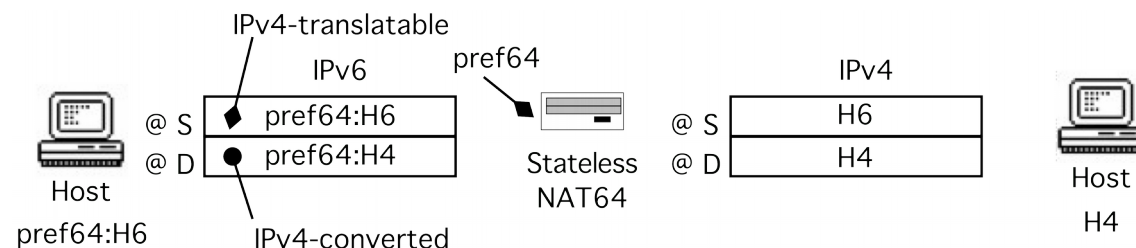
Champ de l'en-tête IPv4	Champ dans le nouvel en-tête IPv6	Valeur
<b>Version</b>	Version	6
<b>IHL</b>		Ignorer
<b>Type Of Service</b>	Traffic Class	Recopier
	Flow label	0
<b>Packet Length</b>	Payload Length	Packet Length - IHL (en-tête IPv4 + options) + 8 (si extension de fragmentation)
<b>Ident./Flag/Offset</b>	Extension Fragmentation	Créer une extension de fragmentation à partir des valeurs IPv4
<b>TTL</b>	Hop Limit	Décrémenter de 1
<b>Protocol</b>	Next Header	Recopier ou extension de fragmentation si besoin. ICMPv4 (1) devient ICMPv6 (58).
<b>Checksum</b>		Ignorer
<b>Source Address</b>	Source Address	Voir le paragraphe Traduction des adresses
<b>Destination Address</b>	Destination Address	Voir le paragraphe Traduction des adresses
<b>Options IPv4</b>		Les options IPv4 ne sont pas traduites.

# RFC 7915 : entête IPv4 à partir d'IPv6

<b>Champ de l'en-tête IPv6</b>	<b>Champ dans le nouvel en-tête IPv4</b>	<b>Valeur</b>
<b>Version</b>	Version	4
	IHL	5
<b>Traffic Class</b>	Type of Service	Recopie
<b>IPv6 Flowlabel</b>		Ignorer
<b>Payload Length</b>	Packet Length	Payload Length + IHL
	Ident./Flag/Offset	0
<b>Hop Limit</b>	TTL	Décrémenter de 1
<b>Next Header</b>	Protocol	Recopier. ICMPv6 (58) devient ICMPv4 (1)
	Checksum	Calculer une fois l'en-tête créé
<b>Source Address</b>	Source Address	Voir le paragraphe Traduction des adresses
<b>Destination Address</b>	Destination Address	Voir le paragraphe Traduction des adresses
<b>Extensions IPv6</b>		Les extensions d'en-tête IPv6 ne sont pas traduites.

# NAT64 : avec état (RFC 6146)

- Conversion de l'IP destination sans état (IPv4 incluse dans l'IPv6)
- Conversion IP sources IPv6 avec état : On associe l'IPv6 interne à l'IP externe à un No de port → usage d'une table
- Usage typique : accès de clients IPv6 only à des serveurs IPv4 only



