

# *IPv6 sous Linux*

---

Roger Yerbanga  
[contact@yerbynet.com](mailto:contact@yerbynet.com)

# Pourquoi IPv6 (1)

- IP : coeur d'Internet, vecteur de communication, numéro des ressources
- IPv4 : organisé en classes au début (A, B, C,...)
  - Problème des classes : trop petites, trop grandes
  - Et 1/8 réservé, classe D multicast
- Politique d'allocation inefficace : pénurie classe B en 1995
- Table de routage Internet en croissance exponentielle : à cause des classes C

# Pourquoi IPv6 (2)

- Invention du classless dans les années 90
  - CIDR
  - 196.0.0.0/8, 196.200.0.0/16, 196.200.64.0/20
- NAT (NAT-P, NAT-4^n) : le sauveur
- Meilleure gestion des adresses :
  - IANA → RIRs → LIRs → Utilisateurs finaux
- => Statistique de pénurie revue à la baisse
- Mais en fév 2011, pool IP IANA épuisé.

# Pourquoi IPv6 (3)

- Quand APNIC ==> PANIC, RIPE ==> PRIE
- AFRINIC ? => NI-AFRIC ? NIC-AFRI ?
- Arrivée massive de smartphones
- Routage devient inefficace
- Problème de gestion de la qualité
- Limite des options de l'entête d'IPv4
- Sécurité d'IPv4

# IPv6 - Arrivée

- Travaux commencé au début 90 pour améliorer IP
- Milieu 90, IPv6 a été retenu comme remplaçant de IPv4.
- Adoption vers fin des années 90
- Nouveau protocole s'appelle IPv6 et non IPv5
- Corrige le problème d'espace d'adressage  $2^{128}$
- On dit qu'IPv6 est censé résoudre les problèmes actuels et futurs de l'Internet

# Caractéristiques

- Simplification du format d'entête (40 Octets)
- Auto-configuration intégrée (NDP)
- $2^{128} = 3,4 \cdot 10^{38}$  adresses possibles =  $7 \cdot 10^{23}$  IP par  $m^2$  de la surface planétaire.
- Amélioration du multicast (scope)
- Amélioration de la gestion des extensions (Next Header)
- Extension des fonctionnalités de sécurité
- Pas de NAT

# Notation (1)

- 128 bits : 8 groupes de 4 hexadécimaux
  - =>XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX
  - 2001:470:c876:ba0:214:78ff:fe7f:90b3/64
  - 2001:0000:2ABC:0000:0000:0000:0000:0001
  - 2001:0000:2ABC::0001
  - 2001:0:2ABC::1
  - 2001:0DB8:0:CD30::/60
  - X:X:X:X:X:X.192.168.0.2

# Notation (2)

- 2001:0db8:0000:0000:0000:0000:0000:0000
- 2001:db8:0:0:0:0:0:0
- 2001:db8::
- 2001:db8::/32 = [2001:0db8:0:0:0:0:0:0 – 2001:0db8:ffff:ffff:ffff:ffff:ffff:ffff]
- 2001:db8::/61 = [2001:db8:0:8:: - 2001:0db8:0000:000f:ffff:ffff:ffff:ffff]



# Types d'adresses

- `::/128` : réservé pour non-spécifié (0.0.0.0)
- `::1/128` : loopback (127.0.0.1/32)
- `2000::/3` : adresses IPv6 routables utilisables pour les unicast (global unicast)
- `FC00::/7` : ULA (privées)
- `FE80::/10` : lien local
- 7/8 du total non assigné en ce moment
- `FF00::/8` (1111 1111) Multicast

# IPv6

Questions ?

