

# Séminaire réseaux 3

## IPv6 : Fonctionnement et enjeux pour l'internet de demain

Gabriel Détraz, le 25 novembre 2014



## I - Déploiement sur le réseau internet

- Avantages de l'IPv6
- Déploiement assez lent malgré l'urgence

## II - Principes généraux de l'IPv6

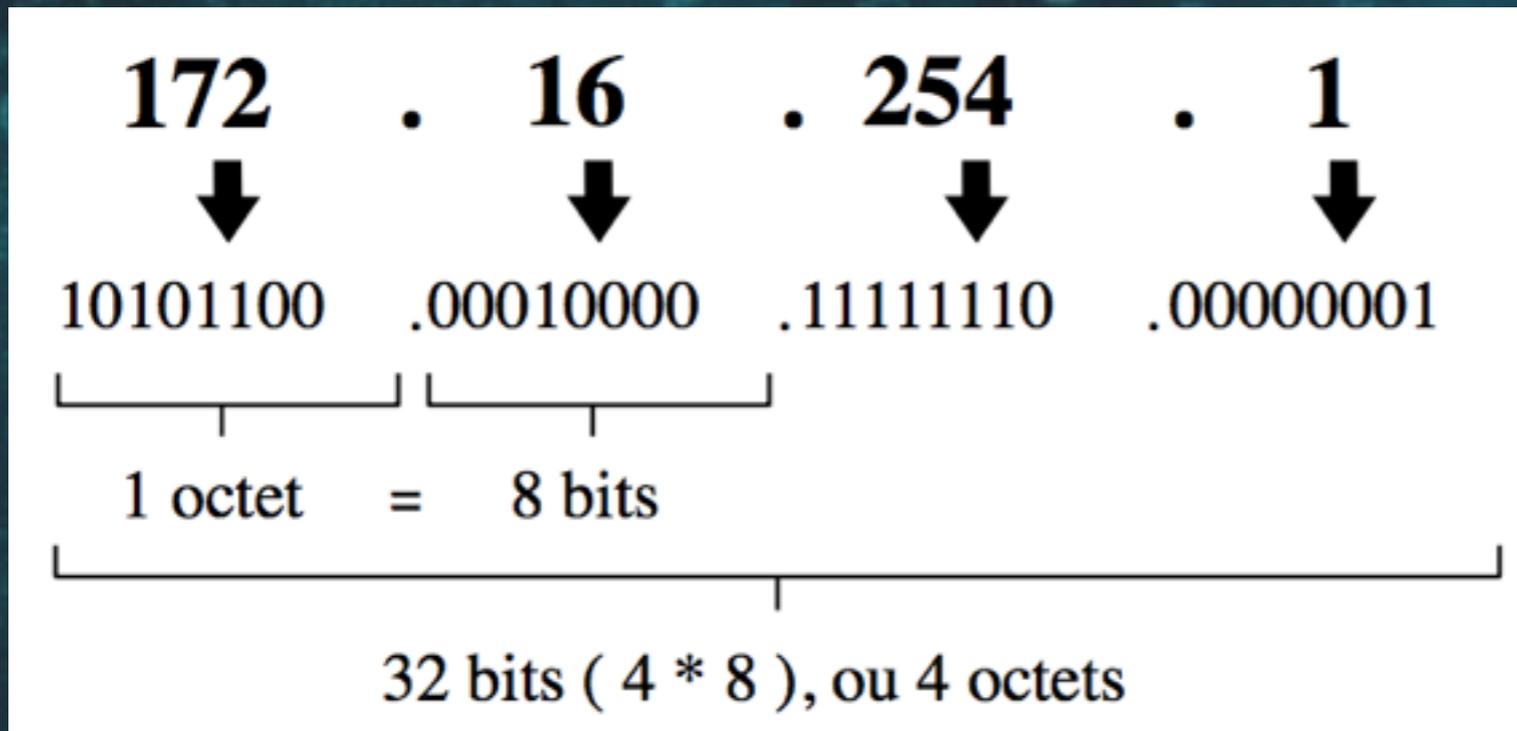
- Différences/Similarités avec l'IPv4
- Fonctionnement de l'adressage IPv6

## III - Implémentation actuelle et envisagée au Crans

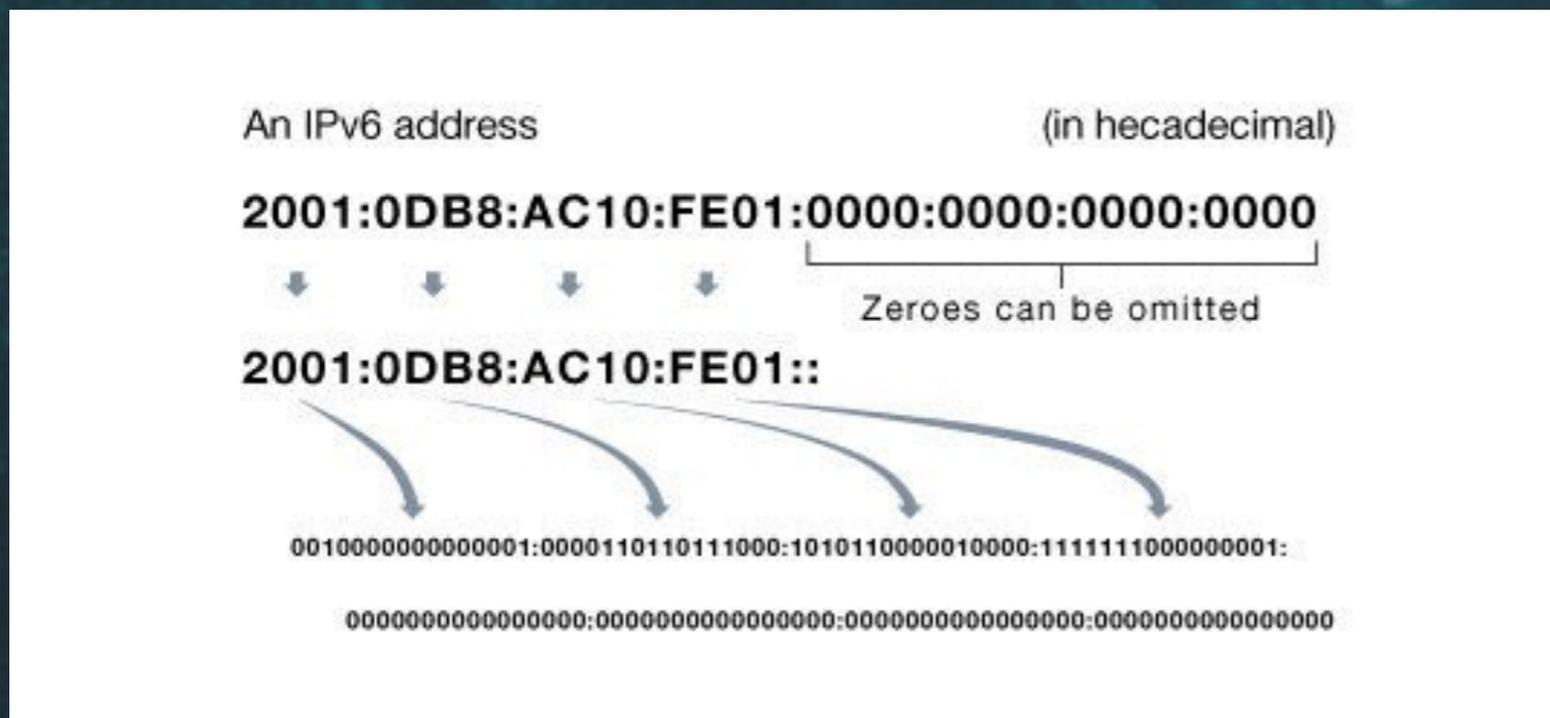
- Tunnel IPv6 actuel
- IPv6 only avec le NAT64

Conclusion/Questions

# I - Déployer l'IPv6 à grande échelle sur l'internet mondial



IPv4 :  $2^{32}$  adresses possibles, soit environ 4 milliards ( $4,29 \times 10^9$ )



