

# Introduction à l'IPv6

N. Lebedev

CPE Lyon  
lebedev@cpe.fr

# Plan

## 1 Introduction

- Etat de l'art de l'IP
- Solutions d'urgence dans IPv4
- Capacité de l'espace d'adressage IPv4
- Ouvertures avec l'IPv6

## 2 Adresses IPv6 (RFC 4291)

- Format et types des adresses IPv6

## 3 Format du paquet IPv6

## 4 Protocoles réseaux et transport

## 5 Découverte de voisinage et autoconfiguration

- Découverte de voisins (RFC 2461)
- Configuration automatique. Exemple : config sans état.
- Configuration avec état : DHCPv6

## 6 Nommage : DNS pour IPv6

## 7 Migration vers IPv6

- Tunnels—mécanismes de transition et coexistence.
- Méthodologie de transition

## 8 IPv6 et les protocoles de routage

## 9 Evolutions

# Développement de l'IP

**RFC 791** : Internet Protocol, Sept. 1981, adresse en 32 bits.

1980s : Usage scientifique et militaire.

1990s : Usage commercial, croissance exponentielle ( $\times 2$  / an).

1993 : [RFC 1338] Epuisement de classe B était attendu, 1/3 d'espace d'adressage utilisé !

1994 : IPng (new generation) spécifié, adresse en 128 bits, IPv6.

2000s : Epuisement attendu des adresses IPv4 globales (routables).

**RFC 2460** : Internet Protocol Version 6 Specification.

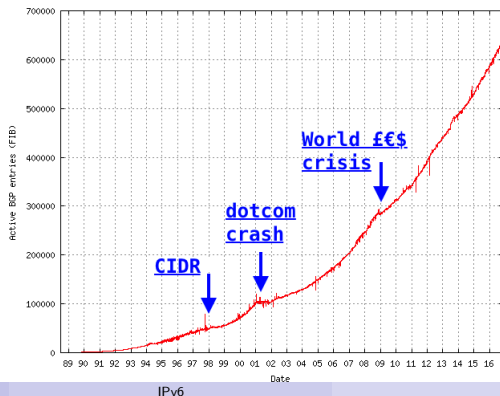
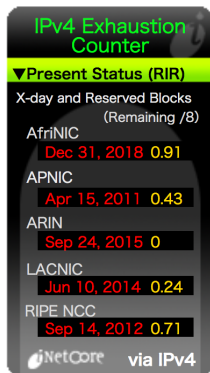
# IP : caractéristiques et raisons du succès

- **InterNet**working : couche 3 réseau.
- Indépendant de l'application (couches sups 4-7) et du support (couches liaison-2, physique-1).
- Communication de **bout en bout**
- Adressage unique hiérarchisée (préfixes IP)
- Service “**meilleur effort**” pour les datagrammes (paquets).
- Robuste mais pas fiable.
- Modèle client/serveur étendu : un poste peut être l'un ou/et l'autre.
- Protocoles de routage dynamiques—adaptabilité.
- Zones autonomes de routage, délégation de préfixes.
- DNS—nommage coordonné des domaines hiérarchisés.

# Insuffisances de l'IPv4 *classfull*

[RFC 1338]

- Pénurie d'adresses IPv4, dont Classe B. (Chine, Inde, Afrique) (Fig)
- Surcharge des tables (routage, MAC) : > 500.000 FIB entrées. (Fig)
- En-tête compliqué  $\Rightarrow$  routeurs surchargés.
- Nvx besoins IPv4 : boxes (IP dédiée), nouvelles applications avec adresses globales (VoIP), mobiles 4G, sécurité.



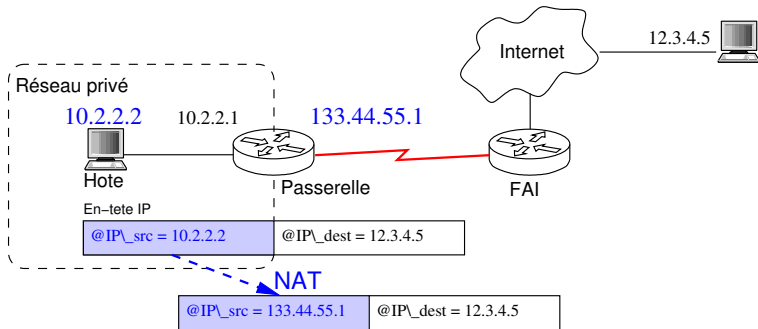
# Addressage privé NAT/PAT [RFC 1918]

- Espace d'adressage privé **grand**, non-routable vers l'extérieur.
- Partage de connexion par port, PAT : [IP\_addr:Port]
- Sécurité élémentaire native (adresses privées).
- ▼ Problème : empêche la “forte” sécurité basée sur la crypto (Nport chiffré).
- ✓ Solutions : NAT-T, IPsec-over-UDP [RFC-3948].

## Plages d'adresses réservées à l'usage privé.

- 1 réseau de classe A : 10.0.0.0 — 10.255.255.255 / 8
- 16 réseaux de classe B : 172.16.0.0 — 172.31.255.255 / 12
- 256 réseaux de classe C : 192.168.0.0 — 192.168.255.255 / 16

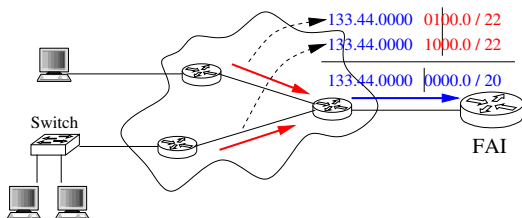
# NAT/PAT—Network/Port Addr Translation [RFC 3022]



## ▼ Inconvénients :

- "Etat" (*stateful*) : pas de transparence de bout en bout, pour l'appli qui la nécessite. Ex : SIP URI « bob@12.3.4.5 ».
- Client-Serveur implicite : cnx initiée depuis le réseau privé. Nécessite des techniques (ex : port forwarding) pour les connexions entrantes sur Web, FTP internes.

# CIDR—Classless Inter-Domain Routing



- Avant : autant d'entrées dans la TabR que de sous-réseaux alloués.
- Après : réduction des TabR des FAIs.
  - CIDR—agrégation des routes (préfixe commun), routage géographique (préfixes contigus=supernets),
  - VLSM : abolition des classes A, B et C, mais masque obligatoire !

NB :

Le masque est aussi obligatoire dans le résumé des routes. Ex : 133.44.0.0/20.



# Capacité IPv4

## Exemple :

164.107.134.5 = 10100100.01101011.10000110.00000101 (32 bits)

- Nb maximal d'adresses à plat  $< 2^{32} \approx 4$  Milliards.
- Classe A : 15 Millions d'adresses.
- Classe B : 64.000 (en réalité bcp moins).
- Classe C : 250
- Espace d'adressage total vu par les routeurs  $\approx 1$  Milliard.

$2^{32}$	4.294.967.296
– Class D,E (multicast, expérimentations)	-536.870.912
– Sous-réseaux 0, 127 (null, loopback)	-33.554.432
– Adresses privées (RFC 1918)	-17.891.328
<b>Disponibles</b>	<b>3.706.650.624</b>

# Combien d'adresses pour le IPng (nouvelle génération ?)

2020 : population de la Terre  $\approx 10$  milliards.