# Attribution d'IP statique (1)

- Sur Debian et la plupart des OS, v6 activé par défaut
- Adresse link local active, utilisable localement
  - Calculer automatiquement
  - Algorithme de calcul basé sur la MAC (EUI-64)
  - Du genre FE80::x:x:x:x
  - Petit exo de 5mn pour trouver l'algo ???
    - FF:FE + inversion du bit u
- A tester avec ifconfig, mtr

# Attribution d'IP statique (2)

- Manuelle avec la commande ifconfig :
  - `ifconfig eth0 add FC00:200::1/64`
  - `ifconfig eth0 del FC00:200::1/64`
- Manuelle avec la commande ip :
  - `ip addr add FC00:200::1/64 dev eth0`
  - `ip addr del FC00:200::1/64 dev eth0`
- `ping6 FC00:200::1`

# Attribution d'IP statique (3)

- Ajout de route : `ip route add 2000::/3 via FC00:200::2 dev eth0`
- Suppression de route : `ip route del 2000::/3 via FC00:200::2`
- Affichage : `ip -6 route`
- Testes : `ping6`, `mtr`, `traceroute6`
- Filtrage de port : `ip6tables`
  - `ip6tables -I INPUT -s FC00::/7 -j DROP`
  - `ip6tables -D INPUT -s FC00::/7 -j DROP`

# Sauvegarde de config d'IP

- Fichier /etc/network/interfaces (comme v4) :

- 

```
iface eth0 inet6 static
```

```
address FC00:200::1
```

```
netmask 64
```

```
gateway FC00:200::2
```

- Lire aussi /etc/sysctl.conf
- On peut avoir besoin de bloquer l'autoconfiguration pour du manuel uniquement
- Faire quelques tests

# IPv6

Questions ?

# Transition v4-v6

- Dual-stack : IPv6 et IPv4 ensemble sur le même noeud
- Tunnel : IPv6 et IPv4 ne sont pas compatibles
- Translation : permettre à des IPv6 purs de communiquer avec des IPv4 purs
- En général, les 3 techniques sont utilisées