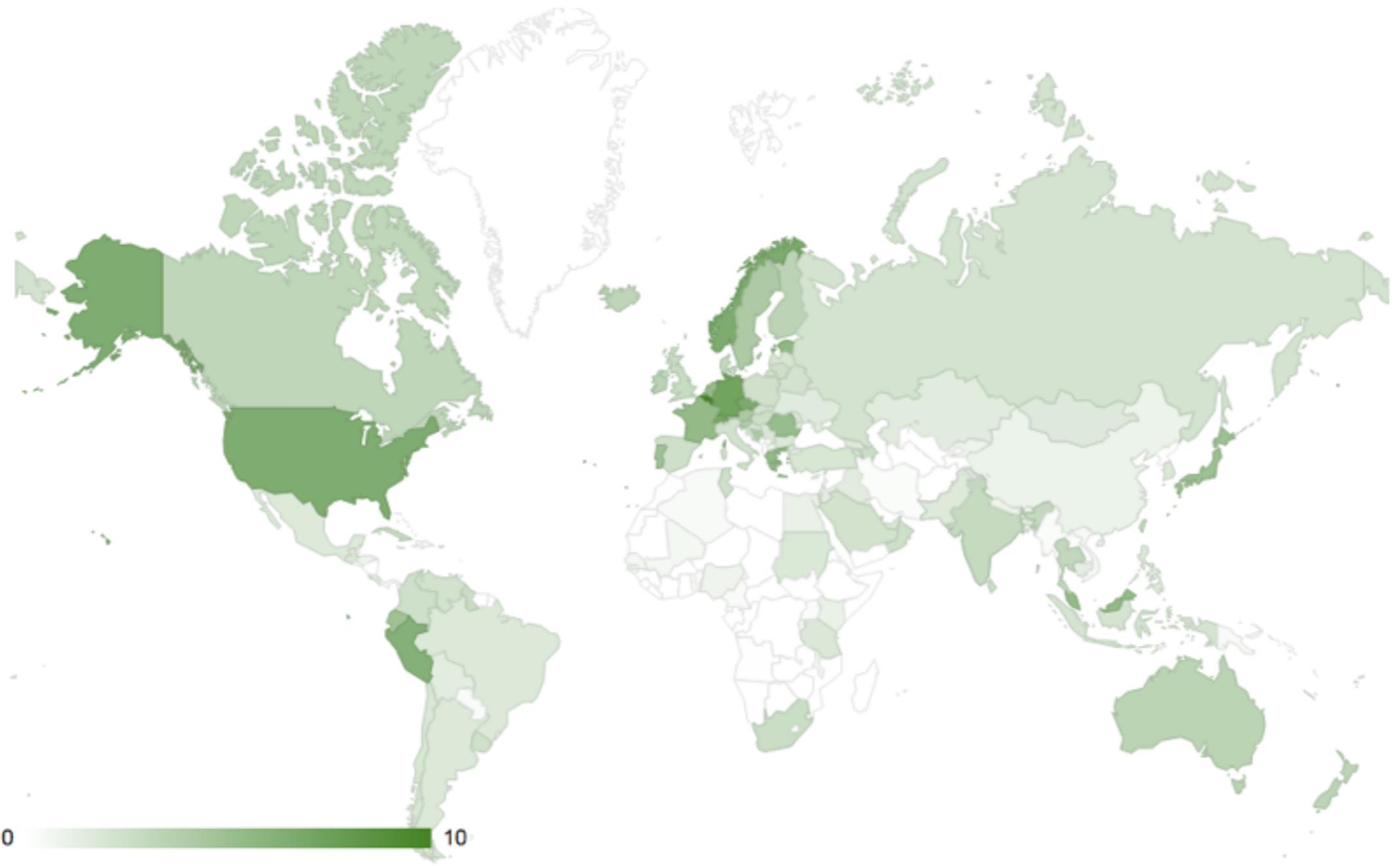
IPv6 : 16 octets (128 bits) en hexadécimal, soit  $2^{128}$  adresses. ( $3,40 \times 10^{38}$ )

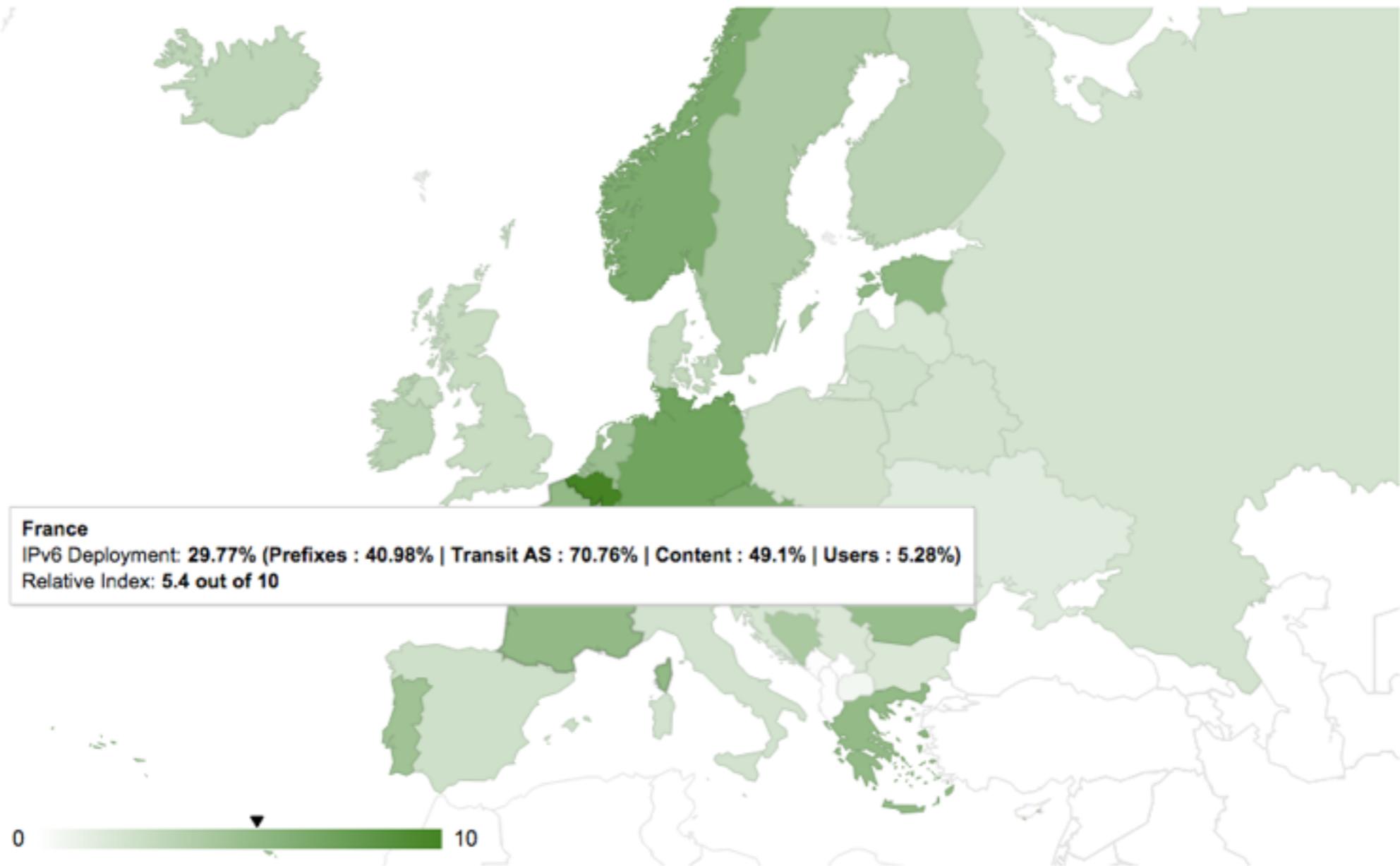
## Epuisement et mauvaise gestion des IPv4 :

- En 2011, le dernier slash disponible est attribué
- Granularité héritée des années 80 (très difficile à corriger)

# Actuellement, déploiement partiel:

Opérateur	Date de déploiement	Taille du préfixe
Renater	1996, IPv6 natif en 2002	?
Free	2007	/61
OVH	mi 2012	/56
SFR	2013	/64
Bouygues	Repoussé aux calendes grecques	
Orange	2015 (peut être)	/56





<http://6lab.cisco.com/stats/>

# D'ou l'urgence de passer à l'IPv6

## Avantages supplémentaires :

- Plus de NAT classique (NAT44) : les machines derrière une box compatible IPv6 peuvent être contactées directement, car ayant une IP publique
- Une machine peut posséder plusieurs IP publiques et privées, ce qui n'était pas le cas en IPv4