- Admettons 100(!) nœuds IP / personne  $\Rightarrow 10^{12}$  nœuds
- Plusieurs interfaces / nœud.
- Plusieurs adresses IPv6 / interface ( $\approx 10$  par nœud).
- Prenons de la marge —  $10^{15}$  adresses.
- Besoin étendu :  $10^{15}$  adresses terminales et  $10^{15}$  préfixes réseaux.
- Nouveaux domaines : mobiles 4G, IoT, 5G, capteurs, domotique, bâtiment, automobile...

Capacité d'adressage IPv6 est encore plus grande !

Pourrait être  $\approx 6 \times 10^{23}$  adresses / m<sup>2</sup> de surface terrestre (océans compris) !

# IPv6 en bref

- Espaces d'adressage "illimité" :  $2^{128} \approx 3,4 \times 10^{38}$  adresses disponibles.
  - VLSM @IP/masque, agrégation des adresses (CIDR).
  - Autoconfiguration—sans routeur ni serveur DHCP.
  - Multi-hébergement facilité (pas de rénumérotation interne ! )
  - Transparence (vs NAT).
  - IPsec, QoS et multicast sont natifs (champ "scope").
  - Routage plus facile via le champ **"flux"** donné.
  - Entête de taille fixe, simplifiés—traitement plus rapide.
  - Identification  $\neq$  localisation, facilité pour applications mobiles.
  - Extension des fonctionnalités (protocole, etc) issus de l'IPv4.
- 
- **Coexistence avec IPv4 *dual-stack*; adoption IPv6**

# Nouvelles applications

- VoIP.
- Embarqué.
- IoT, Internet des objets.
- Téléphonie 4G, réseaux 5G.
- Mobilité (garder la même @IP partout)
- Sécurité en général.
- Domotique et réseaux mobiles de données (capteurs, automobiles,...)
- Vidéoconférence.
- Jeux en réseaux.
- Interface haptiques, télé médecine.

# Taille et format d'une adresse IPv6 [RFC 4291, 3513-old]

- **128 bits.**
- **Notation canonique** hexadécimale en 8 blocs de 16 bits.

**2001 : 0660 : 3001 : 4002 : 0000 : 0000 : 0000 : 0007** ([www.renater.fr](http://www.renater.fr))

- Au début de chaque bloc, on peut supprimer de 1 à 3 "0"s.

**2001 : 660 : 3001 : 4002 : 0 : 0 : 0 : 7**

- **Une seule fois** les blocs successifs de "0000"s s'abrègent en "::".

**2001 : 660 : 3001 : 4002 :: 0 : 7**

**2001 : 660 : 3001 : 4002 : 0 :: 7**

- **Exemple d'une ERREUR :**

**2001 :: beda :: 7**

# Préfixes IPv6

- Un préfixe par lien local (segment LAN). Pls préfixes par LAN possibles.
- La notion de préfixes développée par CIDR est reprise.
- Notation :

**Préfixe IPv6 / longueur du préfixe**

- Exemples :

**2001 :: /16**

**2001 : 0660 :: /32**

- Une adresse fait partie d'un réseau dont le préfixe est de longueur déterminée :

Adresse hôte (www.renater.fr) : **2001 : 0660 : 3001 : 4002 :: 7/48**

Adresse sous-réseau = préfixe : **2001 : 0660 : 3001 :: /48**

# Types d'adresses : \*-cast

- Unicast : adresse d'une interface seule. Peut être :
  - à portée globale, routable ;
  - à portée lien locale, non-routable.
- Multicast (diffusion restreinte multidestinataire) : adresse pour un ensemble des interfaces sur les **nœuds différents** locaux ou distants.
- Anycast : adresse pour un ensemble des interfaces (nœuds différents). Destinataire est **une seule interface la plus proche au sens du routage**.

NB :

Il n'y a plus de **broadcast** en IPv6 !

NB :

L'adresse IPv6 est attribuée à l'interface (et non au nœud).

# Principaux types d'adresses IPv6

Définis par les bits de poids fort (à gauche) :

Type d'adresse	Préfixe binaire	Notation IPv6
Route par défaut	00...0 (128 bits)	:: / <b>0</b>
Indéfinie (démarrage poste)	00...0 (128 bits)	:: / <b>128</b>
Boucle locale (loopback)	00...1 (128 bits)	:: <b>1</b> /128
Multicast	11111111	<b>ff00</b> :: / <b>8</b>
Unicast lien-local (autoconf)	1111111010	<b>fe80</b> :: / <b>10</b>
Unicast global = Internet	0010 - 0011	<b>2000</b> – <b>3FFF</b> :: / <b>3</b>

## NB :

- Nœud :  $N$  interfaces  $\times$   $M$  adresses par interface (unicast, multicast ou anycast).
- Adresse définit la portée (limite d'unicité) : lien ou globale.



# Unicast globale [RFC 3587]

3 bits			45 bits				16 bits				64 bits															
0	0	1	Global prefix				Subnet ID				interface ID															

- **Internet IPv6 = sous-réseaux (2000 – 3FFF :: /3)**, seuls routables.
- **64 bits** préfixe  $\approx$  adresse de sous-réseau en IPv4 :
  - /23—Registrars (RIRs)
  - /32—FAIs
  - /48—topologie publique allouée par le FAI au client.
  - /64 (16 bits)—site, adresse de sous-réseau, ID du segment.
- **64 bits**—ID de l'interface  $\approx$  bits "hôte" en IPv4.

Consigne d'usage : 3-1-4 = (site-subnet-host) blocs de 16 bits

C'est la base pour créer des sous-réseaux plus détaillés.

# Bonnes pratiques

3 bits			45 bits				16 bits				64 bits							
0	0	1	Global prefix				Subnet ID				interface ID							

ID des sous-réseaux en VLSM—pas toujours pratique. A faire :

Sous-réseau par 4 bits *"Subnet on a nibble"*

- Blocs de 4 bits (chiffre hexa-)—masques faciles à utiliser (multiples de 4) :  
**(/64, /60, /56, /52, /48**
- **⇒ Subnet-ID en hexa- : 0001, 0002...)**

$N^{\circ}\text{-VLAN} = N^{\circ}\text{ Subnet-ID}$

(pareil qu'en IPv4)

FAI : parfois attribue un préfixe /56

# Internet IPv6

Variantes du plan d'adressage avec différents préfixes (cf : liste à jour<sup>1</sup>).

- **Permanents** déléguées aux RIR. Ex : **RIPE NCC**
  - **2a00 :: /12**,
  - **2003 : 0000 :: /18**,
  - **2001 : 2000 :: /20**,...
- **Tunnel 6to4 (2002 :: /16)** permettant d'acheminer le trafic IPv6 via le(s) réseau(x) IPv4. Depuis 01/02/2001.
- Toutes les autres adresses routables (> 3/4 de l'espace d'adressage) —réservées pour usage ultérieur.
- 6bone (**3ffe :: /16**) était le préfixe en expérimentation des interconnexions des réseaux IPv6 (fin d'opération 06/06/2006).