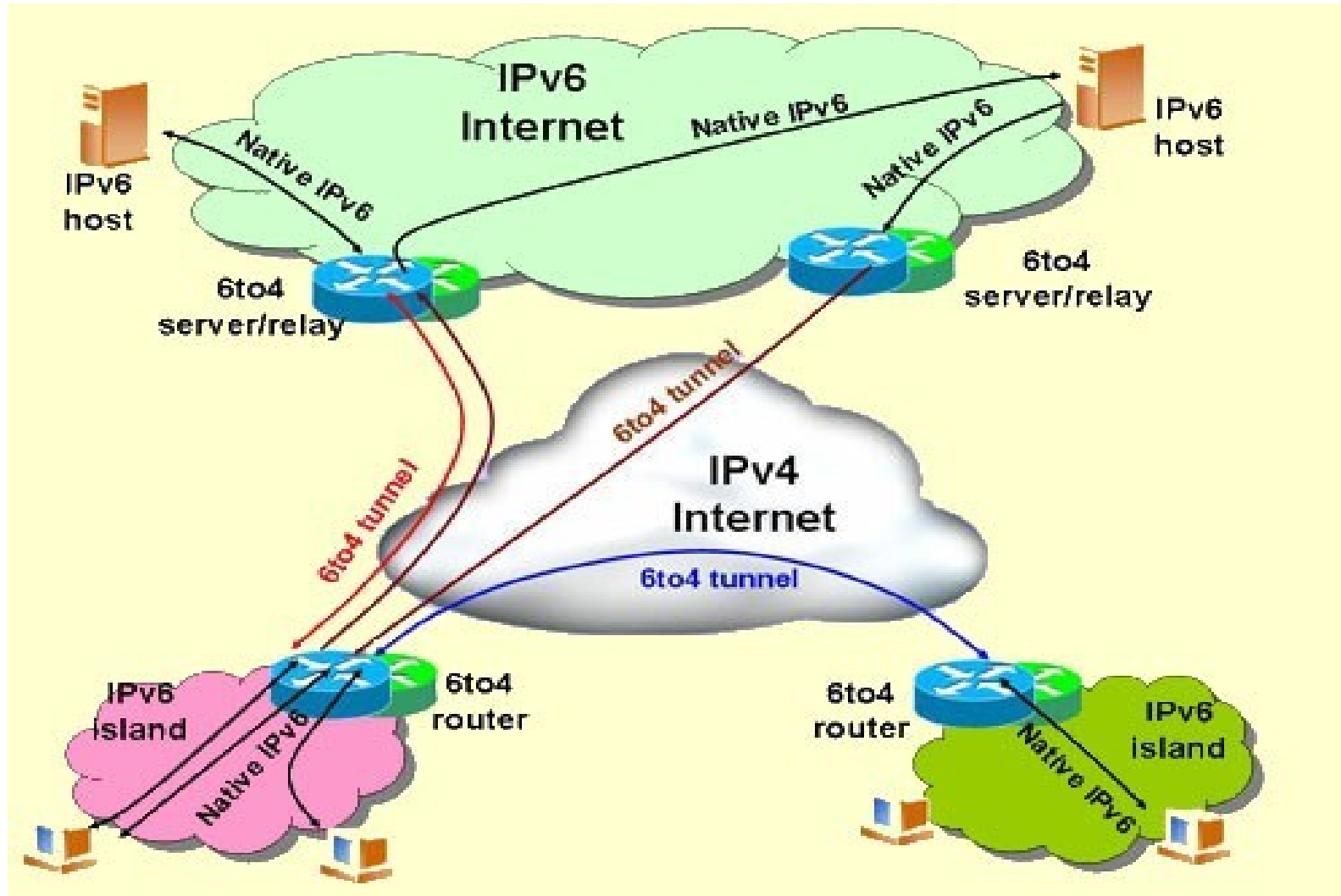
# Dual-stack

- IPv4 et IPv6 sont activés
- L'interface sait communiquer avec ses 2 IPs
- Communique aux hôtes IPv4 à travers son IPv4
- Communique aux IPv6 avec son v6
- Des problèmes de préférences :
  - Soit réglées par le DNS
  - Soit par l'application
- Possibilité de parler d'abord v6, puis v4, ou l'inverse.

# Tunnels

- Permettre à un îlot v6 de communiquer avec un autre îlot v6 en passant par le reste du monde.
- Tunnels manuels
- Semi-automatiques
  - Tunnel Broker
- Automatiques :
  - 6to4
  - ISATAP
  - 6rd