# A la découverte du réseau local et au delà : quelques commandes utiles pour l'IPv6

```
detr@zamok ~ % ping6 vo.crans.org
PING vo.crans.org(vo.crans.org) 56 data bytes
16 bytes from 2a01:240:fe3d:4:224:8cff:fe44:3b70, icmp_seq=0 hlim=63 time=24.602 ms
16 bytes from 2a01:240:fe3d:4:224:8cff:fe44:3b70, icmp_seq=1 hlim=63 time=6.313 ms
16 bytes from 2a01:240:fe3d:4:224:8cff:fe44:3b70, icmp_seq=2 hlim=63 time=1.860 ms
```

```
detr@zamok ~ % traceroute6 google.fr
traceroute vers google.fr (2a00:1450:4007:805::1018) de 2a01:240:fe3d:4:219:bbff:fe3c:4f76,
port 33434, du port 36106, 30 sauts max, 60 octets/paquet
 1 2a01:240:fe3d:4::1 (2a01:240:fe3d:4::1) 0.190 ms 0.192 ms 0.206 ms
 2 2a01:240:fe00:68::1 (2a01:240:fe00:68::1) 12.486 ms 12.155 ms 12.546 ms
 3 frmrs01.sixxs.net (2a01:240:202::2) 12.219 ms 12.670 ms 12.028 ms
 4 sixxs-gw.cust.ipv6.jaguar-network.net (2a01:240:202::1) 19.329 ms 19.680 ms 19.234 ms
```

## SSH en IPv6 :

```
detr@zamok ~ % ssh -6 gdetraz@tahines.ens-cachan.fr  
ssh: Could not resolve hostname tahines.ens-cachan.fr: Name or service not known
```

## Ajout manuel d'une IPv6 :

```
detr@zamok ~ % ip -6 add 3ffe:ffff:0:f101::1/64 dev eth0
```

# Afficher les adresses IPv6 :

```
detrax@zamok ~ % ip -6 addr show
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436
```

```
inet6 ::1/128 scope host
```

```
valid_lft forever preferred_lft forever
```

```
3: crans: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1496 qlen 1000
```

```
inet6 2a01:240:fe3d:4:219:bbff:fe3c:4f76/64 scope global
```

```
valid_lft forever preferred_lft forever
```

```
inet6 fe80::219:bbff:fe3c:4f76/64 scope link
```

```
valid_lft forever preferred_lft forever
```

```
4: crans.2@crans: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1496
```

```
inet6 2a01:240:fe3d:c804:219:bbff:fe3c:4f76/64 scope global
```

```
valid_lft forever preferred_lft forever
```

```
inet6 fe80::219:bbff:fe3c:4f76/64 scope link
```

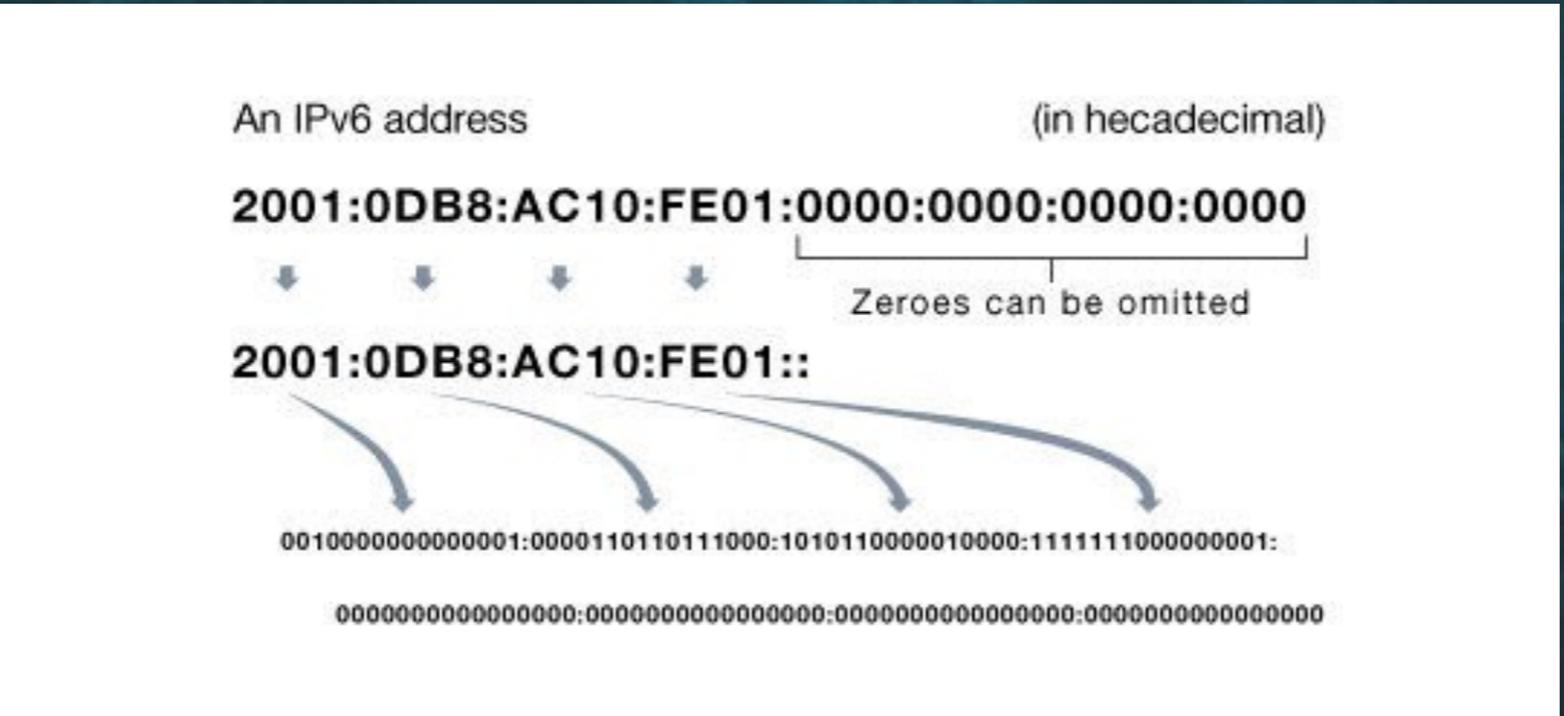
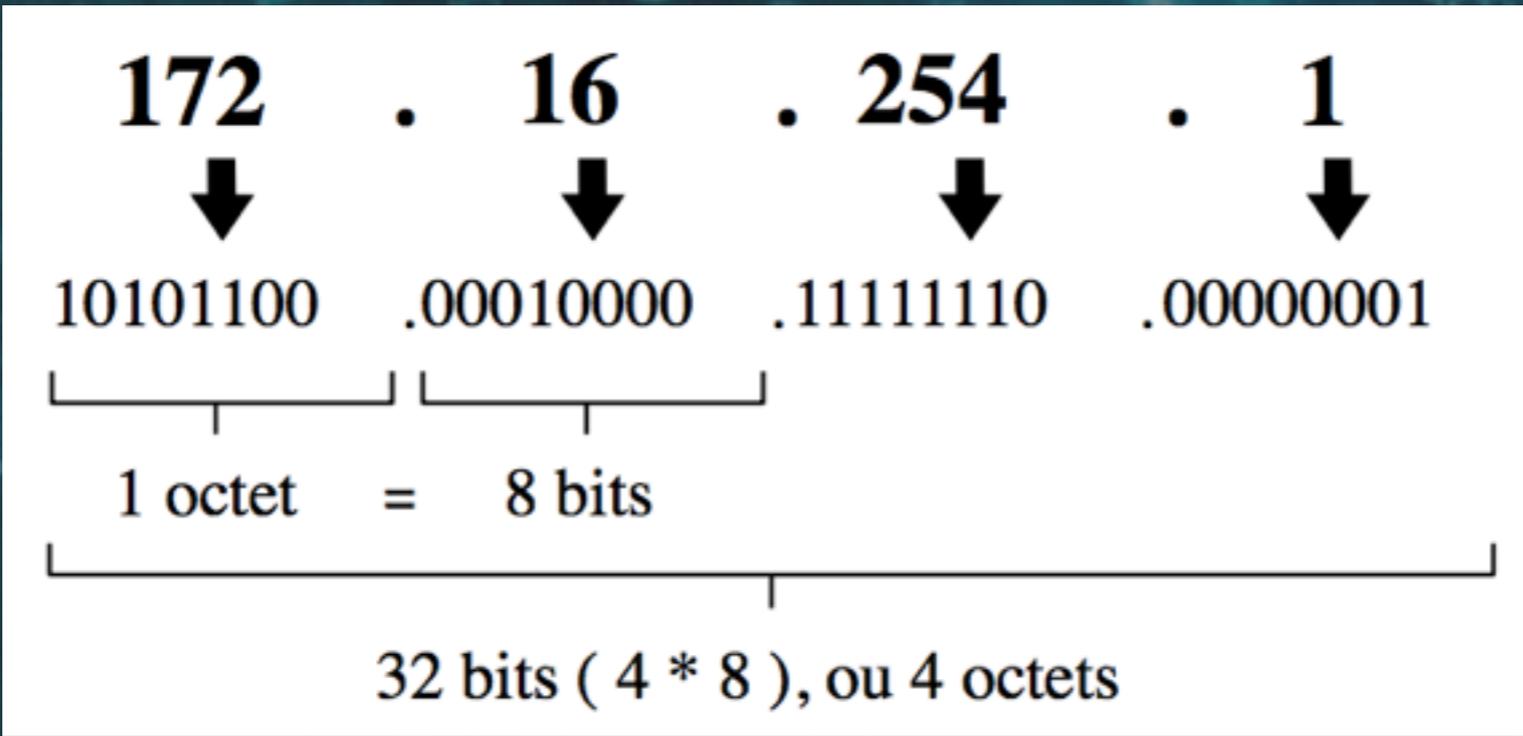
```
valid_lft forever preferred_lft forever
```