---

1. <http://www.iana.org/assignments/ipv6-unicast-address-assignments>

# Unicast lien-local

10 bits	54 bits	64 bits
1111111010	0	interface ID

- **FE80 :: /10**
- Adressage sur un segment LAN local : Ethernet, PPP, Tunnel.
- Usage : découverte de voisins (hôtes et routeurs).
- Le durée du bail est illimitée.
- Autoconf à l'initialisation de l'interface. Routage next-hop sur lien.

```
tp@lxFull:~$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:1c:a7:ab
          adr inet6: fe80::a00:27ff:fe1c:a7ab/64 Scope: Lien
```

**NB**

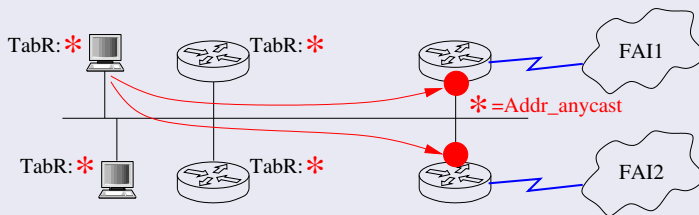
**Jamais routées ⇒ pas besoin du masque !**

# Anycast

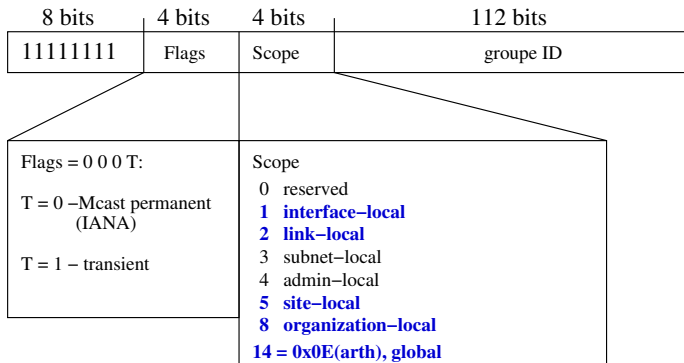
- **Même adresse de plusieurs interfaces sur les nœuds différents !**
- Adresses **destination** "la plus proche" au sens du routage (dynamique).
- Usage : interrogation des serveurs, DNS, DHCP, etc.

Ex :

- Interfaces des routeurs dans le subnet ; mais un seul choisi par son @MAC (ND) comme passerelle.



# Multicast



- ID pour le groupe des interfaces sur les nœuds différents.
- Chaque interface peut faire partie de plusieurs groupes multicast (group\_ID)
- Jamais utilisées comme addr\_src.
- Adresses multicast réservées [RFC 4291]

# Exemples d'adresses Multicast

**FF00 :: — FF0F ::** sont réservés, gérés par IANA. Exemples (sur le lien) :

- **FF02 :: 1** tous les nœuds sur le lien (lien-local)
- **FF02 :: 2** tous les routeurs sur le lien
- **FF02 :: 5** tous les routeurs OSPF sur le lien
- **FF02 :: 6** tous les routeurs désignés (DR) OSPF sur le lien
- **FF02 :: 9** tous les routeurs RIP sur le lien

**FF02 :: 1 : FF/104 XX : XXXX** adresse Multicast du nœud sollicité.

# Adresses obligatoires pour un nœud

## Hôte

- Link-Local **FE80 :: /10** pour chaque interface, auto-identification.
- Loopback :: **1**.
- Multicast, chaque interface :  
**FF02 :: 1** tous les nœud ;  
**FF02 :: 2** tous les routeurs ;  
**FF02 :: 1 : FF/104XX : XXXX** nœud sollicité pour chaque Unicast ou Anycast
- (PIs!) autres Unicast Globales ou Anycast sur chaque iface.
- (Option : autres Multicast pour les groupes du nœud.)

## Routeur en +

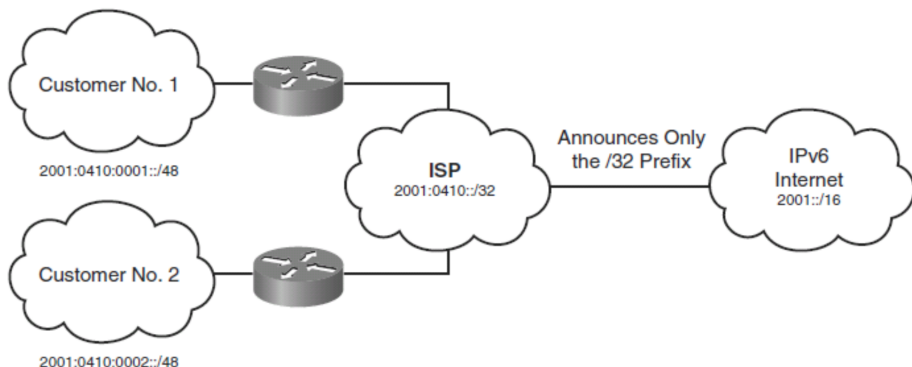
- (PIs!) Unicast Globales —préfixe(s) pour les lien(s) auxquels l'hôte est connecté
- Anycast sur les interfaces où il est "routeur du sous-réseau" (=passerelle par défaut)
- Multicast **ff02 :: 2**, tous les routeurs.



# Résumé sur l'allocation des adresses IPv6 [RFC 3513]

Allocation	Préfixe 8 bits (bin)	Préfixe (héxa)	Fraction
Reserved (trans, loopback)	0000 0000	00xx :: /8	1/256
Unassigned	0000 0001	01xx :: /8	1/256
Reserved (was NSAP)	0000 001	02xx :: /7	1/128 [RFC4048]
Reserved (was IPX Novell)	0000 01	04xx :: /6	1/64
Unassigned	0000 1	08xx :: /5	1/32
Unassigned	0001	1xxx :: /4	1/16
<b>Global Unicast</b>	<b>001</b>	<b>(2 – 3)xxx :: /3</b>	<b>1/8 [RFC3513]</b>
Unassigned	010–110	(4 – D)xxx :: /3	5/8
Unassigned	1110	Exxx :: /4	1/16
Unassigned	1111 0	F(0 – 7)xx :: /5	1/32
Unassigned	1111 10	F(8 – B)xx :: /6	1/64
<b>Unique Local Address (ULA) (private)</b>	<b>1111 110</b>	<b>F(C – D)xx :: /7</b>	<b>1/128</b>
Unassigned	1111 1110 00	FE(0 – 7)x :: /9	1/512
<b>Link-Local Unicast</b>	<b>1111 1110 10</b>	<b>FE(8 – B)x :: /10</b>	<b>1/1024</b>
Deprecated (was site-local)	1111 1110 11	FE(C – F)x :: /10	1/1024
<b>Multicast Addresses</b>	<b>1111 1111</b>	<b>FFxx :: /8</b>	<b>1/256</b>

# Global Unicast (suite)



- Dans le préfixe unicast global (2001 : :/16).
- Agrégation des préfixes depuis clients vers FAIs

# IDs des interfaces

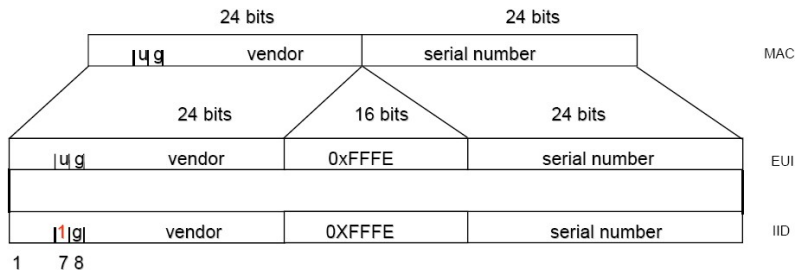
- Respect EUI-64 bits format (compatibilité Firewire IEEE 1394, IEEE 802.15.4).
- Méthodes de génération :
  - ① Basé sur L2 adresse @MAC—préconisation. Portée universelle.
  - ② Aléatoire [RFC 3041].
  - ③ Cryptographique, basée sur la Clé Publique [RFC 3972].
  - ④ Manuelle.
- ID à portée locale possibles :
  - Lorsque l'@ MAC est indisponible (liaison série, extrémité du tunnel)
  - Voir "IPv6 over <Protocole\_L2>"
- Usage : poste client, pas adapté aux serveurs.