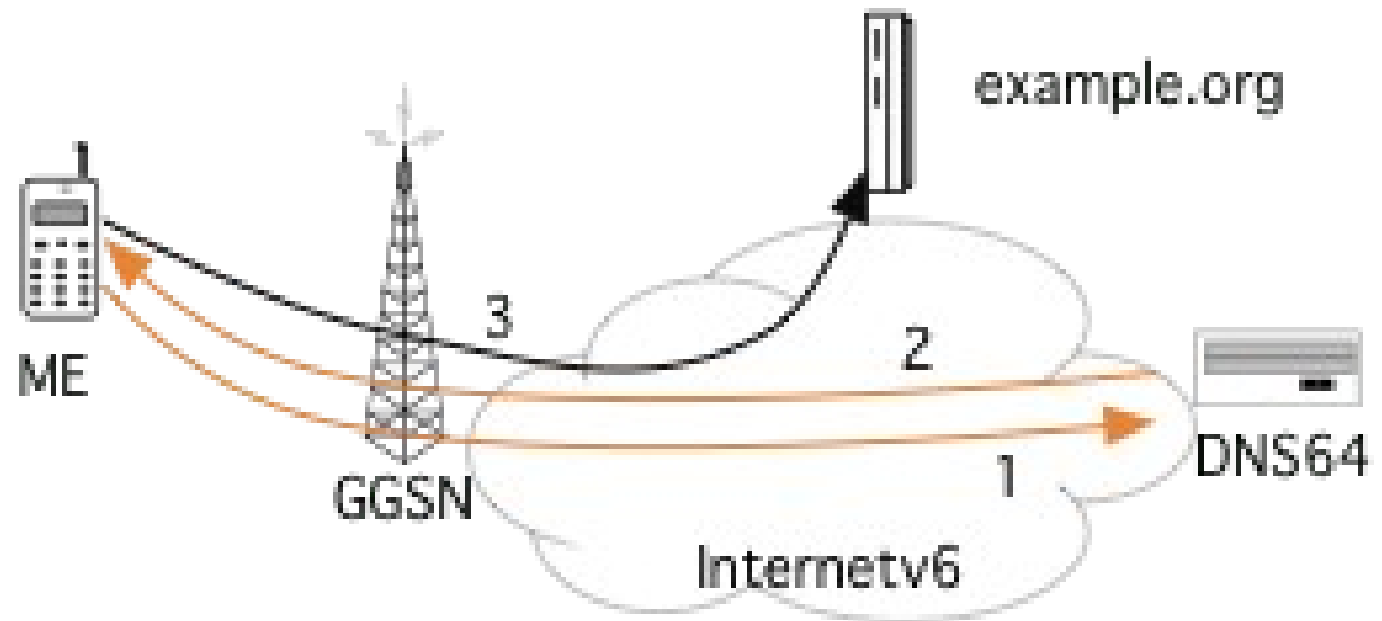
# NAT64 avec état : limites

- Les serveurs doivent être désignés par leur nom dns (pour DNS64)
- Applications et protocoles du clients doivent être compatibles IPv6
- Certains protocoles incluant l'IPv4 dans la charge utile sont incompatibles : FTP, SIP, Skype, MSN (source : [https://en.wikipedia.org/wiki/NAT64#cite\\_note-4](https://en.wikipedia.org/wiki/NAT64#cite_note-4))
- Solutions :
  - 464XLAT
    - Un convertisseur SIIT convertit le paquet IPv4 initial en paquet IPv6 (sur le client en général)
    - Transmis à la passerelle NAT64 qui le convertit en IPv4
    - SIIT : Stateless IP/ICMP Translation : conversion réversible d'un paquet IPv4 en IPv6 défini dans la rfc7915
  - Utiliser des passerelles applicatives dont le mandataire/proxyWeB est un cas particulier