## Afficher les routes IPv6 :

```
detrax@zamok ~ % ip -6 route show
2a01:240:fe3d:4::/64 dev crans metric 1024
2a01:240:fe3d:c804::/64 dev crans.2 proto kernel metric 256
fe80::/64 dev crans proto kernel metric 256
fe80::/64 dev crans.2 proto kernel metric 256
default via fe80::1 dev crans metric 1024
```

Etc

# II - Fonctionnement de l'IPv6, routage avec ce protocole



# En pratique

- Ecriture conventionnelle :

Les 0 à gauche (excédentaires) ne sont pas affichés

↓ ↓  
2a01:340:fe3d:c04:20f7:30b0:ef:bfca

Si on a un ou plusieurs octets de 0, on contracte l'adresse

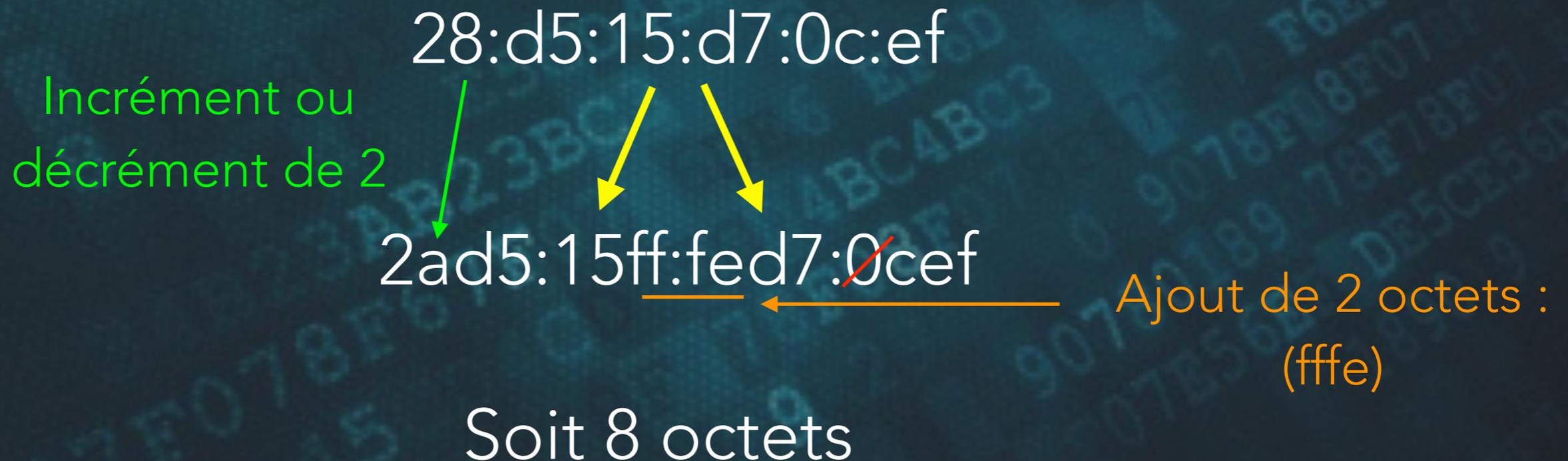
↓  
fda8:5d34:a228:c04::1ce6:4bad

# Avoir une adresse IPv6 routable :

- Autoconfiguration avec le protocole NDP

On utilise d'adresse mac de l'appareil (unique), on crée une adresse IPv6 avec.

C'est le protocole SLAAC (Stateless Address Autoconfiguration)



Le préfixe est annoncé par le routeur grâce à NDP. (Router Advertisement)

Il est composé de 8 autres octets :

2a01:240:fe3d:c04

On complète la partie précédente et on obtient une adresse IPv6 :

2a01:240:fe3d:c04:2ad5:15ff:fed7:cef

Cette adresse est routable, unique et publique.

# Le protocole NDP (Neighbor Discovery Protocol)

Il remplace ARP et en partie le DHCP en IPv6. Il s'occupe à la fois de la découverte du réseau local et de l'attribution des IP. On peut le décomposer comme ceci :

- \* Router Solicitation : demande de Router Advertisement
- \* Router Advertisement : réponse d'un routeur, qui annonce les préfixes publics
- \* Neighbor Solicitation : équivalent à l'ARP en IPv4, mais permet aussi de vérifier que personne n'a la même IP que soi.
- \* Neighbor Advertisement : réponse à une Neighbor Solicitation

142	6.353248000	138.231	138.231	NBNS	92 Name query NB WPAD<00>
143	6.355171000	138.231.	239.255.	SSDP	175 M-SEARCH * HTTP/1.1
144	6.356197000	Apple_	Broadcast	ARP	56 Gratuitous ARP for (Request)
145	6.357110000	Apple_32:5d:4e	Broadcast	XID	56 Basic Format; Type 1 LLC (Class I LLC); Window Size 0
146	6.358326000	HewlettP_	Broadcast	ARP	56 Who has 138.231. ?
147	6.555385000	HewlettP	Broadcast	ARP	56 Who has 138.231.
148	6.556810000	fe80::	ff02::16	ICMPv6	90 Multicast Listener Report Message v2
149	6.558084000	fe80::	ff02::16	ICMPv6	90 Multicast Listener Report Message v2
150	6.559358000	fe80::4434:	ff02::1:3	LLMNR	84 Standard query 0xdfdb ANY Alex
151	6.560405000	138.231.148	224.0.0.252	LLMNR	64 Standard query 0xdfdb ANY Alex
152	6.562236000	138.231.145.107	239.255.255.250	SSDP	167 M-SEARCH * HTTP/1.1
153	6.563267000	HewlettP_00:f4:04	Broadcast	ARP	56 Who has ;? Tell
154	6.565229000	138.	255.255.255.255	UDP	170 Source port: 56855 Destination port: 10007
155	6.568449000			NBNS	92 Name query NB ISATAP<00>
156	6.760392000	LiteonTe_	Broadcast	ARP	56 Who has ? Tell 0.0.0.0
157	6.761700000	::	ff02::1:	ICMPv6	78 Neighbor Solicitation for fe80
158	6.762965000	::	ff02::1:	ICMPv6	78 Neighbor Solicitation for 2a01
159	6.764067000	::	ff02::1:	ICMPv6	78 Neighbor Solicitation for 2a01
160	6.765205000	::	ff02::1:	ICMPv6	78 Neighbor Solicitation for fda8
161	6.766608000	::	ff02::1:	ICMPv6	78 Neighbor Solicitation for fda8
162	6.767669000	fe80::4434:	ff02::2	ICMPv6	62 Router Solicitation
163	6.769824000	fe80::a2d3:	ff02::1	ICMPv6	198 Router Advertisement from a0:d3
164	6.773807000	fe80::4434:	ff02::16	ICMPv6	170 Multicast Listener Report Message v2
165	6.775821000	138.231.148.107	239.255.255.250	SSDP	175 M-SEARCH * HTTP/1.1

## Autres solution :

- Utiliser un DHCPv6
- Effectuer un tirage pseudo-aléatoire

# Avoir une adresse IPv6 non routable :

Adresse de lien local :

On utilise un préfixe en fe80::/10 et on complète avec le SLAAC  
(un préfixe est composé de 8 octets) :