## NB

- ID de l'interface est unique pour chaque préfixe (subnet).
- ... mais la même ID de l'interface avec différent préfixes possible.



# Transition @MAC → EUI-64



- Utilisé pour les types d'adresses *global* ou *lien-local*.

# IPv4 contenues dans IPv6

## Adresses IPv6 mappées sur IPv4 :

Représentation d'un hôte IPv4 (réseau) comme IPv6 (pour appli).

Usage : sockets créés par un daemon IPv6, qui accepte (bind) une @ IPv4.

NB : n'est pas supporté par toutes les implémentations  $\Rightarrow$  separate sockets.

**:: ffff : a.b.c.d/96**

## Adresse IPv6 compatible avec IPv4 (obsolète) :

Tunnels dynamique (et automatique) IPv6 dans le réseau IPv4 (6to4).

**:: 0000 : a.b.c.d/96**

# Adresses IPv6 et URL [RFC 2732]

- Dans une URL, une adresse IPv6 doit être encadré de [ ] :

**http : //[3ffe : 0 : a88 : 85a3 :: ac1f : 8001]/index.html**

- Lève l'ambiguïté si N° de port est présent :

**http : //[3ffe : 0 : a88 : 85a3 :: ac1f] : 8001/index.html**

## NB :

L'URL n'est pas l'unique cas où cette différence est importante :

- XWindow—système de fenêtrage Unix/Linux.
- Protocole SIP de signalisation téléphonique (VoIP).

# Rappel : entête IPv4

Version	IHL	ToS	Total Length	
Identification			Flgs	Fragment Offset
Time To Live	Protocol		Header Checksum	
Source Address				
Destination Address				
Options				Padding

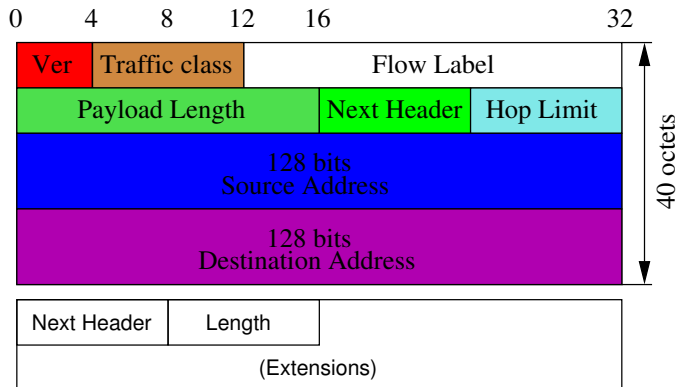
# IPv6 vs IPv4

- Champs conservés :
  - Version (4 bits).
  - TOS (Type Of Service)  $\Rightarrow$  Traffic class (1 octet).
  - Longueur du paquet  $\Rightarrow$  Longueur de la partie données (2 octets).
  - Protocol type  $\Rightarrow$  Next Header (1 octet)
  - TTL (Time To Live)  $\Rightarrow$  Hop Limit (1 octet)
- Champs supprimés :
  - Longueur de l'entête (car entête de taille fixe dans l'IPv6)
  - Champs de fragmentation (ID, Flags, Offset), car **fragmentation par la source**.
  - Somme de contrôle sur l'entête (header checksum).
- Rajouté : "Flow label" (20 bits).
- IPv4 options  $\Rightarrow$  IPv6 entêtes d'extensions.
  - Extensions (sauf "hop-by-hop") sont analysées uniquement par le destinataire, rapide !



# En-tête IPv6

- En-tête *de taille fixe* : 40 octets, 3,8% MTU (cf IPv4 : entête variable 20—60 octets avec les options, 3,4% MTU).
- MTU = 1280 octets, mécanisme de découverte (si MTU > 1280)



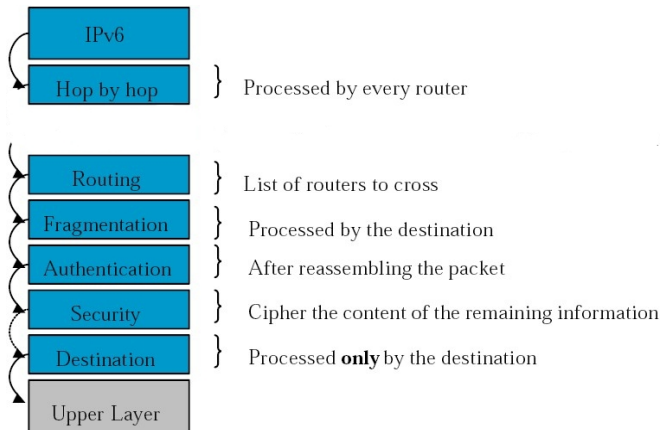
# En-têtes étendus de l'IPv6 (extensions)

- Champ "Next Header" pour extension OU protocole encapsulé.
- Ignorés par les routeurs intermédiaires (exception—"Hop-by-Hop" Options header "Next Header = 0").

Valeur du champ "Next Header"	Extension	Valeur	Protocole
0	<b>Proche-en-proche</b>	4	IPv4
43	Routage	6	TCP
44	Fragmentation	17	UDP
51	Authentification	58	ICMPv6
50	Confidentialité	41	IPv6
59	Fin des entêtes		
60	Destination		

# Ordre des entêtes d'extension est important

Implémentation complète doit inclure tout !



NB

Entête “destination” peut aussi suivre “Routage”, traité par chaque routeur.

# Spécification des types de flux en IPv6 [RFC 6437]

## DEF : "Flux"

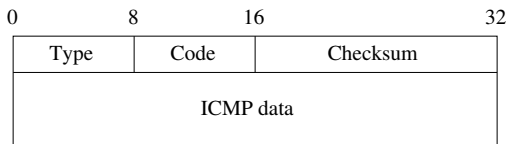
Séquence de paquets src-dest pour lesquels la source souhaite un traitement particulier (QoS par RSVP, TRéel,...). Ex : flux applicatif TRéel.

- ID du flux existait déjà en IPv4 : socket (mais rarement utilisé)  
 $ID\_FLUX = (IP\_src, N^o\_Port\_src, IP\_dest, N^o\_Port\_dest, Flow\_label)$
- IPv6 :
  - $ID\_FLUX = RAND()$ , car  $N^o\_Port$  inconnu si cryptage.
  - $ID\_FLUX = (Addr\_src, Addr\_dest, Flow\_label)$ .
- **Crée le "contexte", pour éviter de lire la TabR pour chaque paquet...** et les entêtes d'extension !
- Paquet hors flux :  $Flow\_label=0$ .

