

# Intégration & Transition IPv6

**Bruno STEVANT**

**Alain AINA**

# Groupe de travail IETF v6ops

- Définir les processus qui permettent la migration de IPv4 vers IPv6
- Définir et spécifier les mécanismes obligatoires ou optionnels que les constructeurs doivent implémenter dans les hôtes, les routeurs et les autres composants de l'Internet.

[www.ietf.org/html.charters/v6ops-charter.html](http://www.ietf.org/html.charters/v6ops-charter.html)

# Co-existence et Transition IPv4-IPv6

- Un large éventail de techniques :

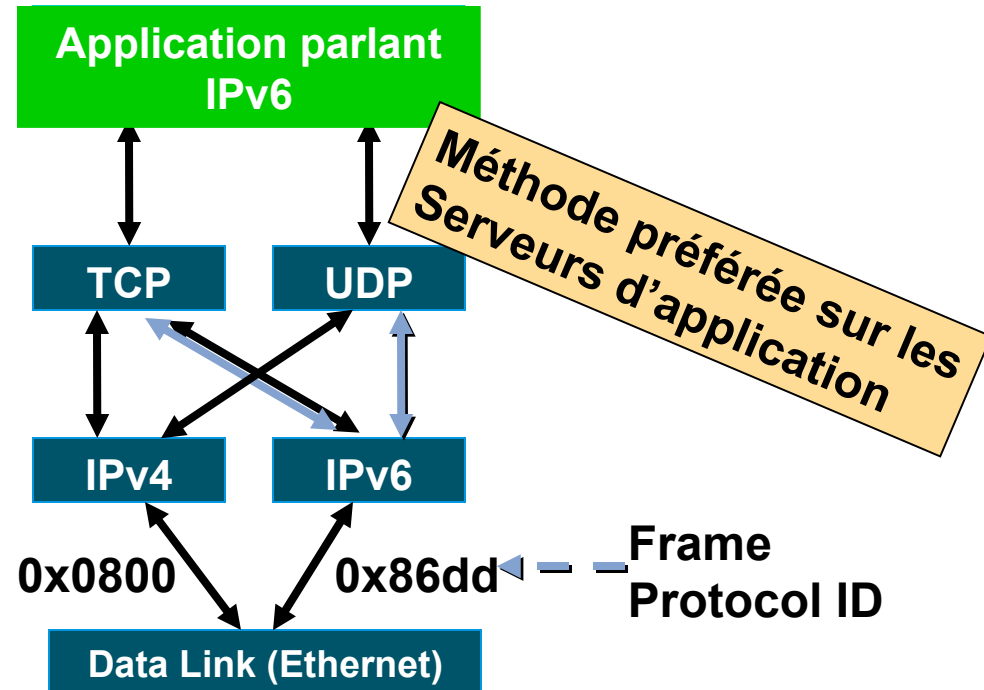
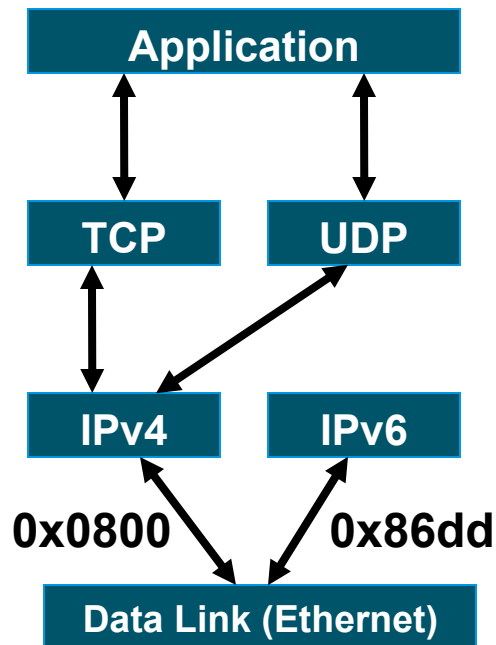
**Techniques Dual-stack**, IPv4 et IPv6 co-existent sur le même noeud