**fe80::2ad5:15ff:fed7:cef**

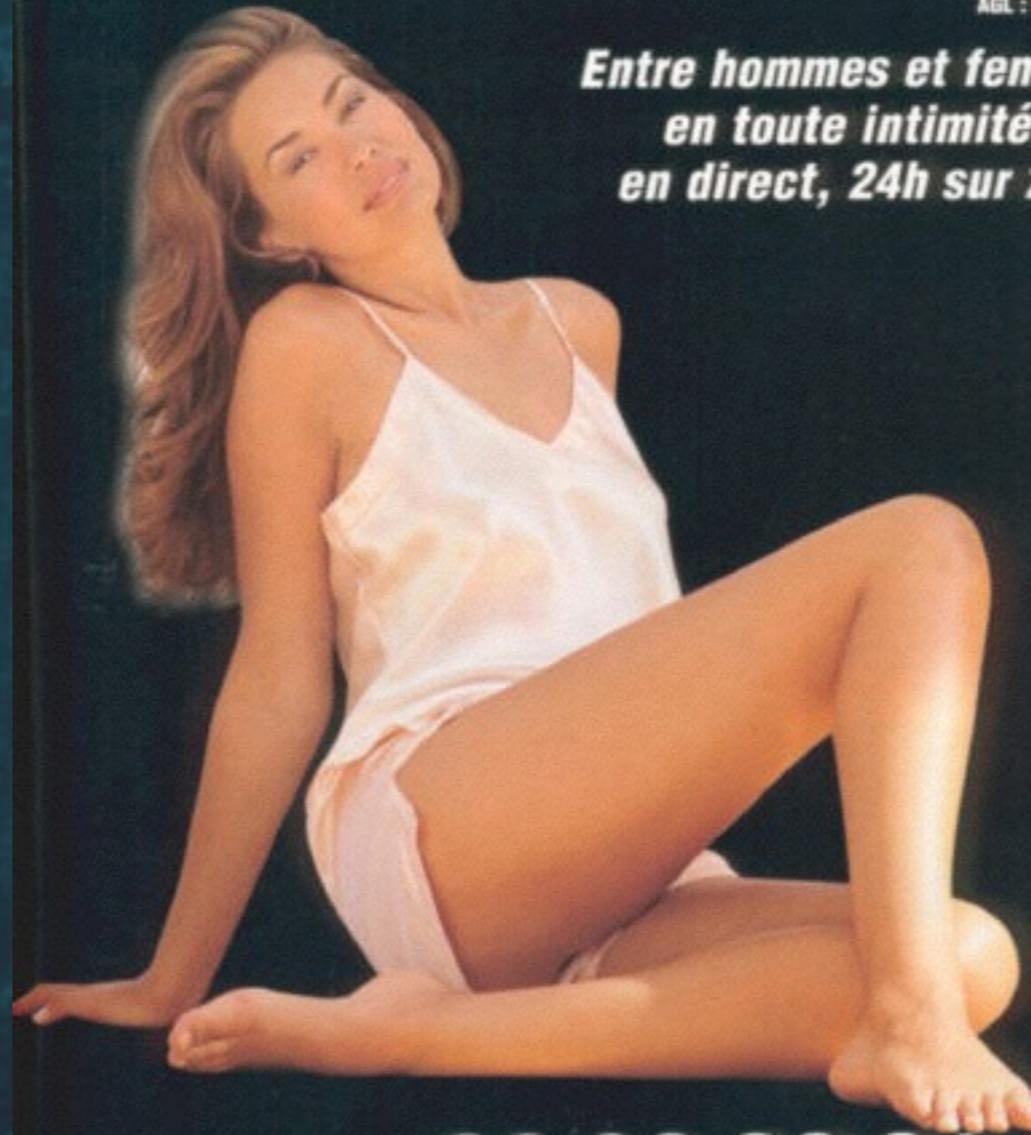
Soit fe80:0000:0000:0000 au lieu de 2a01:240:fe3d:c04

- Adresse IPv6 ULLA :

**36 15 ULLA**

AGL : 1,29 F/mn

*Entre hommes et femmes  
en toute intimité,  
en direct, 24h sur 24.*



Par téléphone : **08.36.68.21.11**

AGL : 2,23 F/mn

Par internet : [www.ulla.com](http://www.ulla.com)

## Non, ULA ! (Unique Local Address) :

Elles utilisent un préfixe `fc00::/7`, le suffixe (64 derniers bits) est généré de la même manière que pour une IP routable. Il doit être généré de façon aléatoire.

Avantage : cette adresse, locale, n'est pas tributaire du matériel : si on change un serveur, on n'a pas à changer l'IP dans tous les appareils situés sur le même réseau local (il suffit de redonner l'ULA au nouveau serveur)

Inconvénient : ceci est totalement dépendant du caractère aléatoire lors de la génération de l'IP

# Préfixes en IPv6 :

fe80::/10 : adresses de lien local non routables

/10 : de fe80:0:0:0:0:0:0:0 à febf:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Explication :

/10 : les 10 premiers bits (1 octet + 2 bits) sont fixes : fe = 1 octet, et 8 s'écrit 1000

Donc on va de 1000 à 1011, soit de 8 à 12, et 12 = b en hexadécimal

2000::/3 : adresses routables sur internet

2000::/3 : de 2000:0:0:0:0:0:0:0 à ?

fc00::/7 : adresses locales uniques (ULA)

de fc00:0:0:0:0:0:0:0 à fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

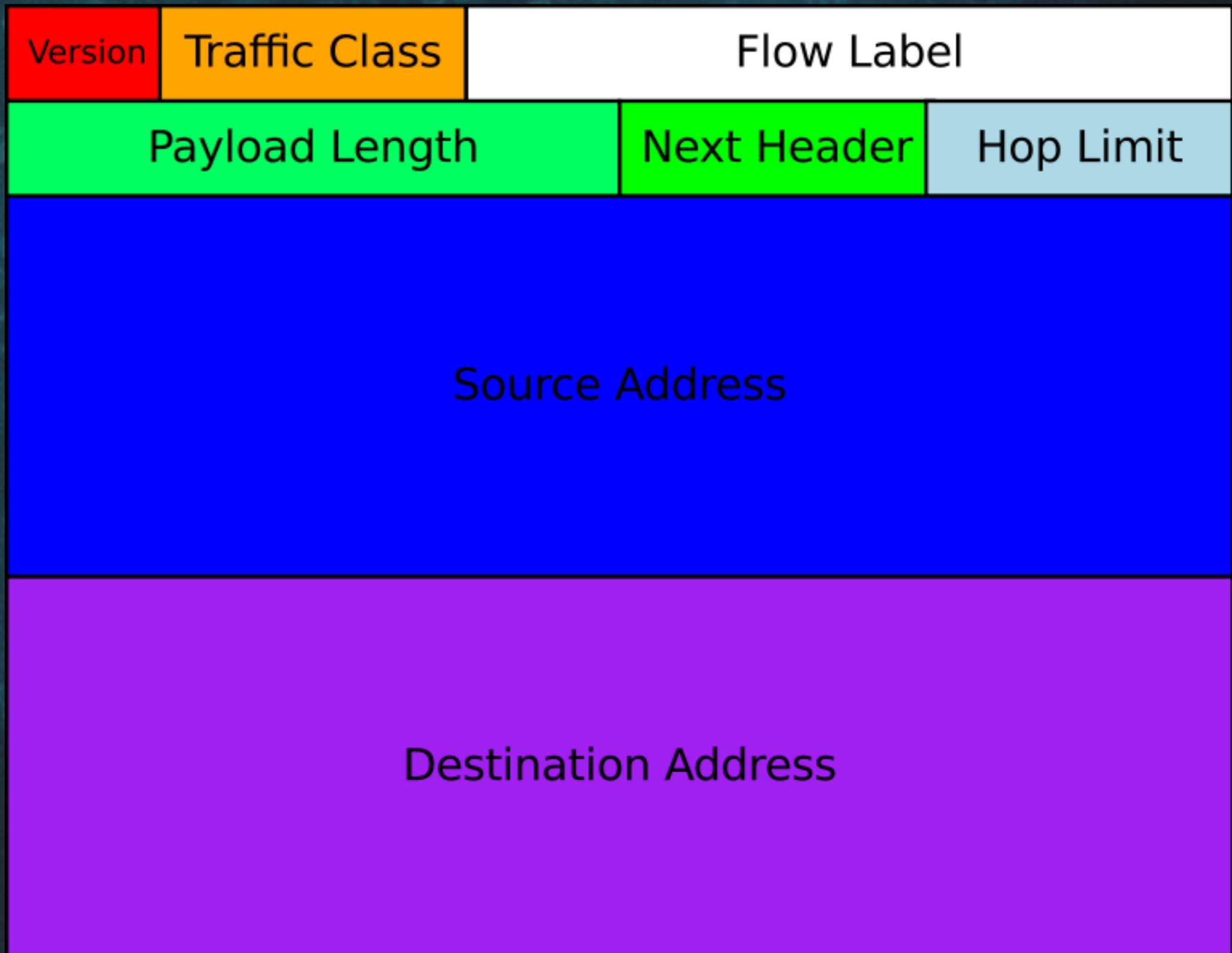
ff00:/8 : adresses multicast

de ff00:0:0:0:0:0:0:0 à ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Il n'y a plus de broadcast