# ICMPv6 [RFC 2463]

- Fonctions améliorées (vs ICMP) :
  - détection d'erreurs (inaccessibilité d'un nœud / lien / port) ;
  - tests (ping, traceroute) ;
  - *autoconfiguration (découverte des routeurs,...) ⇒ protocole ND—Neighbor Discovery* ;
  - reprise de fonction de l'ARP via multicast.
- Nouvelles fonctionnalités dans ICMPv6 :
  - *gestion des groupes de multicast MLD ⇒ protocole MLD—Multicast Listener Discovery*

# Format de l'en-tête et type de messages



Type	
Valeur	Usage
1,2,3 et 4 (< 127)	Gestion des erreurs
<b>128,129</b>	<b>Demande/Réponse d'echo (ping6)</b>
130–132	Gestion multicast (MLD for IPv6, [RFC 2710])
<b>133–137</b>	<b>Découverte de voisins (ND)</b>
144–147	Mobilité [RFC 3775]

- Type : nature du message
- Code : cause

# Fonctionnalités du protocole *Neighbor Discovery (ND)*

- Autoconfiguration d'adresses.
  - Détection de duplication d'adresse (DAD).
  - Découverte de(s) routeur(s), passerelles.
  - Découverte de(s) préfixe(s).
  - (Opt : découv. des params du lien physique (linkMTU, Max Hop Limit)).
- Résolution d'adresses par ICMPv6 (equiv. ARP).
- Détection d'inaccessibilité des voisins.
- Redirection : meilleure route vers la destination.

## NB :

- Le protocole ND est utilisé sur un segment réseau (lien local)
- C'est un hybride d' « auto-DHCP » et d'ARP, mais :
  - sans état, ni diffusion (les routeurs ne mémorisent pas les choix des postes)
  - il n'y a pas de bail.

# Messages ICMPv6 pour la *ND—Neighbor Discovery*

Type	Usage	Données
133	Sollicitation du routeur (RS)	<ul style="list-style-type: none"> <li>• Au démarrage du poste sur Mcast routeur <b>ff02::2</b></li> </ul>
134	Annonce du routeur (RA)	<ul style="list-style-type: none"> <li>• liste des préfixes sur le lien (<math>\neq</math> DHCP)</li> <li>• Max Hop Limit (sinon 64/défaut)</li> <li>• linkMTU.</li> </ul>
135	Sollicitation d'un voisin (NS)	<ul style="list-style-type: none"> <li>• détection de duplication d'@</li> <li>• déterminer l'@ MAC d'un voisin (<math>\approx</math> ARP)</li> <li>• ou vérifier son (in)accessibilité (<math>\approx</math> ping6)</li> </ul>
136	Annonce d'un voisin (NA)	<ul style="list-style-type: none"> <li>• réponse à un NS,</li> <li>• annonce d'un changement d'@ IPv6</li> </ul>
137	Redirection	Routeur annonce le meilleur chemin.



# Résolution d'adresse L2 (MAC) : rappel.

## Problème

Trouver l'@ MAC à partir de l'@ IP\_cible.

## Solution IPv4 & ARP

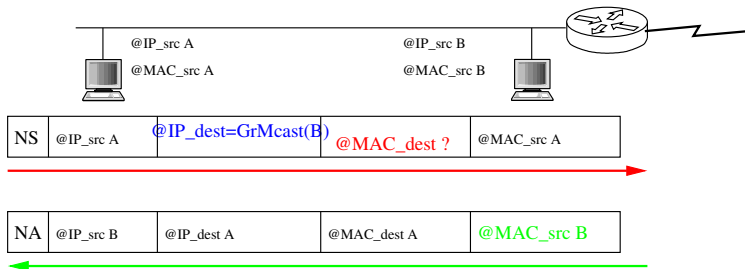
- Diffusion de la requête ARP contenant :  
(dest = FF-FF-FF-FF-FF-FF, src = @ MAC\_src)
- Unicast de la réponse ARP vers l'@ MAC\_src de la requête :  
(dest=@ MAC\_src, src = @ MAC\_cible)
- Table ARP poste source :

Cache ARP		
@ IP_cible	@ MAC_cible	Interface de sortie

- Q : Est-ce efficace ?

# Résolution d'adresse L2 (pour IPv6)

- Au démarrage, chaque nœud *doit* joindre 2 groupes spéciaux de multicast *pour chaque interface réseau*.
  - All-nodes multicast link-local groupe (M1) : **ff02 :: 1**
  - Groupe multicast du nœud sollicité (M2) : **ff02 :: 1 : ff/104XX : XXXX**
- ND[NS]—sollicitation du voisin est transmis à M2 de dest. (pas d'AR!)
- ND[NA]—annonce du voisin.



# Dérivation de l'@ multicast du nœud sollicité [RFC3307]

Procédé pour Ethernet

@ IPv6_dest	2001 : 0660 : 3001 : 4002 : 4421 : 21ff : fe	24 : 87c1
	↓ ↓	↓ ↓
@ IP Multicast sollicitée	ff02 :: 1 : ff	24 : 87c1
	↓ ↓	↓ ↓
@ MAC multi- cast	33 : 33 : ff	24 : 87c1

Rappel MAC

Bit 7 = "0/**1**" global/**local**  
 Bit 8 = "0/**1**" unicast/**multicast**

# Objectifs d'autoconfiguration

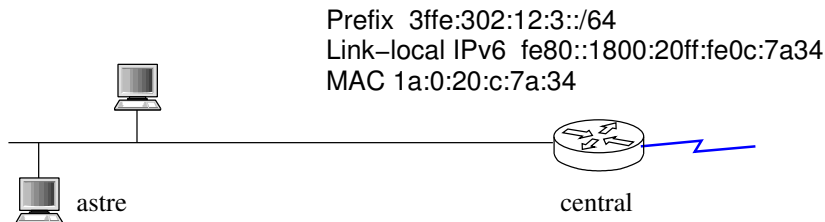
Activation "plug-and-play" de l'interface réseau

- Auto-attribution de l'adresse lien-locale. Puis :
  - Vérifier son unicité sur le lien  $\Rightarrow$  comm locale.
  - Déterminer la(es) @ unicast globales par une des deux méthodes.
- **Méthodes dynamiques :**
  - ① **"Sans état" (stateless).** L'adresse est construite à partir de :
    - Préfixe annoncé par le(s) routeur(s) dans les *RA—Router Advertisements*.
    - @ MAC de l'interface.
  - ② **"Avec état" (DHCPv6),** l'attribution est strictement contrôlée.
    - But : attribuer les mêmes adresses, ex : serveurs.

NB :

S'il y a un routeur sur le lien, la machine doit appliquer la méthode indiquée dans le RA.

# Segment du LAN



Local-link IPv6 fe80::0a00:20ff:fe0a:aa6d ?