**Techniques de Tunnel**, pour éviter les dépendances dans la déploiement

**Techniques de Translation**, permettre des hôtes pur IPv6 et communiquer à des hôtes pur IPv4

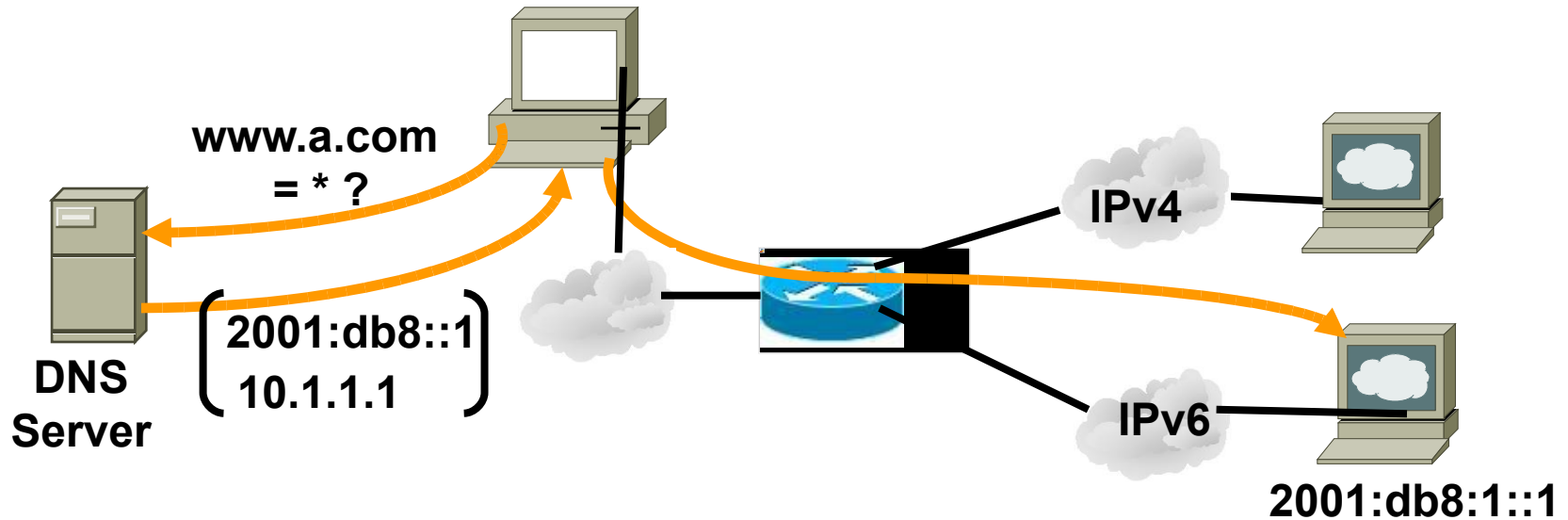
- On utilise les trois en combinaison

# Approche Dual Stack



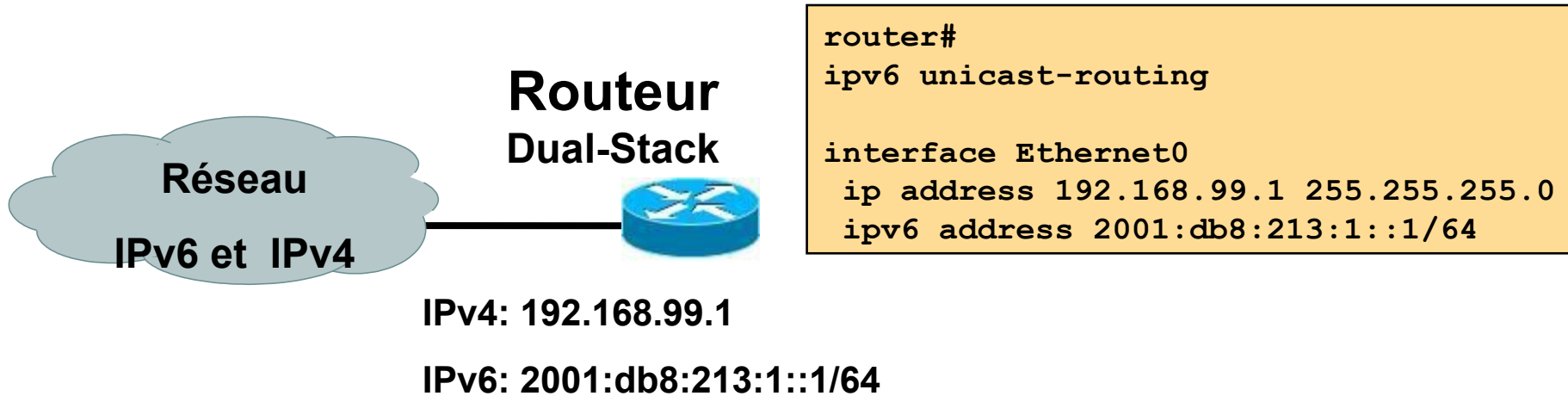
- Dual stack implique:
  - Piles IPv4 et IPv6 activées
  - Les applications communiquent avec IPv4 et IPv6
  - Le choix de la version IP est basé sur le résultat de la requête DNS ou de la préférence de l'application.

# Approche Dual Stack & DNS



- Dans le cas dual stack, une application :  
Qui communique en IPv4 et IPv6  
Demande tous types d'adresses au DNS  
Choisit une adresse, et par exemple, se connecte à l'adresse IPv6

# Configuration Dual Stack



- Routeur IPv6

Si IPv4 et IPv6 sont présents sur la même interface  
Telnet, Ping, Traceroute, SSH, DNS client, TFTP,...

# Tunnels pour le déploiement IPv6

- Plusieurs techniques possibles:

## Configurer manuellement

- Tunnel manuel (RFC 2893)
- GRE (RFC 2473)

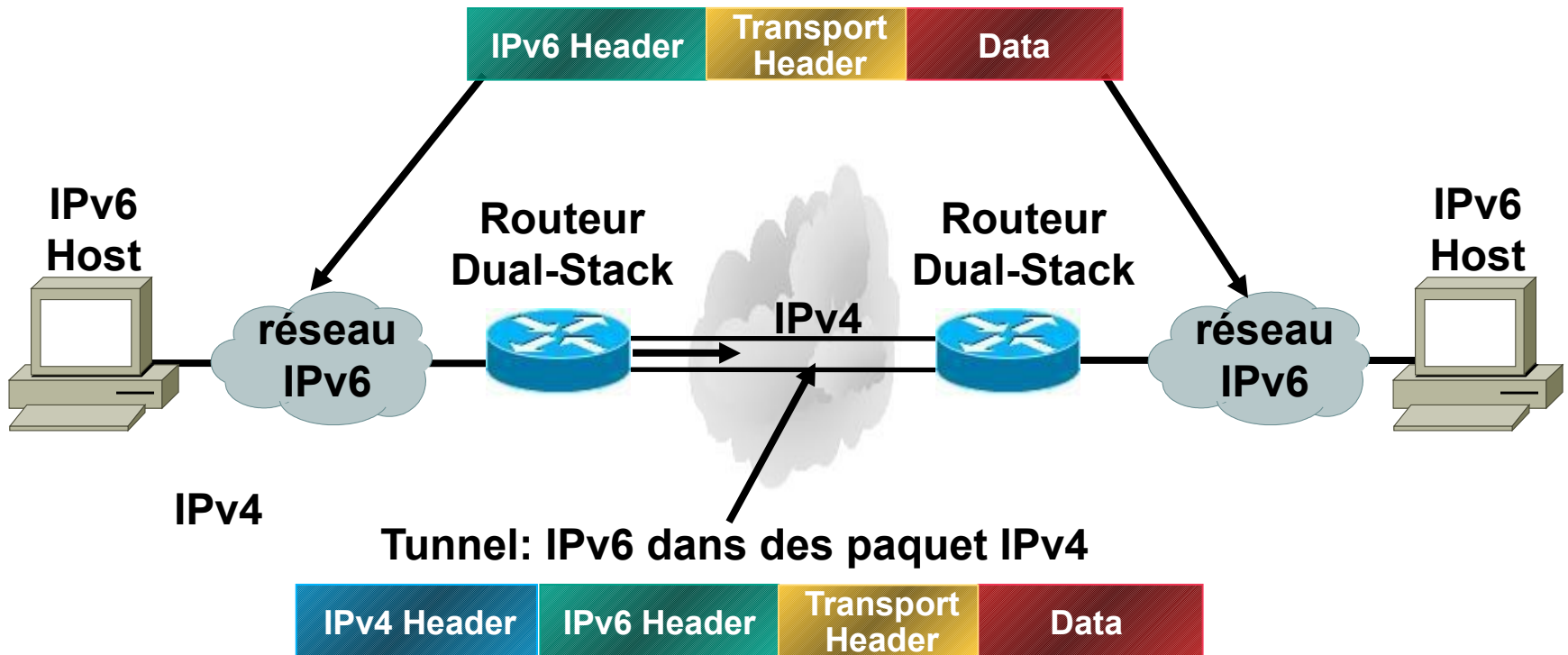
## ➤ Semi-automatiques

Tunnel broker

## ➤ Automatiques

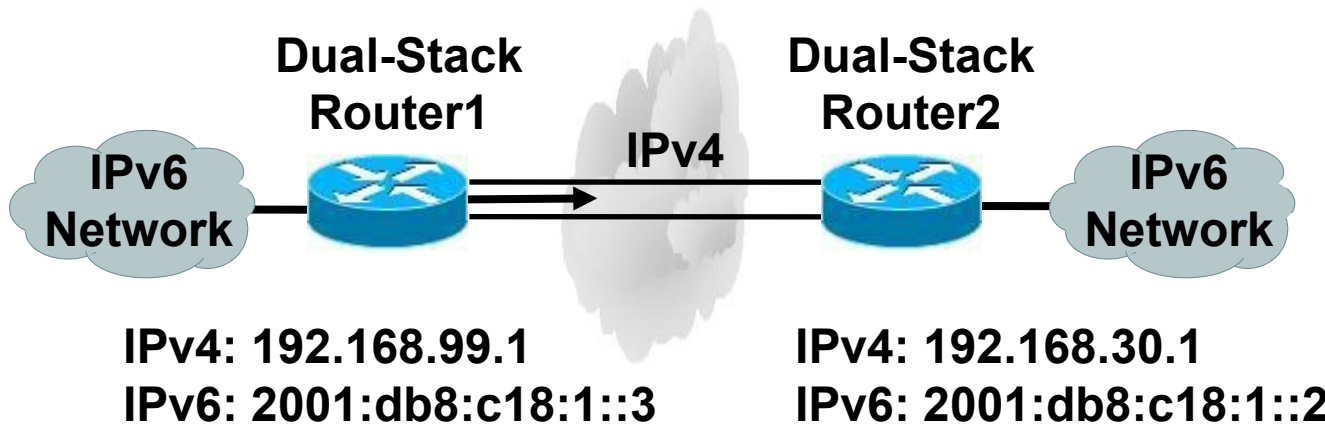
- 6to4 (RFC 3056)
- ISATAP
- 6rd

# Tunnels IPv6 sur IPv4



- Encapsulation des paquets IPv6 dans IPv4
- On peut utiliser cette technique pour des hôtes ou des routeurs.