# En têtes IPv4 vs IPv6 :

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



### III - Le Crans, un cas pratique de la « coexistence pacifique »



# Pourquoi un tunnel?

```
detraz@zamok ~ % mtr 8.8.8.8
```

```
My traceroute [v0.82]
zamok (0.0.0.0)
Resolver: Received error response 2. (server failure)er of fields quit
Sun Nov 23 13:48:05 2014
```

Host	Packets			Pings			
	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1. odlyd.crans.org	0.0%	24	0.2	0.1	0.1	0.2	0.0
2. pioneer2.zrt.ens-cachan.fr	0.0%	24	0.3	0.3	0.3	0.6	0.1
3. 193.49.65.1	0.0%	23	0.9	2.4	0.4	24.6	5.6
4. vl402-gi8-2-cachan-rtr-021.noc.renater.fr	0.0%	23	0.8	2.1	0.6	29.8	6.0
5. te3-3-jussieu-rtr-021.noc.renater.fr	0.0%	23	1.3	1.4	1.0	5.2	0.9
6. te0-1-0-3-paris2-rtr-001.noc.renater.fr	0.0%	23	44.7	14.0	2.0	121.8	29.7
7. te1-1-paris2-rtr-021.noc.renater.fr	0.0%	23	1.1	2.4	1.0	26.5	5.3
8. google-te1-6-paris2-rtr-021.noc.renater.fr	0.0%	23	1.3	2.1	1.2	14.8	2.9
9. 72.14.238.228	0.0%	23	4.6	6.2	1.5	45.9	10.1
10. 209.85.245.72	0.0%	23	2.4	1.9	1.6	2.5	0.2
11. 216.239.51.196	0.0%	23	6.9	6.6	6.4	7.1	0.2
12. 72.14.239.98	0.0%	23	6.5	7.4	6.4	18.6	2.5
13. ???							
14. google-public-dns-a.google.com	0.0%	23	6.8	6.3	6.1	6.9	0.2

Problème : Rubis ne fait pas d'IPv6 !

# On passe donc par un tunnel : on encapsule la trame IPv6 dans une trame IPv4 (jaguar-network)

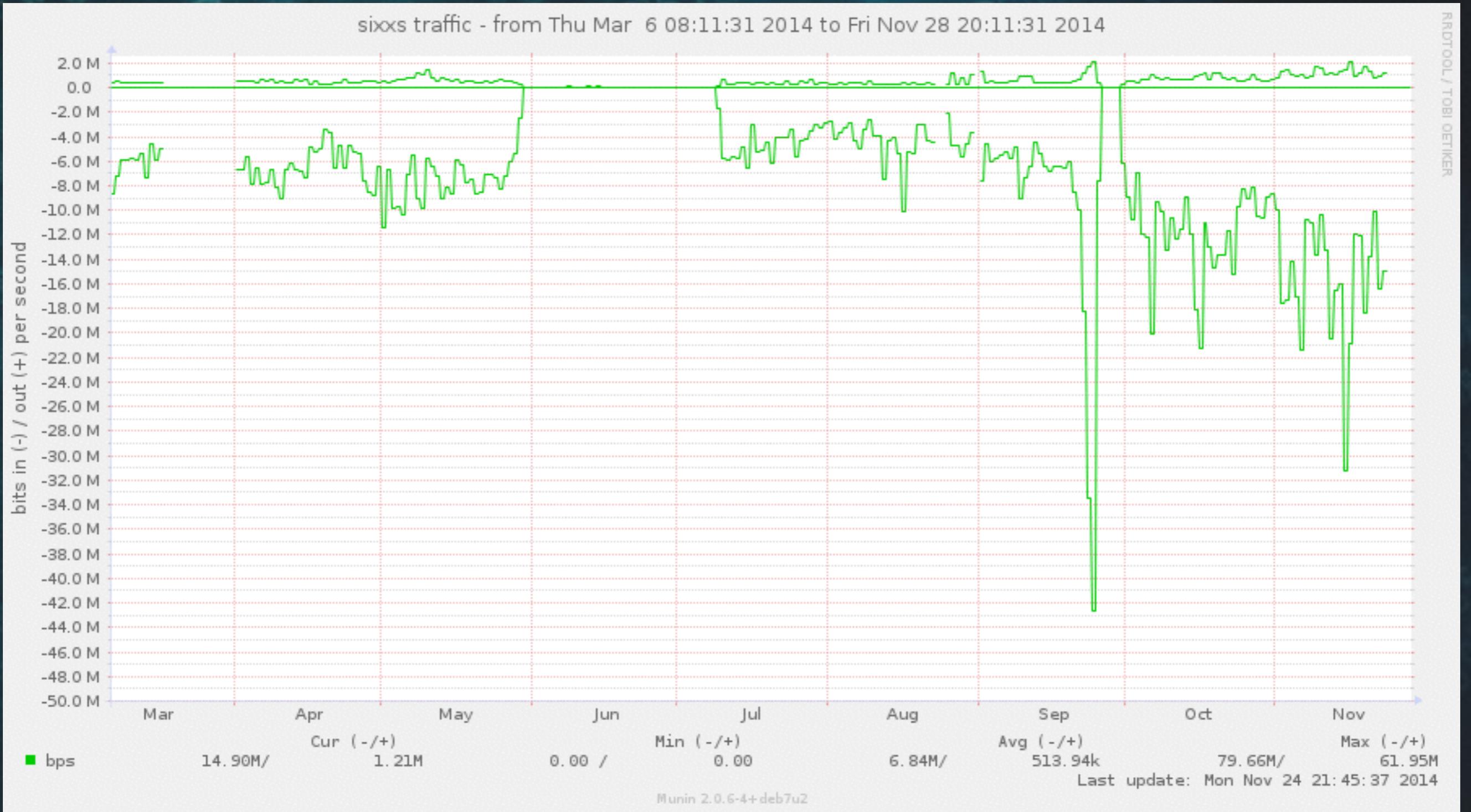
detraz@zamok ~ % mtr -6 google.fr

```
My traceroute [v0.82]
zamok (::)
Resolver: Received error response 2. (server failure)er of fields quit
Sun Nov 23 13:53:14 2014
```

Host	Packets		Pings				
	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1. 2a01:240:fe3d:4::1	0.0%	17	0.2	0.2	0.2	0.9	0.2
2. 2a01:240:fe00:68::1	0.0%	17	12.4	12.9	11.9	14.6	0.6
3. frmrs01.sixxs.net	0.0%	17	14.5	13.5	12.0	18.9	1.7
4. sixxs-gw.cust.ipv6.jaguar-network.net	0.0%	17	14.0	19.0	13.1	24.1	3.1
5. fe3-1.er01.mar01.ipv6.jaguar-network.net	11.8%	17	16.5	15.0	13.0	16.8	1.3
6. google1.franceix.net	0.0%	17	21.7	15.9	12.8	43.7	7.5
7. google1.franceix.net	0.0%	17	14.0	14.8	13.0	27.9	3.5
8. 2001:4860::1:0:7608	0.0%	17	27.8	20.9	13.6	39.3	6.9
9. 2001:4860::8:0:5e19	0.0%	17	24.8	25.5	23.3	30.0	1.6
10. 2001:4860::1:0:9f2	0.0%	17	25.8	28.9	23.8	48.3	7.4
11. 2001:4860:0:1::3a1	6.2%	16	24.4	26.0	24.2	30.5	1.8
12. par03s12-in-x0f.1e100.net	0.0%	16	24.0	25.0	23.6	28.2	1.2



# Trafic dans le tunnel :



# Fonctionnement actuel du routage au Crans :

Une machine d'adhérent dispose d'une IPv4 et éventuellement d'une IPv6 selon sa configuration.

L'adhérent veut contacter un site ayant une adresse IPv4?