Global unicast IPv6 3ffe:302:12:3:a00:20ff:fe0a:aa6d

MAC 8:0:20:a:aa:6d

# Initialisation et Détection d'Adresse Dupliquée (DAD)

## Ethernet

Src: 08:00:20:0a:aa:6d

Dst: **33:33:ff:0a:aa:6d** (multicast L2 Ethernet du dest)

Type: 0x86dd (@IPv6)

## IPv6

Version: 6    Priorité: 0xf0    Label: 000000

Longueur: 24 octets (0x0018)    Protocole: 58 (0x3a, ICMPv6)

Nombre de sauts: 255 (0xff) (provient du réseau local)

@IPv6 Source: :: (non-spécifiée)

@IPv6 Dest: **ff02::1:ff0a:aa6d** (multicast sollicité de l'@ cible)

## ICMPv6

Type: 135 (0x87, NS - sollicit. d'un voisin)

Code: 0 Checksum : 0xfe37

Cible: **fe80::a00:20ff:fe0a:aa6d** (soi-même - astre, lien-local)

# Annonce d'un voisin (NA) $\Rightarrow$ collision

## Ethernet

Src: 08:00:20:0a:aa:6d

Dst: **33:33:00:00:00:01**

Type: 0x86dd (@IPv6)

## IPv6

Version: 6    Priorité: 0xf0    Label: 000000

Longueur: 32 octets (0x0020)    Protocole: 58 (0x3a, ICMPv6)

Nombre de sauts : 255 (0x0ff)

Source: **fe80::a00:20ff:fe0a:aa6d** (La même que astre, "Cible" !)

Dest: **ff02::1** (multicast, tous les noeuds du lien)

## ICMPv6

Type: 136 (0x88, Annonce d'un voisin)    Code: 0    Checksum : 0xe036

Bits (0x7) R = 0 (not router) S = 0 (answer), 0 = 1 (overwrite cache)

**Cible: fe80::a00:20ff:fe0a:aa6d**

Option:    Type: 2 (Adresse physique cible)    Lg : 8 octets (0x01):

08:00:20:0a:aa:6d

Sinon, si pas eu de collision, suite à 2 NS, l'unicité de l'adresse serait confirmée.

# Sollicitation du routeur (RS)

## Ethernet

Src: 08:00:20:0a:aa:6d

Dst: **33:33:00:00:00:02** (MAC multicast de tous les routeurs)

Type: 0x86dd (@IPv6)

## IPv6

Version: 6 Priorité : 0xf0 Label: 000000

Longueur: 16 octets (0x0010) Protocole : 58 (0x3a, ICMPv6)

Nombre de sauts: 255 (0x0ff)

Source: fe80::a00:20ff:fe0a:aa6d (astre, lien-local)

Dest: **ff02::2** (multicast, tous les routeurs du lien)

## ICMPv6

Type: 133 (0x85, Sollicitation du routeur) Code: 0 Checksum: 0xd63e

Options: Type: 1 (Adresse physique source) Lg: 8 octets (0x01):

08:00:20:0a:aa:6d



# Annonce du routeur (RA)

3 possibilités :

- ① Routeur fournit les paramètres suffisant en IPv6 : Préfixe, sa Longueur, Passerelle par défaut.
- ② Mêmes + paramètres DNS à obtenir auprès du serveur DHCPv6
- ③ Routeur : message "Redirect". Le poste devra contacter le serveur DHCPv6 pour la configuration.

# Annonce du routeur (RA)

Commande :

```
R1(config)# ipv6 unicast-routing
```

## Ethernet

Src : 1a:00:20:0c:7a:34

Dst: **33:33:00:00:00:01** (MAC multicast de tous les postes)

Type: 0x86dd (@IPv6)

## IPv6 ...

Src: fe80::1800:20ff:fe0c:7a34 (central, lien-local)

Dest: **ff02::1** (multicast, tous les noeuds du lien)

## ICMPv6 ...

M=0 (not Managed by DHCP) O=0 H=0 (not Home agent for mobiles)

...

Options:

Type: 1 (Adresse phy src) Lg: 8 octets (0x01) : 1a:00:20:0c:7a:34

Type: 3 (Préfixe(s)) Lg: 32 octets (0x04)

Drapeaux: L=1 (même Lien) A=1 (Préfixe OK pour Adr Ucast Globale) R=1

Durée de validité : Durée de préféralité:

**Préfixe: 3ffe:302:12:3::/64**

# Principes de base DHCPv6

- Configuration plus riche (impossible par autoconf) : une 30-aîne de paramètres.
- Affectation pendant une durée (bail) :
  - attribution fixe : même IPv6 à la même MAC.
  - annonce des adresses IP des serveurs de noms (DNS).
  - TFTP pour les téléphones IP :  
tftp://A.B.C.D/var/lib/tftpboot/SIPphone/
  - découverte automatique des proxys web.
- Client-serveur DHCPv6 sur le lien ou via le Relais-DHCPv6.
- Complémentarité :
  - Autoconf sans état : unicast globale.
  - Avec état : serveur DHCP associe (*paramètres, état*) pour le client. Usage :
    - Contrôle stricte d'attribution, ex : même adresse à l'hôte (serveurs, imprimantes,...)
    - La RFC du DHCPv6 est maintenant stabilisée.

# Résumé autoconfiguration

- ① Création adresse IPv6 lien-locale.
- ② Détection des adresses dupliquée (DAD).
- ③ Envoi d'une requête au(x) routeur(s) sur leurs adresses multicast : sollicitation des routeurs (RS)
- ④ Obtention de(s) préfixe(s) global(ux) grâce aux annonces des routeurs (RA), création des IPv6 avec.
- ⑤ Détection des adresses dupliquée (DAD).
- ⑥ Configuration de la passerelle par défaut.

## NB

- Routeurs et équipements cœur de réseau : configuration manuelle.
- Stations : autoconfiguration.

## Important

Pb : Résolution nom ↔ adresse auto-configuration ??? Sol : MAJ DNS IPv6 automatique suite à l'auto-configuration.

# Rappel sur DNS

- Annuaire hiérarchique réparti.
- Un serveur de noms (NS, Name Server) gère le domaine : `cpe.fr`
- NS stocke les enregistrements (**RR, Resource Records**) permettant
  - Résolution `< Nom > ↔ < Adresse >`.
  - Résolution inverse.
  - Information sur les domaines, serveurs de noms, etc.
- Indispensable en IPv6 avec l'espace d'adressage et fonctions évolués. Pbs :
  - Auto-configuration est incompatible avec un plan d'adressage.
  - Besoin des adresses IPv6 fixes pour les serveurs, switches.
  - Sol : fonction hôte **MAJ automatique DNS** des enregistrements suite à l'attribution d'une configuration.

# Format des enregistrements IPv6

Deux nouvelles extensions :

- Enregistrement **AAAA** (quad A) pour la résolution directe.
- Toutes les adresses sont publiées : IPv4 et/ou IPv6.

6bone.net.	3600	IN	A	209.71.226.24
6bone.net.	3600	IN	AAAA	2001:5c0:0:2::24

- Enregistrement **PTR** stocké dans la sous-arborescence **ip6.arpa.** pour le nommage inverse.

<@IPv6 inversée par blocs de 4 bits en pointée>.ip6.arpa

- Délégation de zone, même qu'en IPv4.