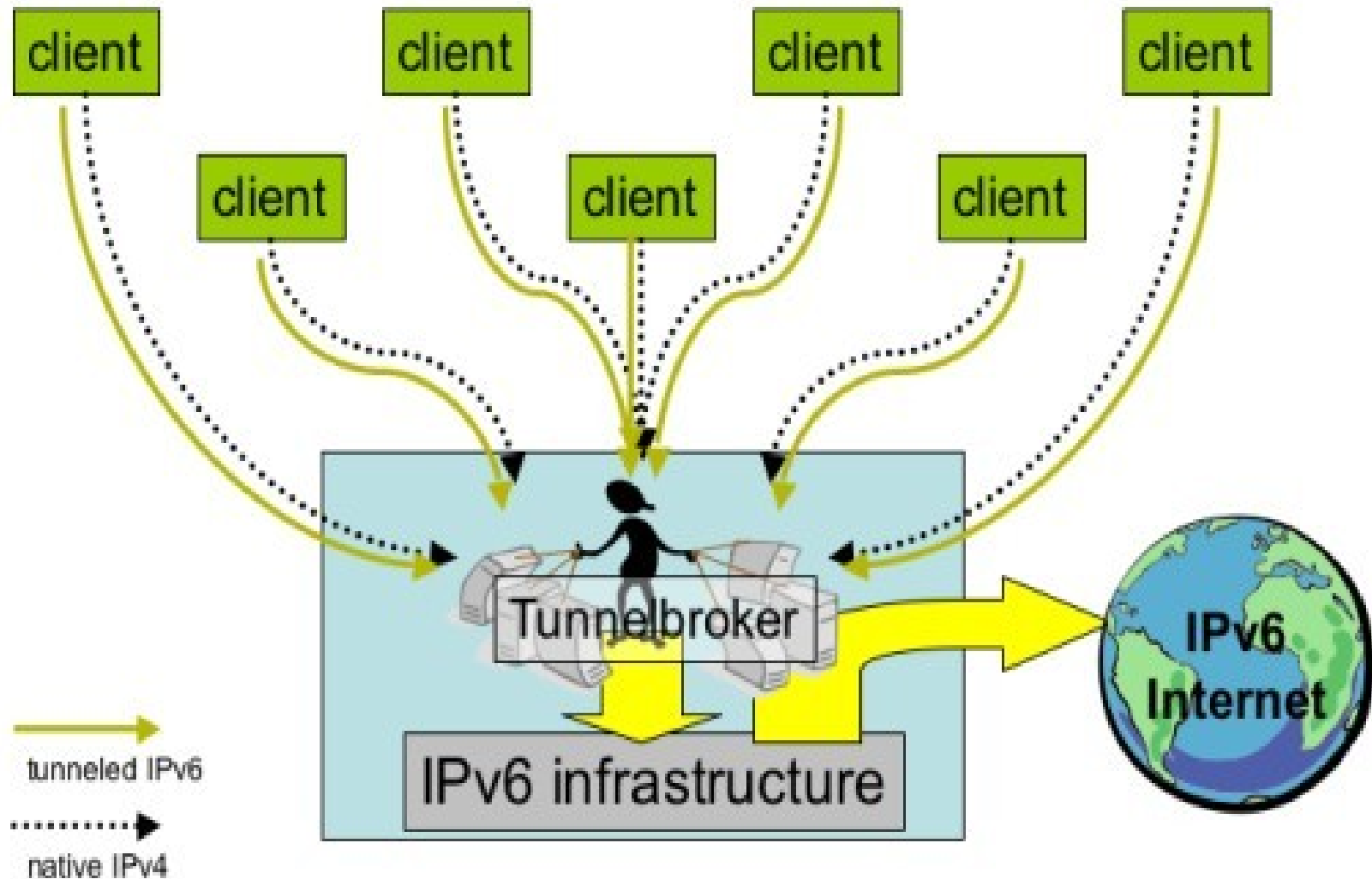
# 6to4



# Tunnel Broker (1)

- Basé sur des serveurs dédiés « Tunnel Brokers »
- Gèrent automatiquement les demandes de tunnel des clients
- Bien adapté pour les petits sites IPv6 isolés
- Gratuits :
  - HE : [www.he.com](http://www.he.com)
  - Go6 : [www.go6.net](http://www.go6.net)
- Config client souvent automatique

# Tunnel Broker (2)



# Tunnel chez HE (1)

- <http://tunnelbroker.net/>
- S'enregistrer et s'authentifier
- Et prendre un tunnel :
  - Choisir le point recommandé (ou proche)
  - Donner son IPv4
- Lire un exemple de configuration
- Puis faire la même chose sur sa bécane (Copier/coller marche bien en général)

# Tunnel chez HE (2)

- Possible d'avoir un /48 et de distribuer des IPs sur vos réseaux
- Possible d'avoir du reverse-DNS
- Supprimer des tunnels qui ne sont plus utilisés
- Quelques outils de test : looking glass

# Config Tunnel

*auto tun6to4*

*iface tun6to4 inet6 v4tunnel*

*endpoint 216.66.84.42 # l'IPv4 chez HE*

*local 196.200.55.1 # l'IP routable locale*

*ttl 255 # ttl optionnel*

*address 2001:478:88f1:6d9::2 # IPv6 assignée*

*netmask 64 # Longueur du préfixe obtenu*

*mtu 1480*

*# puis la route v6 par défaut*

*up /sbin/ip -6 route add 2000::/3 dev tun6to4 metric 1||true*

*down /sbin/ip -6 route del 2000::/3 dev tun6to4 metric 1||true*



# IPv6

Questions ?