# Configuration de Tunnel manuel (RFC2893)



```
router1#  
  
interface Tunnel0  
  ipv6 address 2001:db8:c18:1::3/64  
  tunnel source 192.168.99.1  
  tunnel destination 192.168.30.1  
  tunnel mode ipv6ip
```

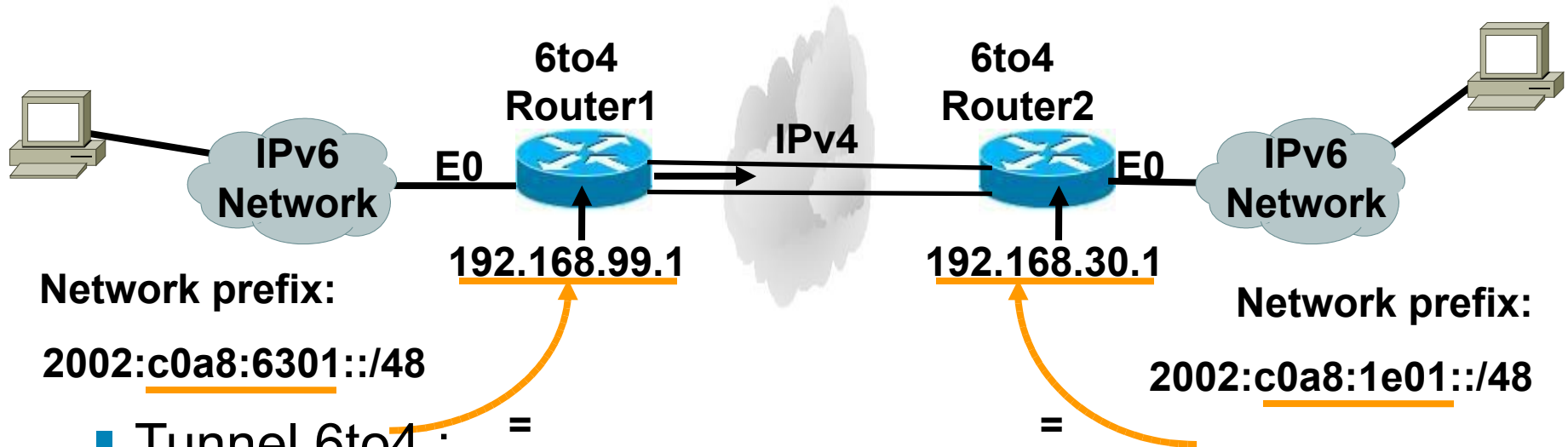
```
router2#  
  
interface Tunnel0  
  ipv6 address 2001:db8:c18:1::2/64  
  tunnel source 192.168.30.1  
  tunnel destination 192.168.99.1  
  tunnel mode ipv6ip
```

- Les tunnels configurés manuellement:

Dual stack aux extrémités

Configuration d'adresses IPv4 et IPv6 aux extrémités

# Tunnel 6to4 (RFC 3056)



- Tunnel 6to4 :

Méthode automatique

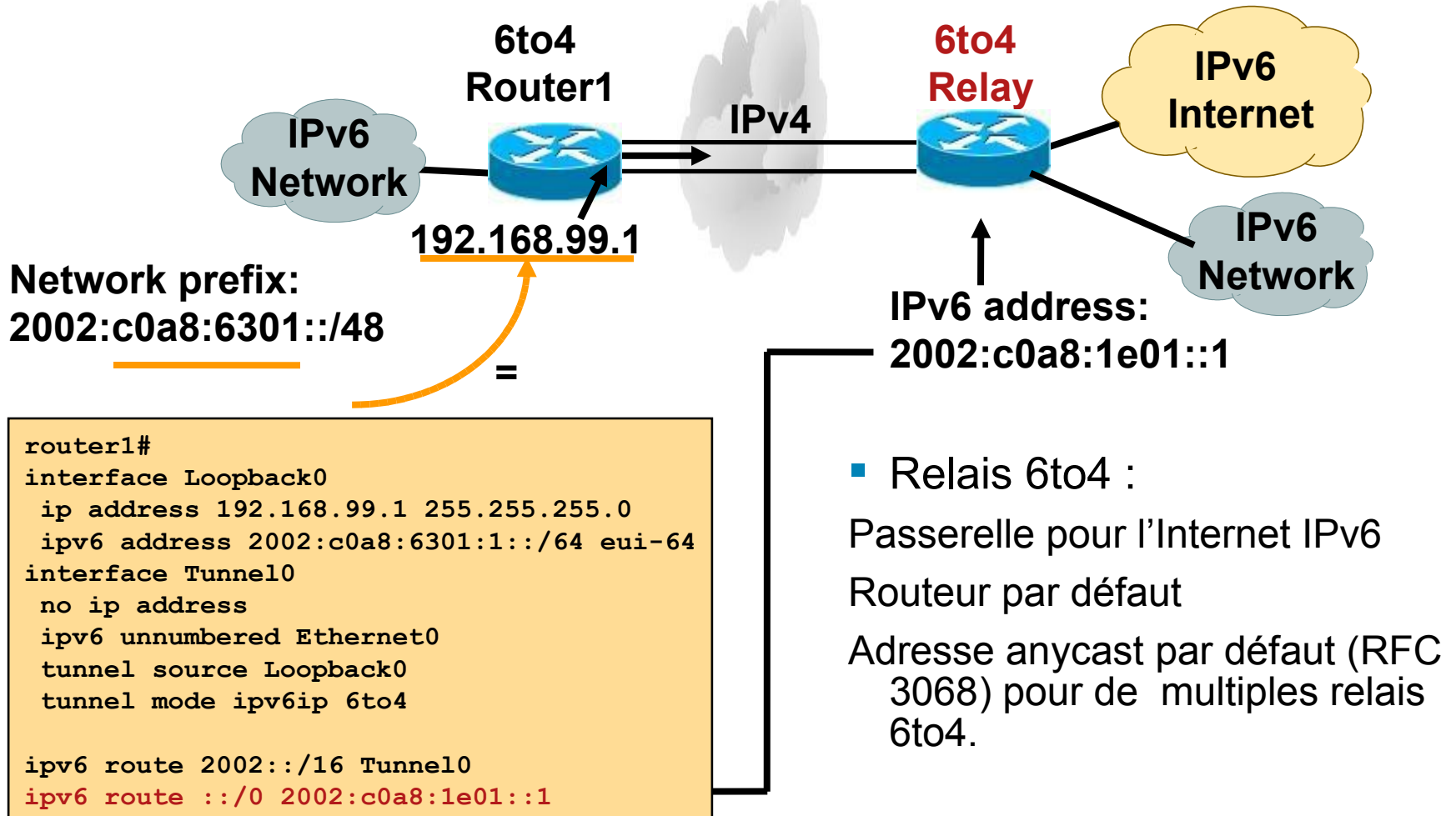
Donne un préfixe au réseau IPv6 relié.