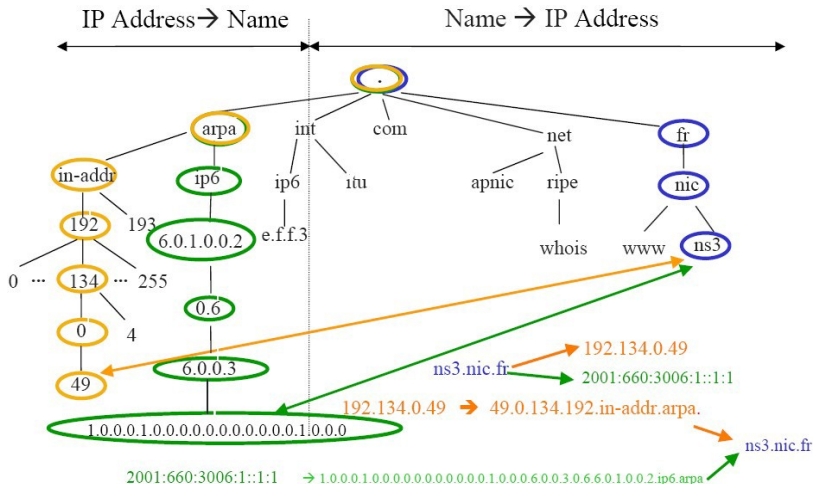
# Implémentations

- Support complet du DNS pour l'IPv6 : enregistrements AAAA et PTR sous l'arborescence `ip6.arpa` et transport IPv6 des messages DNS.
- Support partiel : uniquement les enregistrements AAAA et PTR.
- Certains produits logiciels sont uniquement client ou serveur.

## BIND 9 de l'ISC—Internet Systems Consortium

- Référence
- Complet : client (“resolver”), serveur, outils.
- Pour la majorité de plate-formes : Unix\*, Windows,...

## Arborescence DNS pour IPv6

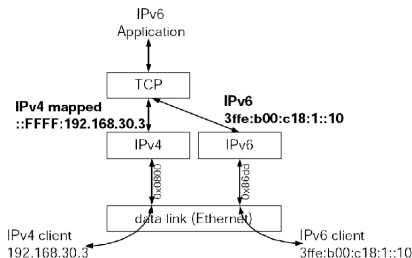




# Transition vers IPv6

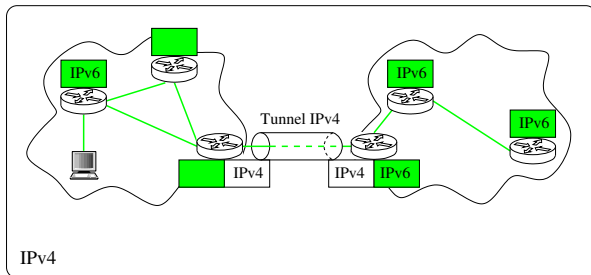
- Normalisation est finalisée.
- *Pas de jour "J"* ni basculement.
- Phase de coexistence, mais pas de méthode universelle.
- Mécanismes et techniques :
  - Double pile logicielle (*dual stack*) *IPv6* et *IPv4*.
  - *Encapsulation de IPv6 dans IPv4 (tunelling)*.
  - *Traduction des en-têtes IPv6 ↔ IPv4. NAT protocol translation (NAT-PT) between IPv6 and IPv4.*
- Motivation : pénurie des adresses et applications phares.

# Mécanisme double pile *Dual Stack*



- Réseau cœur : chaque nœud implémente deux piles IPv4 et IPv6.
- Stations : double pile dans l'environnement mixte.
- Interface : deux adresses, IPv4 et IPv6 !
- Applications socket IPv6 :
  - Peut dialoguer en IPv6 et/or IPv4.
  - Utilise les adresses mappées IPv4 pour communiquer avec les nœuds IPv4.
- Services :
  - Résolveur DNS (stub) retourne IPv6, IPv4 ou les deux.

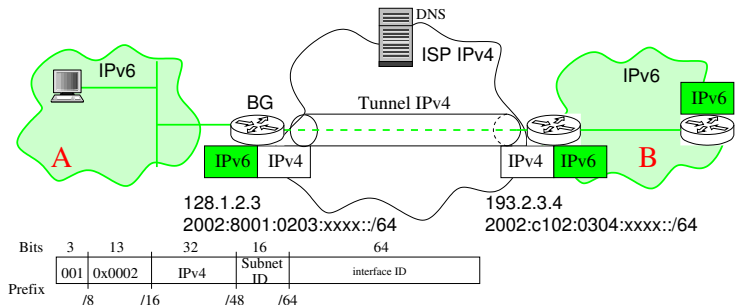
# Mécanisme et types de tunnels



- Manuel (static)
- Automatique
- Tunnel broker
- 6to4
- 6over4

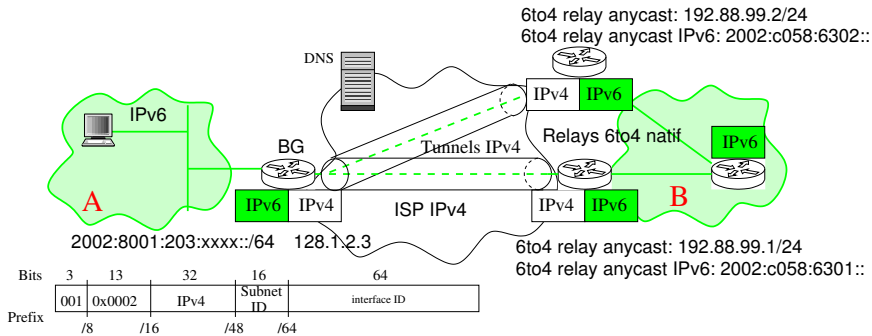
- Générique : IP (v4 ou v6) ou non-IP dans IPv6 [RFC 2473].
- Transport de IPv6 sur IPv4 (proto 41(0x29) en-tête IPv4).
  - Routeur ↔ Routeur, Routeur ↔ Hôte ou Hôte ↔ Hôte.
  - Connexion des îlots IPv6 au dorsal IPv6 à travers le transport IPv4.

## 6to4



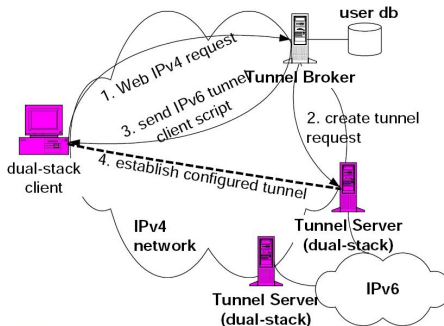
- Interconnexion de deux îlots IPv6 (6to4).
- Création automatique des tunnels IPv6 dans IPv4 (Protocole=41)
- 6to4 :
  - Le préfixe **2002 :: /16** a été alloué par l'IANA à ce type d'adressage.
  - Le préfixe /48 du site via l'@ IPv4 (32 bits) du routeur de frontière (BG).

## 6to4 natif



- Interconnexion des îlots IPv6 avec la dorsale IPv6 natif.
- Création automatique des tunnels IPv6 dans IPv4 (Protocole=41)
- Natif :
  - Routeur relais (ou tunnelier) 6to4, et **anycast** @IPv4=192.88.99.0/24, route par défaut vers Internet IPv6

# Tunnel broker



- Broker : fournisseur de service tunnel.
- Client : **hôte double pile isolé dans IPv4** à connecter en l'IPv6.
- Provisionnement : statique et dynamique, via
- *TSP—Tunnel Setup Protocol*.
- Fournissent les préfixes /48.