# DNS et IPv6

- Nouveau record de type **AAAA**
- `dig AAAA ns.refer.sn`
  - `ns.refer.sn. 86400 IN AAAA  
2001:470:c876:d10::66`
  - Extrait de `/etc/bind/local/db.refer.sn` :

```
ns      A      213.154.65.66
        MX   10 smtp
        AAAA 2001:470:c876:d10::66
```

# Reverse-DNS et IPv6

- Domaine : **ip6.arpa** au lieu de in-addr.arpa  
*zone "6.7.8.c.0.7.4.0.1.0.0.2.ip6.arpa." {  
type master;  
file "/etc/bind/db.6.7.8.c.0.7.4.0.1.0.0.2.ip6.arpa." ;  
};*
- Dans le fichier de zone :  
*8.7.0.1.d.0 IN PTR mail-  
dakar.refer.sn.  
4.7.0.1.d.0 IN PTR  
bobo.upb.bf.*

# Adressage automatique des PC

- Il faut un ou plusieurs RA (Router Advertisement) sur le réseau
- Les RA envoient les préfixes
- Et les clients s'autoconfigurent
  - Préfixe récupéré + EUI-64
  - Préfixe récupéré + autre méthode pour avoir 64bits
  -

# RADVD (1)

- Router Advertisement Daemon
- Permet à une machine linux d'être un router RA, et d'annoncer des préfixes
- Préfixes annoncés dépendants des interfaces
  - Peut annoncer un préfixe sur une interface
  - Et un autre préfixe à travers une 2è interface
  - Ainsi de suite ...
- ***aptitude install radvd***

# RADVD (2)

- Exemple simple de configuration (/etc/radvd.conf) :

```
interface eth0
{
    AdvSendAdvert on;
    prefix 2001:470:1F01:1908::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
};
```

- man radvd.conf : très détaillé

# RADVD (3)

- Quelques options de l'interface :
  - **IgnoreIfMissing** on|off : ignorer l'existence ou non de l'interface
  - **AdvSendAdvert** on|off : envoi ou non de ra et réponses
  - **MaxRtrAdvInterval** secs : temps max entre les envois de ra (doit être entre 4 et 1800 secondes)
  - **MinRtrAdvInterval** secs : temps min
  - **AdvLinkMTU** *n* : la MTU à utiliser par les clients
  - **AdvDefaultLifetime** secs : durée de vie

# RADVD (4)

- Quelques options du préfixe :
  - **AdvOnLink** on|off : il faut le mettre à on. Permet de dire s'il est on-link.
  - **AdvValidLifetime** secs|infinity : validité du préfixe en seconde
  - **AdvPreferredLifetime** secs|infinity : délai de préférence du préfixe.
  - **AdvAutonomous** on|off : Préfixe peut être utilisé pour la config d'adresse autonome.



# RADVD (5)

- Autre exemple de config :

```
interface eth0
{
    AdvSendAdvert on ;
    AdvLinkMTU 1480;
    prefix 2001:db8:0:1::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
    clients
    {
        fe80::21f:16ff:fe06:3aab;
        fe80::21d:72ff:fe96:aaff;
    };
};
```

# IPv6

Questions ?

# Filtrage IPv6

- Une seule commande : **ip6tables**
- ip6tables -F
- ip6tables -P FORWARD DROP
- ip6tables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
- ip6tables -A FORWARD -o tun6to4 -i eth1 -j ACCEPT
- ip6tables -A FORWARD -o \$IFDMZ -p udp -d \$NSV6 --dport 53 -j ACCEPT
- ip6tables -A FORWARD -i tun6to4 -o \$IFDMZ -p tcp --dport 25 -d \$MAILV6 -j ACCEPT

# Filtrage IPv6

- `ip6tables -A FORWARD -o eth1 -i eth2 -j ACCEPT`
- `ip6tables -A FORWARD -o eth2 -i eth1 -j ACCEPT`
- `ip6tables -A FORWARD -i tun6to4 -o $IFDMZ -p icmpv6 -j ACCEPT`
  
- Et le NAT ?
  - **on le range aux oubliettes**

# IPv6

Questions ?