2002::/16 assigné à 6to4

Nécessite une adresse IPv4 globale par site

```
router2#  
interface Loopback0  
 ip address 192.168.30.1 255.255.255.0  
 ipv6 address 2002:c0a8:1e01:1::/64 eui-64  
interface Tunnel0  
 no ip address  
 ipv6 unnumbered Ethernet0  
 tunnel source Loopback0  
 tunnel mode ipv6ip 6to4  
  
ipv6 route 2002::/16 Tunnel0
```

# Relais 6to4



# 6to4 dans l'Internet

- Le préfixe 6to4 est 2002::/16
- 192.88.99.0/24 est le réseau IPv4 anycast pour les routeurs 6to4
- Service de relais 6to4

Un ISP qui fournit les facilités pour offrir la connectivité à travers IPv4 entre des îlots IPv6.

est connecté à l'Internet IPv6 et annonce 2002::/16 via BGP à l'Internet IPv6

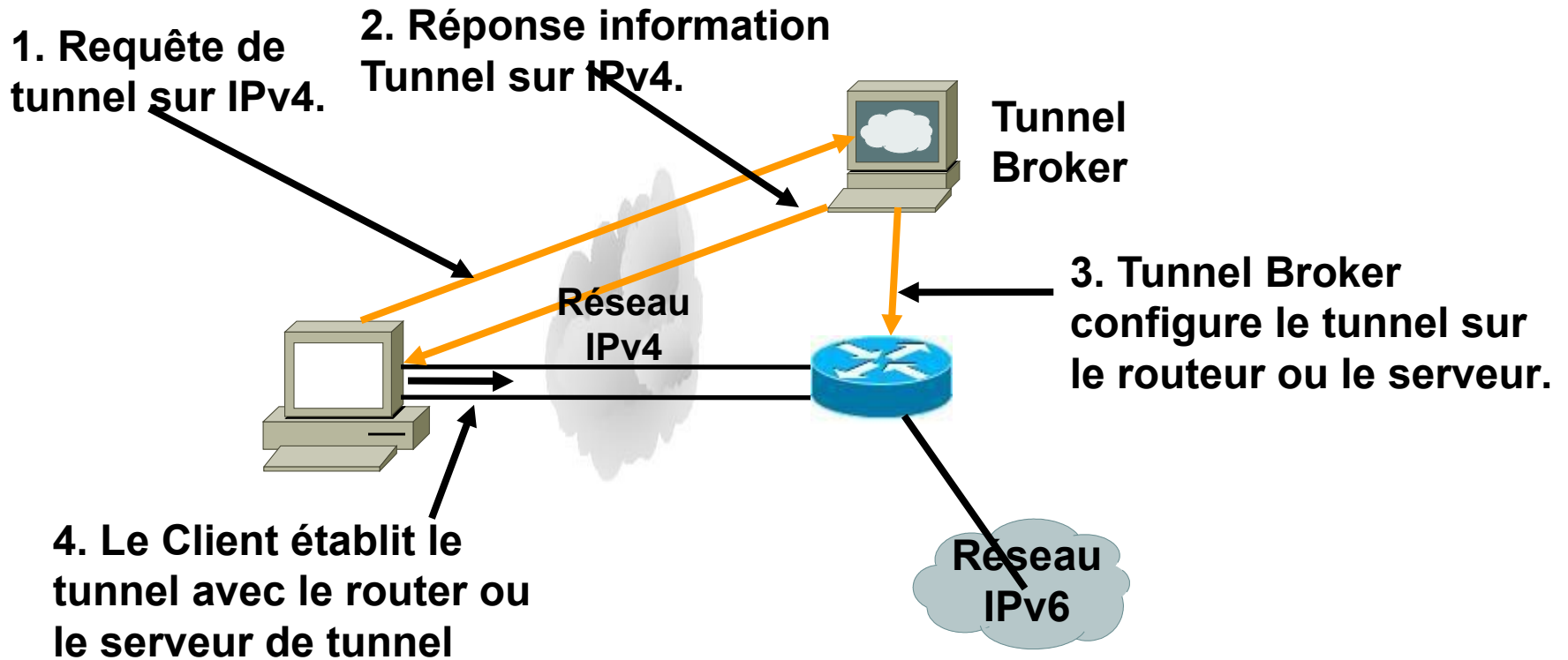
est connecté à l'Internet IPv4 et annonce 192.88.99.0/24 via BGP à l'Internet IPv4

Le routeur est configuré avec l'adresse 192.88.99.1

# configuration 6to4 Relay

```
interface loopback0
  ip address 192.88.99.1 255.255.255.255
  ipv6 address 2002:c058:6301::1/128
!
interface tunnel 2002
  no ip address
  ipv6 unnumbered Loopback0
  tunnel source Loopback0
  tunnel mode ipv6ip 6to4
  tunnel path-mtu-discovery
!
interface FastEthernet0/0
  ip address 105.3.37.1 255.255.255.0
  ipv6 address 2001:db8::1/64
!
router bgp 100
  address-family ipv4
    neighbor <v4-transit> remote-as 101
    network 192.88.99.0 mask 255.255.255.0.
  address-family ipv6
    neighbor <v6-transit> remote-as 102
    network 2002::/16
!
ip route 192.88.99.0 255.255.255.0 null0 254
ipv6 route 2002::/16 tunnel2002
```

# Tunnel Broker



# ISATAP – Intra Site Automatic Tunnel Addressing Protocol

- Tunnelling IPv6 sur IPv4
- Un seul domaine administratif
- Crée un lien virtuel IPv6 sur tout le réseau IPv4
- Tunnel automatique avec une adresse ISATAP qui inclue:
  - Identification ISATAP
  - Adresse IPv4 du noeud
- Les hôtes ISATAP sont dual stack

# Format d'adressage ISATAP

- Une adresse ISATAP d'un noeud est définie comme suit:

Un préfixe /64 dédié au lien ISATAP

Interface ID:

32 bits le plus à gauche = 0000:5EFE:

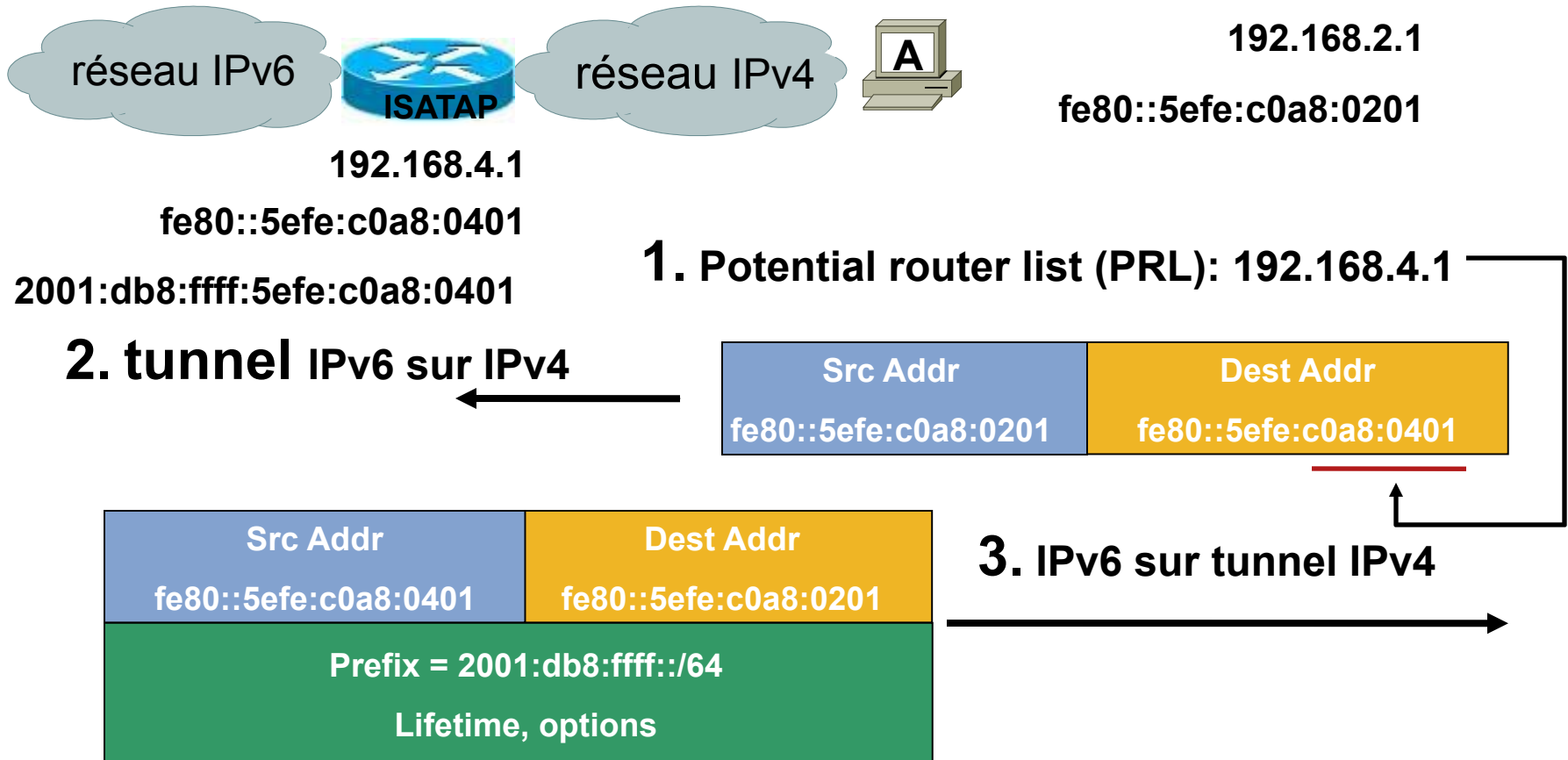
identifie ceci comme une adresse ISATAP

32 bits le plus à droite = <ipv4 adresse>

L'adresse IPv4 du noeud

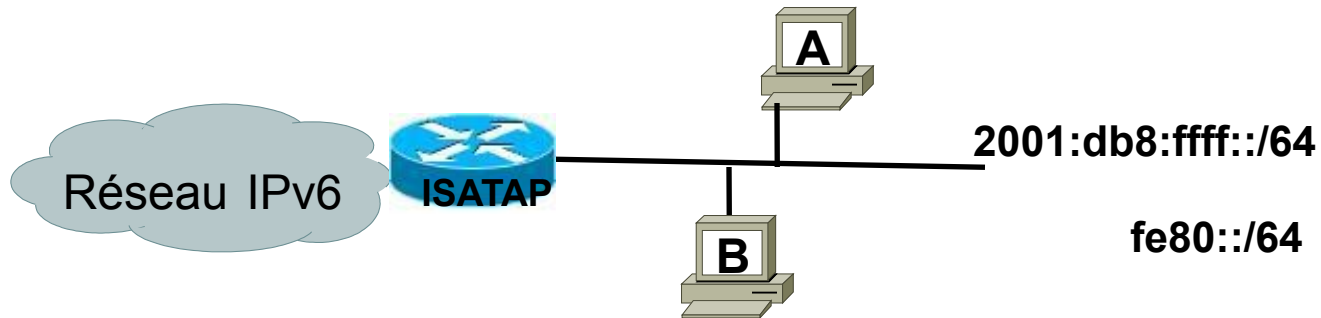
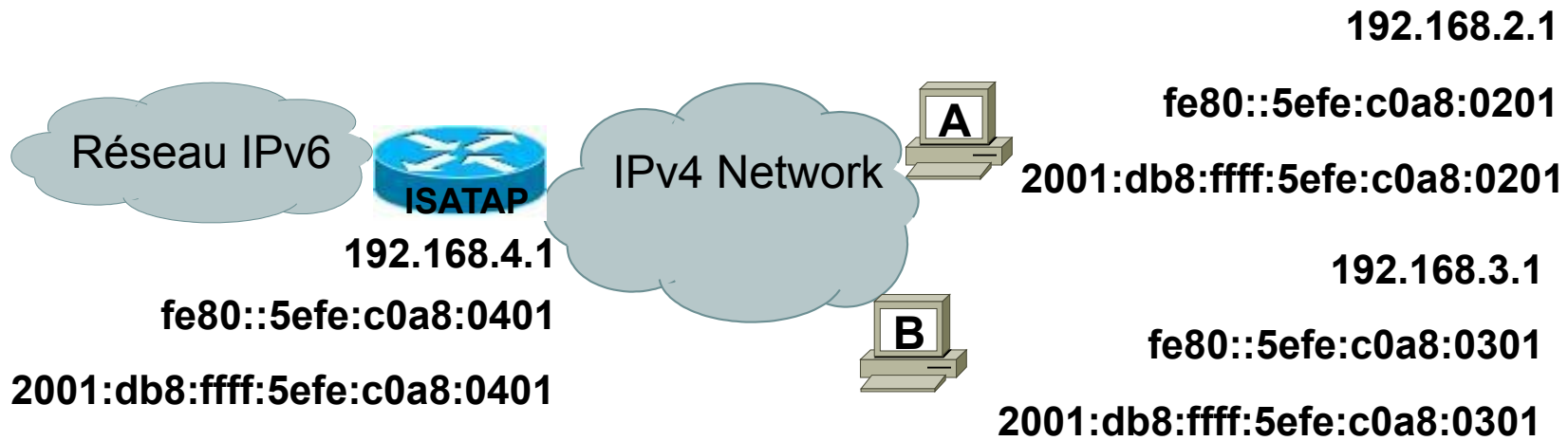
ISATAP dedicated prefix	0000:5EFE	IPv4 address
-------------------------	-----------	--------------

# Annonce du Préfixe ISATAP



4. Hôte A configure l'adresse IPv6 en utilisant le préfixe ISATAP 2001:db8:ffff::/64

# configuration ISATAP



# Mécanismes de Translation IPv6 to IPv4

- Translation
  - NAT-PT (RFC 2766 & RFC 3152)
  - TCP-UDP Relay (RFC 3142)
  - DSTM (Dual Stack Transition Mechanism)
- API
  - BIS (Bump-In-the-Stack) (RFC 2767)
  - BIA (Bump-In-the-API)
- Application Layer Gateway (Passerelle applicative)
  - SOCKS-based Gateway (RFC 3089)
  - NAT-PT (RFC 2766 & RFC 3152)