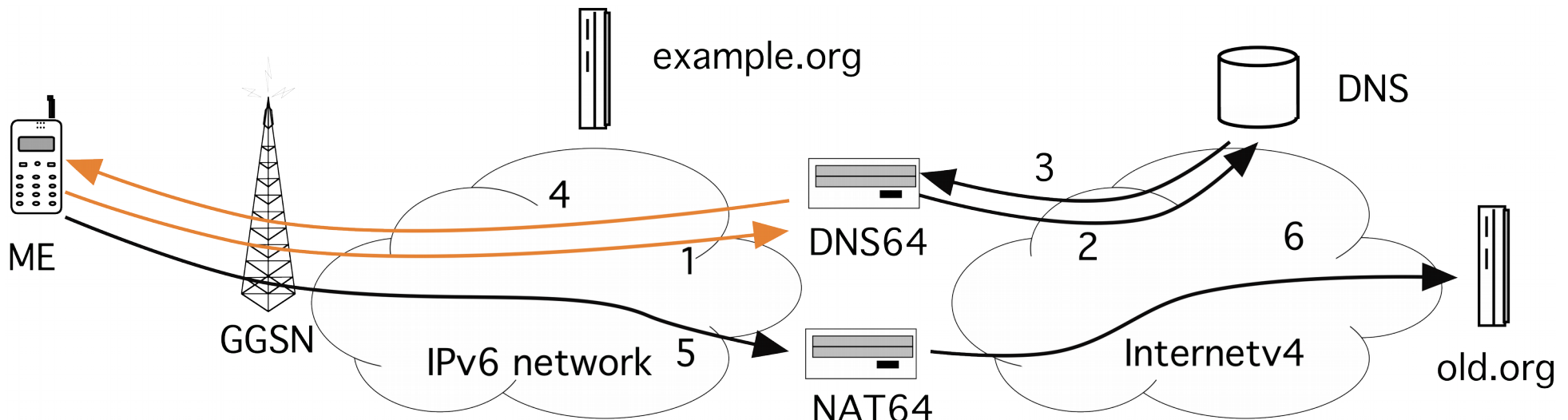
# NAT 64 : cinématique vers SRV IPv6

- 1 : requête DNS AAAA example .org
- 2 : réponse DNS AAAA example.org
- 3 : connexion au serveur en IPv6



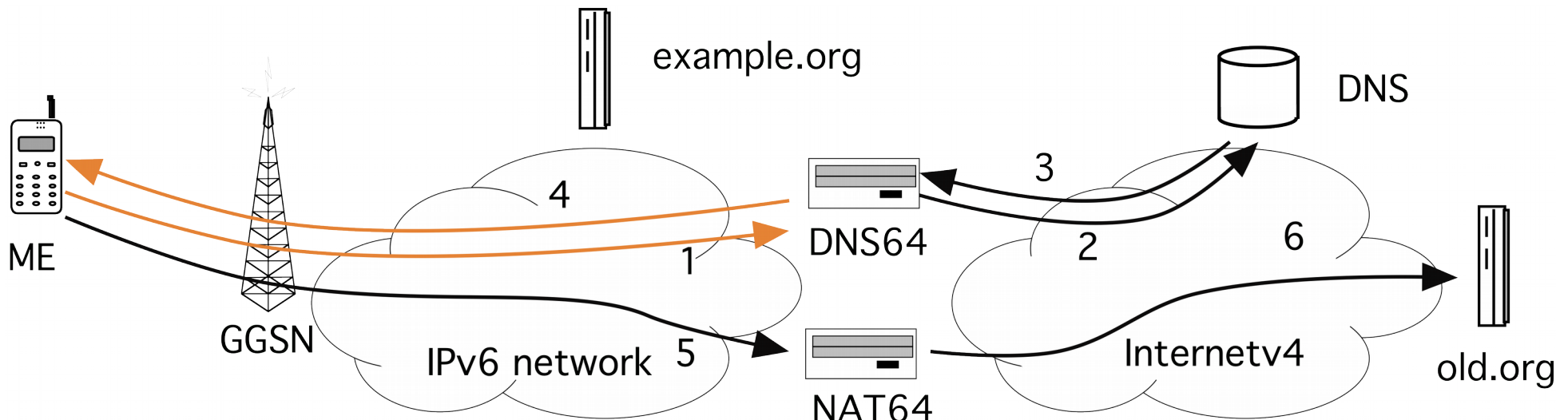
# NAT 64 : cinématique vers SRV IPv4

- 1 : ME interroge DNS64 sur AAAA old.org
- 2 et 3 : ME obtient les RR A et AAAA mais pas de AAAA
- 4 : transmission au client des IPv6 obtenues en transformant les IPv4 du serveur old.org (rfc 6052)



# NAT 64 : cinématique vers SRV IPv4

- 5 : le client se connecte à cette IPv6 appartenant au pool déclaré par NAT64
- 6 : NAT64 traduit cette IPv6 en IPv4 et
  - Note l'association port source, IPv4 source (l'une des siennes), IPv6 source dans une table
  - transmet à old.org



# NAT 64 : cinématique vers SRV IPv4

- Le paquet retour sera retourné par old.org à l'IPv4 de NAT64
- NAT64 utilisera sa table pour déterminer l'IPv6 interne destination
- Le paquet sera envoyé à ME