# Interface utilisateur pour la création du tunnel

- S'enregistrer : formulaire sur le site, email de confirmation etc (peut être loooong, qqs jours ?)
  - Choix de l'OS, fournir son adresse IPv4.
  - Authentification (ID, mot de passe).
- Demande du type (dyn/static) et création du tunnel par le serveur de tunnel.
- Télécharger/recevoir un script de configuration.
- Exécution du script crée le tunnel côté client. Ou installer l'appli
- Choisir le tunnel
- Liste avec PoP en France : Hurricane Electric, SixXS, Renater

# Etapes importantes

## Des couches basses vers les couches sups

- ❶ Equipement réseau : Routeurs, Firewalls, Switchs.
- ❷ Services “proches” matériel : **DNS**.
- ❸ Infrastructure IPv6 :
  - Adressage équipement et rajout entrée pour serveurs DNS.
  - Enregistrement sont encore de type A (IPv4).
- ❹ Migration de serveurs et de postes en double pile.
  - Par groupe de services liés
- ❺ Configuration des clients :
  - Installation des daemons pour l'autoconf (réponse aux sollicitations)
  - Sinon DHCPv6 si l'implémentation existe.



# Routage en IPv6

- **Static**
- **RIPng**
- **OSPFv3**
- IS-IS for IPv6
- EIGRP for IPv6
- Multiprotocol BGP version 4 (MP-BGPv4)

## NB :

Ces modes doivent être activés via

```
R# ipv6 unicast-routing
```

# Routage static

- Configuration  $\approx$  IPv4.
- Spécificités :
  - Le routeur voisin sur un lien est reconnue par son adresse lien-local.
  - Or, préférer celle-ci comme "prochain saut".
- Exemples :
  - Route directe  
`R-FAI(config)# ipv6 route 2001:CC1E::/32 serial 0/0/0`
  - Route recursive (2  $\times$  TabR : interface)  
`R-FAI(config)# ipv6 route 2001:CC1E::/32 FE80:CC1E::2`
  - Route complète : interface de sortie ET prochain saut. Usage : interface accès multiple (Ethernet)  
`R-FAI(config)# ipv6 route 2001:CAFE::/32 FastEthernet0/0  
FE80:CAFE::2`
  - Route par défaut :  
`R-Home(config)# ipv6 route ::/0 s0/1/0`

# RIPng (new generation), [RFC 2080]

- Basé sur IPv4 RIP version 2 (RIPv2).

=

- Nombre de sauts max est 15.
- Distance administrative (priorité) est 120.
- Utilise « split horizon » et « route inverse empoisonnée » pour empêcher les boucles de routage.

≠

- Echanges des préfixes et réseaux IPv6 uniquement.
- L'adresse du prochain saut est en IPv6.
- Port UDP 520 (au lieu de port UDP 521).
- Groupe de multicast FF02 ::9 (au lieu de 224.0.0.9).

NB :

RIPng est activé sur une INTERFACE

# Exemple activation RIP

- 1 Activer le processus RIPng sur une **interface**, avec un nom "RIPfa00".

```
R1(config)# ipv6 unicast-routing
R1(config)# int fa0/0
R1(config-if)# ipv6 rip ?
WORD    User selected string identifying this RIP process

R1(config-if)# ipv6 rip RIPfa00 ?
default-information  Configure handling of default route
enable              Enable/disable RIP routing
...

R1(config-if)# ipv6 rip RIPfa00 enable
```

- 2 Configurer le processus RIPng, désactiver le clivage d'horizon.

```
R1(config)# ipv6 router rip RIPfa00
R1(config-rtr)#?
...
distribute-list      Filter networks in routing updates
poison-reverse        Poison reverse updates
redistribute          Redistribute IPv6 prefixes from another routing prot.
split-horizon         Split horizon updates
...

R1(config-rtr)# no split-horizon
```

# OSPFv3 pour IPv6

- Basé sur OSPF version 2 (OSPFv2), mais indépendant
- Distribue les préfixes IPv6.
- Encapsulation dans IPv6 en natif.
- Spécificités :
  - Adresses en 128 bits.
  - Quelques adresses spécifiques : lien-locale **FE80 ::**.
  - Plusieurs adresses par interface.
  - Authentification (IPsec).
  - Unité OSPFv3 est le lien, pas le sous-réseau.

# OSPF vs OSPFv3 pour IPv6

=

- Types de paquets : Hello, DBD, LSR, LSU, LSA.
- Découverte de voisin et adjacence.
- Algorithme SPF : variante de Dijkstra.
- Election du DR.
- Aires multiples, topologies différentes (NBMA, point-to-multipoint, point-to-point and broadcast).

≠

- "Link" équivalent à "subnet" ou "network".
- OSPFv3 est configuré sur une interface.
- Multiple instances OSPFv3 par interface.
- Nouvelles adresses multicast.

NB :

Router-ID est toujours en 32-bits !

# Exemple OSPFv3

```
R1(config)# ipv6 unicast-routing
R1(config)# int fa0/0
R1(config-if)# ipv6 ospf ?
    <1-65535>          Process ID
    authentication     Enable authentication
    cost               Interface cost
    ...
    network            Network type
    priority            Router priority

R1(config-if)# ipv6 ospf 10 ?
    area  Set the OSPF area ID
R1(config-if)# ipv6 ospf 10 area 0
```

# Implémentations

- Mac OS X 10.2 Jaguar
- Unices : FreeBSD, OpenBSD, IRIX, Linux (kernel > 2.2), Solaris, HP-UX,...
- Windows : Windows Server 2003, Windows XP with Service Pack 1,2,3 ; Microsoft Windows CE .NET.
- Routeurs : Cisco (IOS > 12.2(2)T), Huawei, HP, Hitachi, Nortel Networks, Juniper Networks,...
- Firewall IPv6 avec accélération matérielle : Checkpoint, Juniper, Cisco,...



# Applications

- ipv6gen pour générer les préfixes
- Projet WIDE : <http://www.wide.ad.jp/>.
- VideoLAN Client and Server (VLC).
- openssh.
- www Server : Apache, tthttpd (native!).
- www Client : Mozilla, Konqueror (KDE), Lynx, IE.
- OpenLDAP.
- Wireshark.
- OpenSSH.
- BIND.

# Liens utiles

## Actifs

<http://livre.g6.asso.fr> , "IPv6 théorie et pratique", Novembre 2005, O'Reilly (accès libre).

<http://www.ipv6forum.org>

<http://www.renater.fr/> , Le Réseau National de télécommunications pour la Technologie l'Enseignement et la Recherche.

<http://www.ripe.net/ipv6> Réseau IP Européen—allocation et attribution des @IPv6

<http://www.ipv6.ru> Russian National IPv6 Forum

## Projets arrêtés, mais beaucoup d'infos utiles

<http://www.iot-inc.com/gogo6/> gogoNET and Freenet6. Etait un portail d'accès gratuit à la dorsale IPv6, > 100.000 membres. Fermé en 2017.

<http://6bone.net> Plate-forme de test historique IPv6 (arrêtée au 6/06/2006, domaine HS).

<http://www.euro6ix.org> Projet de construction de l'IPv6 en Europe ; livrables intéressants (fin 2005).

<http://www.kame.net> KAME, USAGI, TAHI projects. But : fournir les implémentations gratuites de la pile IPv6, IPsec et Mobile IPv6 pour BSD et Linux. (fin en 2006)