# NAT-PT pour IPv6

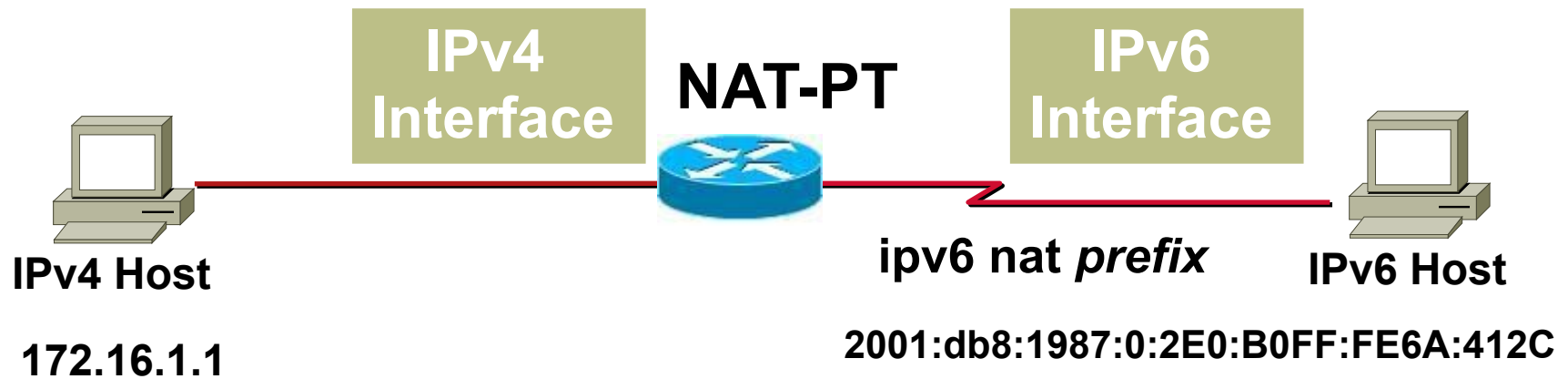
- NAT-PT

(Network Address Translation – Protocol Translation)

RFC 2766 & RFC 3152

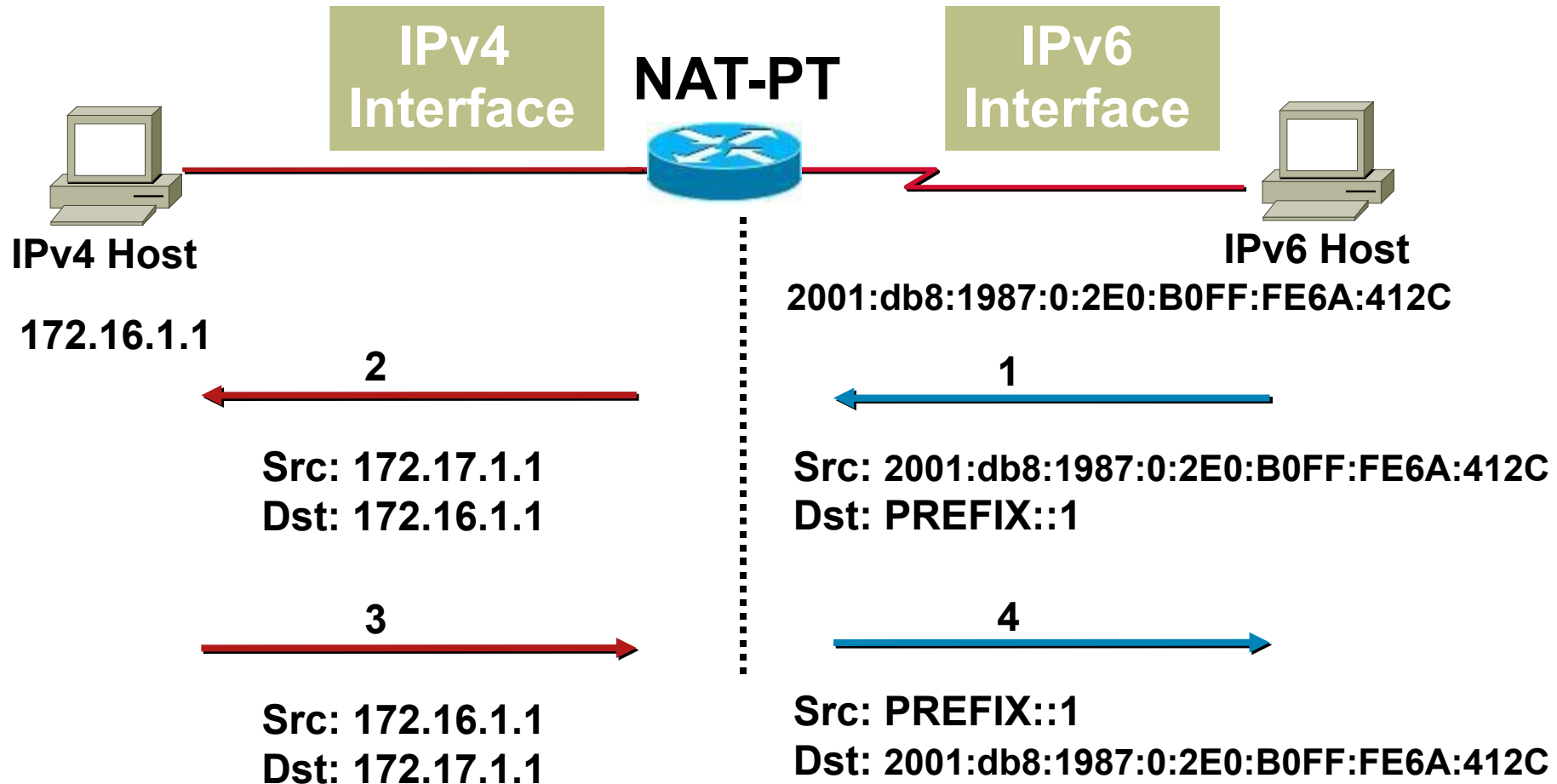
- Permet la communication entre nœud et applications IPv6-only et IPv4-only et vice-versa