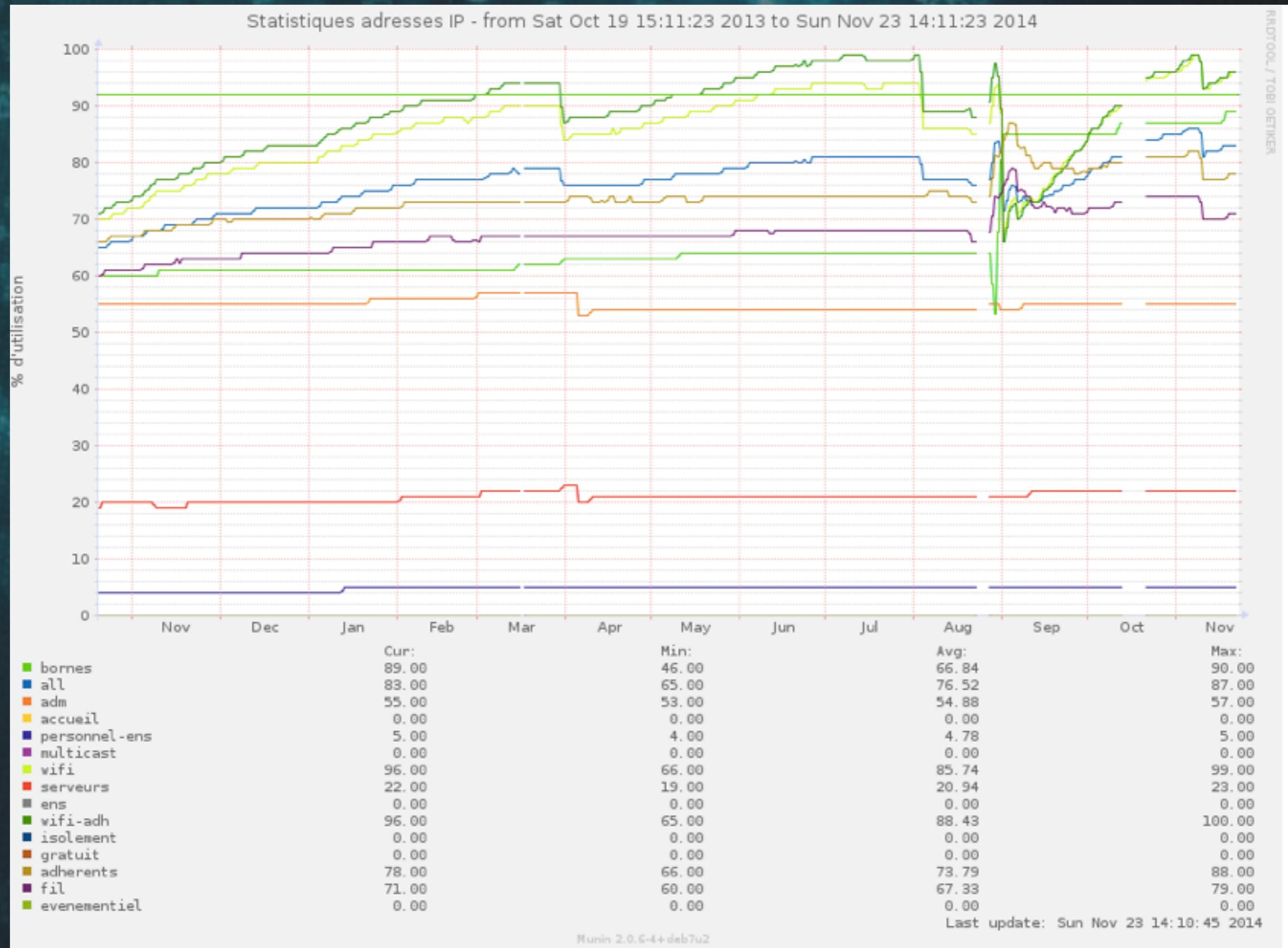
-> Routage classique en IPv4

Exemple : le site de microsoft n'a pas d'IPv6 (étonnant)

L'adhérent veut contacter un site ayant une adresse IPv6 (à l'exception notable de google)

-> Routage effectué par odlyd via le tunnel IPv6.

Mais IPv4 et IPv6 sont incompatibles : pour contacter une adresse IPv4 , je dois avoir moi même une IPv4 (sinon je ne peux pas contacter le site de microsoft... )

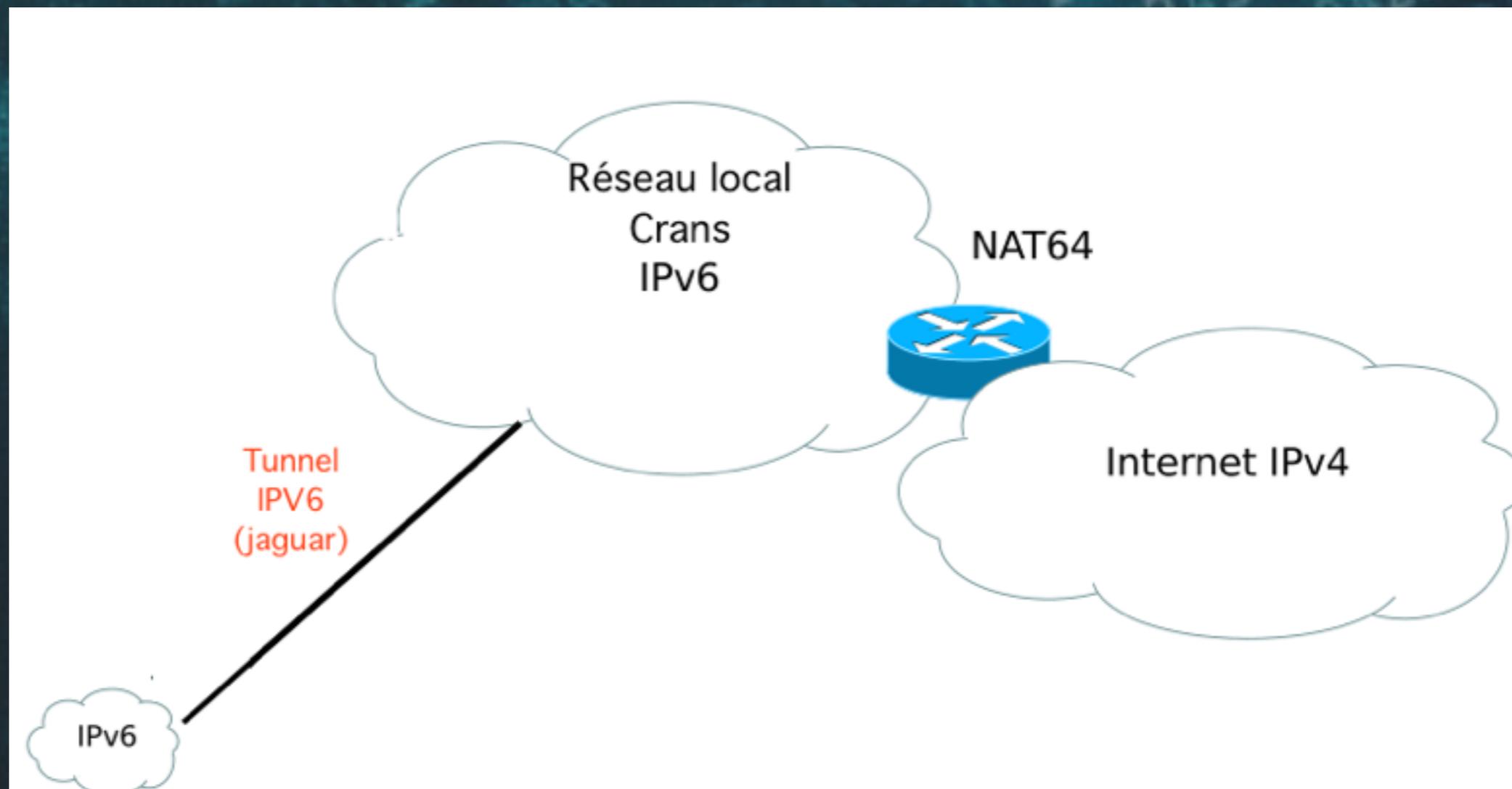


Or :

# Solution : le NAT 64 pour un réseau local IPv6 only

Principe : utilisation unique d'une adresse IPV6 pour la machine :

- Si je veux contacter une adresse IPv6, je passe par le tunnel actuel (pas de changements)
- Si je veux contacter une adresse IPv4, je fais du NAT6->4