# Concept NAT-PT



- *prefix* 96-bit permet de router le traffic vers le noeud IPv6

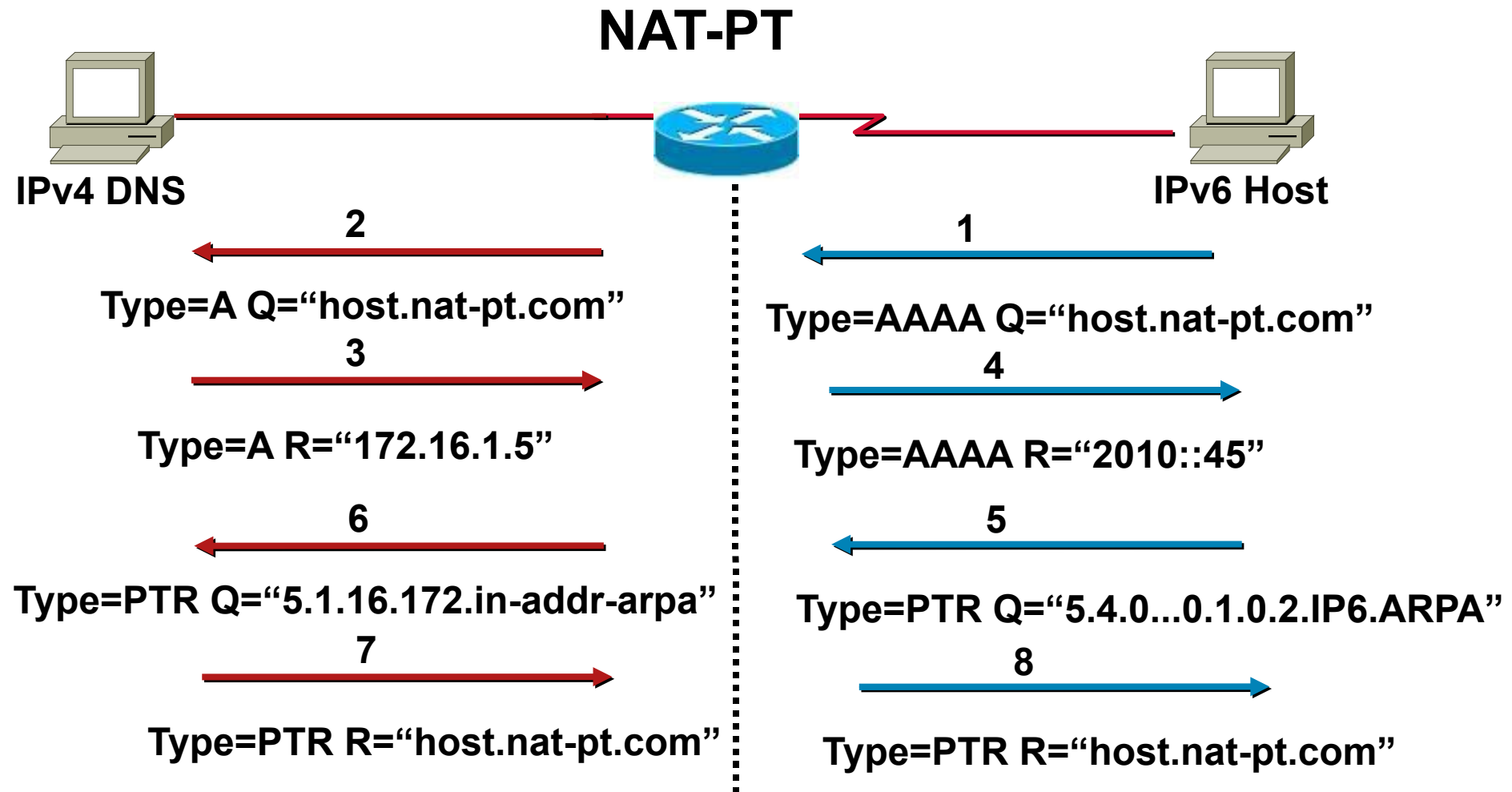
# Flux de paquets NAT-PT



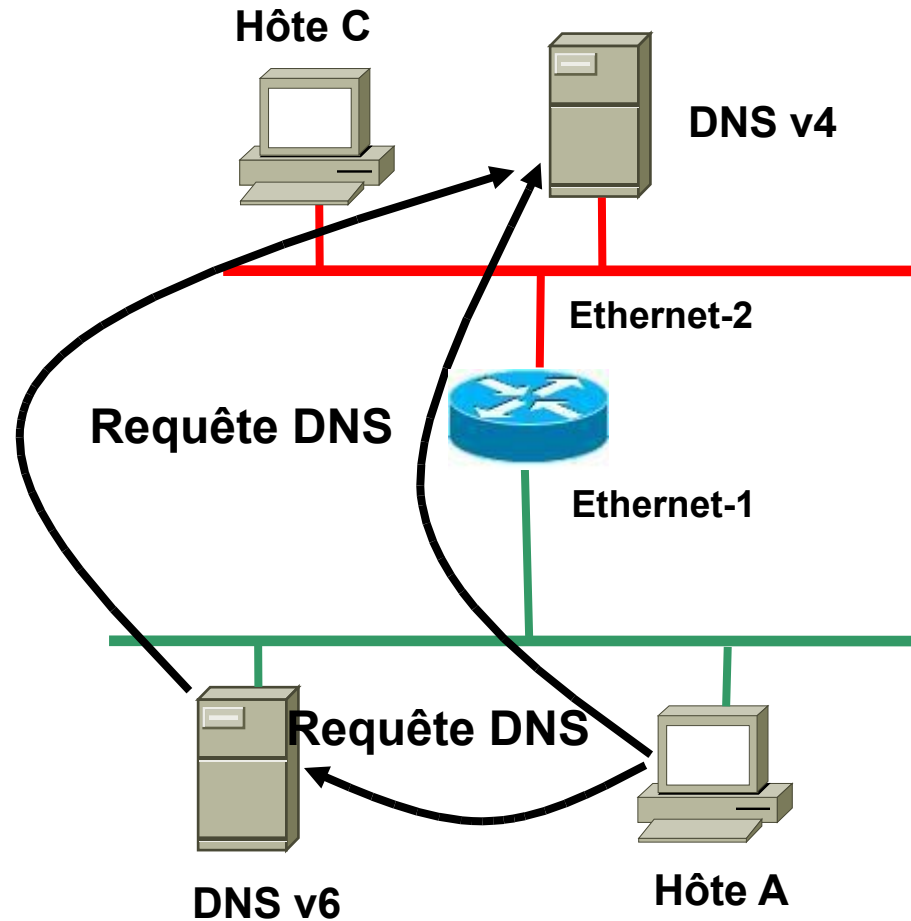
# IP ICMP Translation sans état

<i>IPv6</i>	<i>IPv4</i>	<i>Action</i>
Version = 6	Version = 4	Re-écrit
Traffic class	DSCP	copie
Flow label	N/A	Met à 0
Payload length	Total length	ajuste
Next header	Protocol	copie
Hop limit	TTL	copie

# DNS Application Layer Gateway



# Assignation d'adresse par un DNS ALG



- TTL DNS = 0

# Configurer NAT-PT (1)

- Mise en route NAT-PT

```
[no] ipv6 nat
```

- Configurer global/per interface NAT-PT prefix

```
[no] ipv6 nat prefix <prefix>::/96
```

- Configurer translation statiques

```
[no] ipv6 nat v6v4 source <v6 address> <v4 address>
```

```
[no] ipv6 nat v4v6 source <v4 address> <v6 address>
```

# Configurer NAT-PT (2)

- Configuring translation dynamique

```
[no] ipv6 nat v6v4 source <list,route-map> <ipv6 list,  
route-map> pool <v4pool>
```

```
[no] ipv6 nat v6v4 pool <v4pool> <ipv4 addr> <ipv4addr>  
prefix-length <n>
```

- Configure le maximum des translations

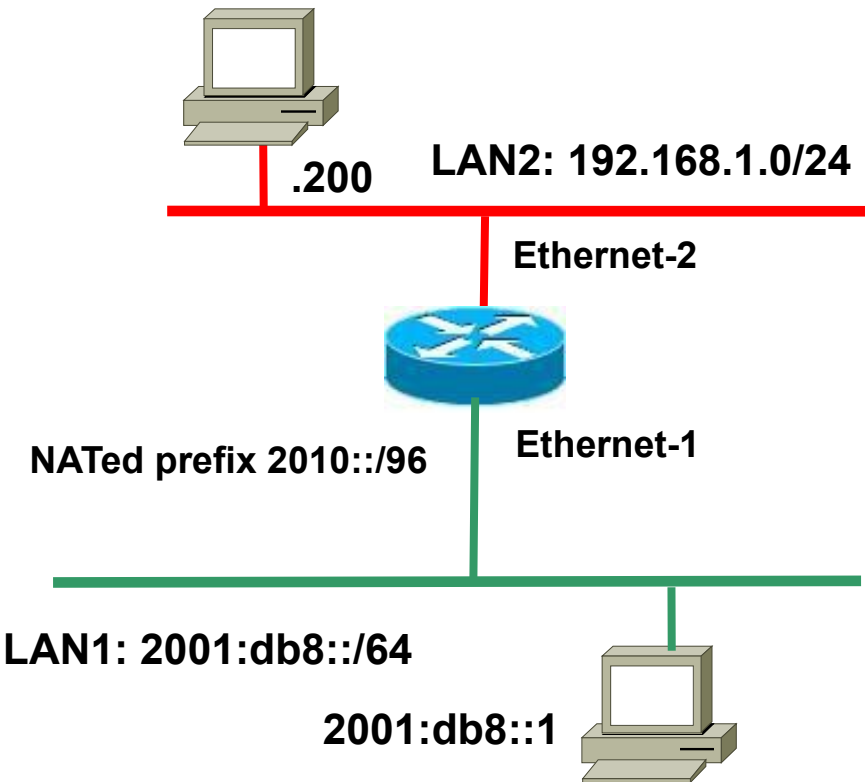
```
[no] ipv6 nat translation max-entries <n>
```