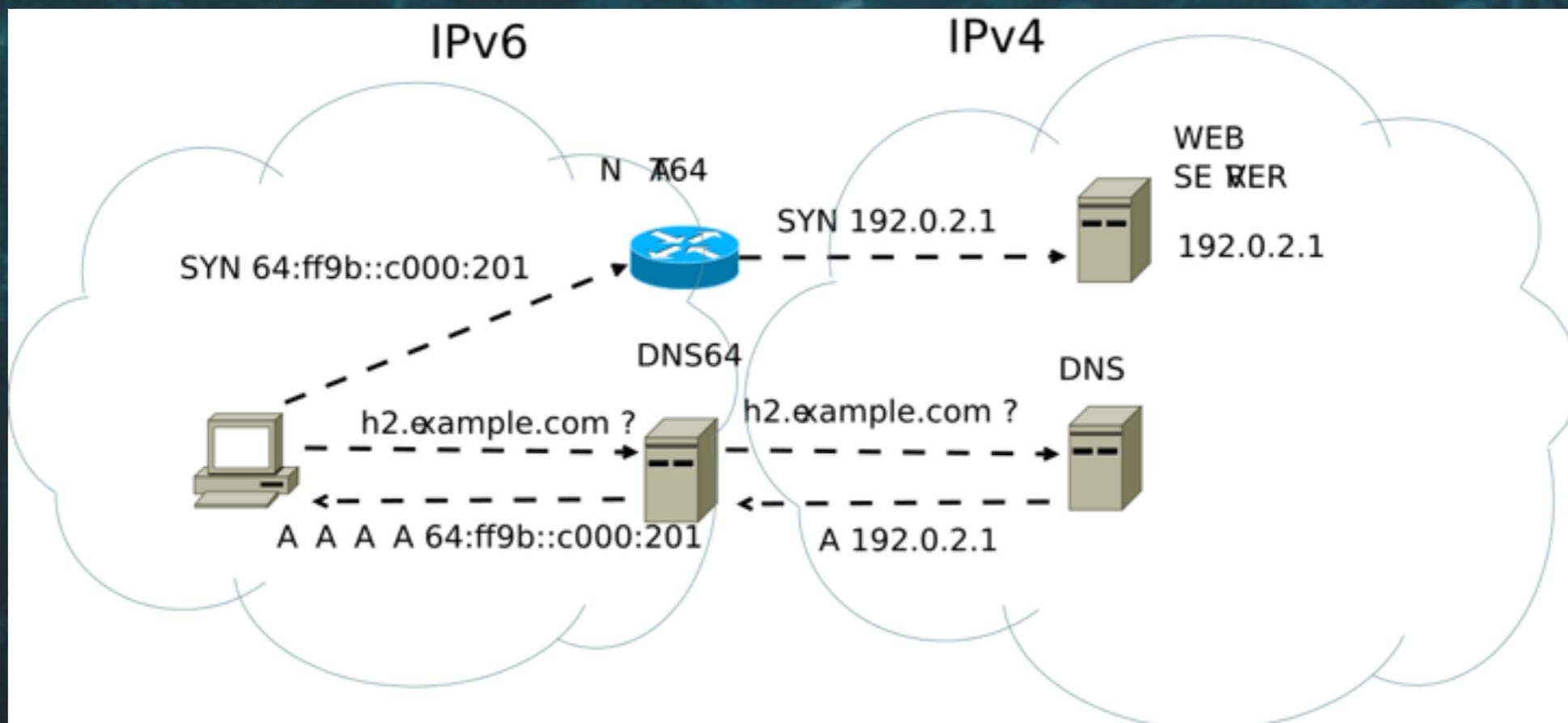


## En détails :

Comment faire pour que la réponse du DNS soit une IPv6 y compris pour le site de microsoft ? (une réponse IPv4 étant exclue)

On utilise un DNS 64 : quand on demande une résolution DNS pour une adresse IPv4, il donne une IPv6 « bidon », contenant l'IPv4, que le serveur NAT 64 saura interpréter.

Il utilise une adresse du type : 64:ff9b::/96 Et 96 + 32 = 128 ...



Work in progress :

Le serveur virtuel NAT64 est en place, et marche

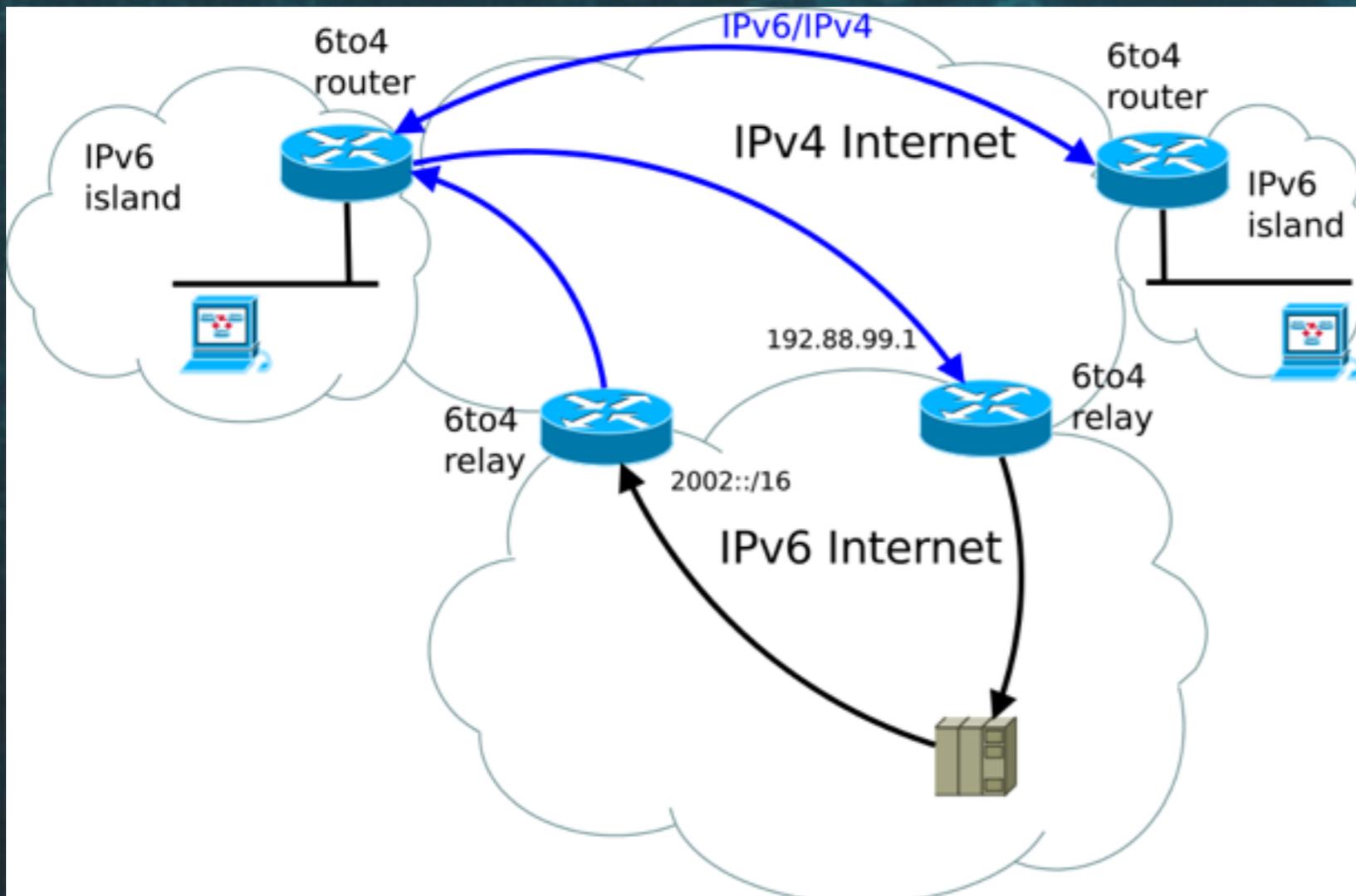
Mais ...

Problèmes pour utiliser le Vlan v6-only

# Autres techniques :

-> 6 to 4 : on encode une IPv4 dans une IPv6  
Pour ce faire, le préfixe 2002:/16 est réservé au 6d4.

On écrit l'adresse IPv4 de destination dans le préfixe. A l'entrée de la zone IPv4, l'hôte aura alors la destination v4 du paquet :



IPv4: 192.0.2.4  
IPv6: 2002:c000:0204::/48

-> Teredo, utiliser l'UDP pour un tunnel classique IPV6

# Conclusion / Questions