- Commandes de Debug

```
debug ipv6 nat
```

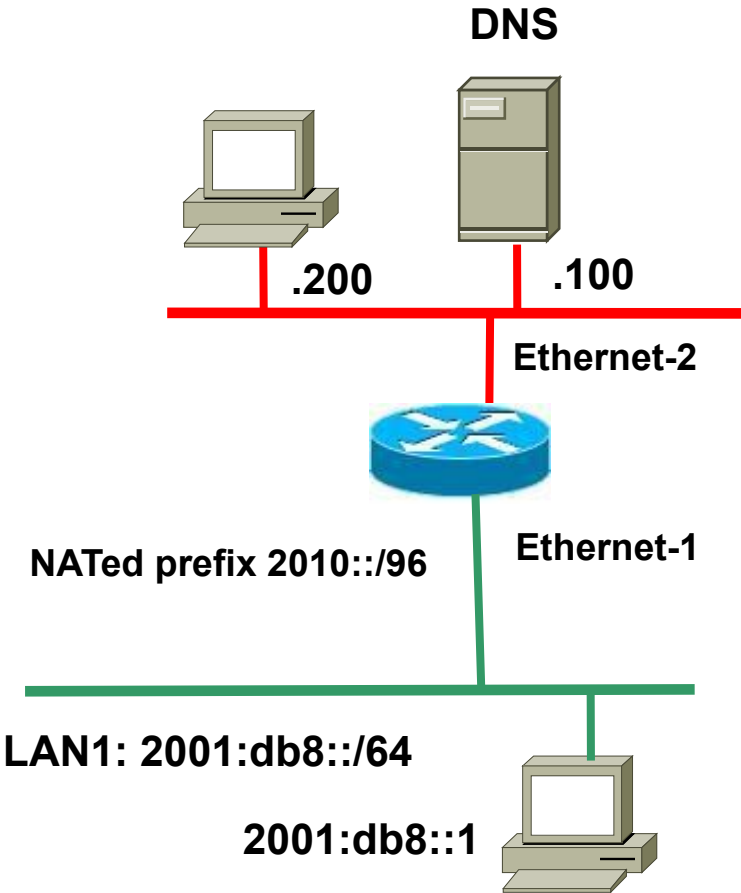
```
debug ipv6 nat detailed
```

# Exemple de configuration Cisco IOS NAT-PT



```
interface ethernet-1
  ipv6 address 2001:db8::10/64
  ipv6 nat
!
interface ethernet-2
  ip address 192.168.1.1 255.255.255.0
  ipv6 nat prefix 2010::/96
  ipv6 nat
!
ipv6 nat v6v4 source 2001:db8::1 192.168.2.1
ipv6 nat v4v6 source 192.168.1.200 2010::60
!
```

# Configuration Cisco IOS NAT-PT avec DNS ALG

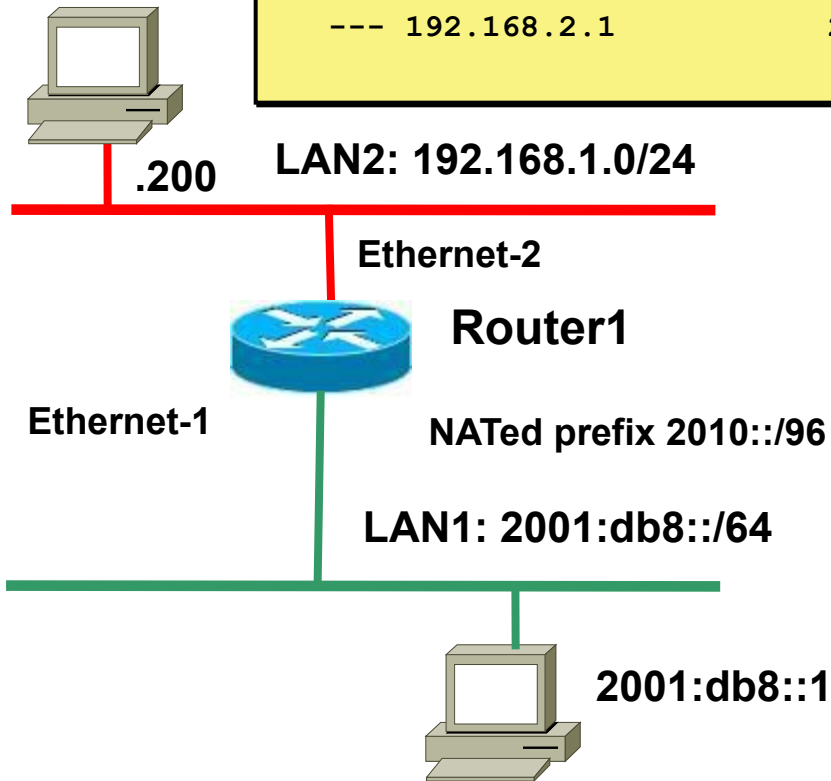


```
interface ethernet-1
  ipv6 address 2001:db8::10/64
  ipv6 nat
!
interface ethernet-2
  ip address 192.168.1.1 255.255.255.0
  ipv6 nat
  ipv6 nat prefix 2010::/96
!
ipv6 nat v4v6 source 192.168.1.100 2010:::1
!
ipv6 nat v6v4 source route-map map1 pool v4pool1
ipv6 nat v6v4 pool v4pool1 192.168.2.1 192.168.2.10
prefix-length 24
!
route-map map1 permit 10
  match interface Ethernet-2
```

# Cisco IOS NAT-PT display (1)

```
Router1 #show ipv6 nat translations
```

Pro IPv4 source	IPv6 source	IPv6 destn	IPv4 destn
---	---	2010::60	192.168.1.200
---	192.168.2.1	2001:db8::1	---



# Résumé NAT-PT

- Point d'attention:

Une ALG par application incluant des adresses

Pas de sécurité de bout en bout

Pas de DNSSEC

Pas d'IPsec car les adresses changent

# Sécurité IPv6

Bruno STEVANT

Alain AINA

# Agenda

- Problèmes communs à IPv4 et IPv6
- Problèmes spécifiques à IPv6

IPsec partout, dual-stack, tunnels et IPv6 Mobile

- Règles communes pour la sécurité IPv6
- Faire respecter les règles de sécurité IPv6

ACL et Firewalls

- Sécuriser le déploiement sur une infrastructure publique

# Problèmes communs



# Reconnaissance en IPv4

Facile à faire

1. DNS/IANA crawling (whois) pour connaître les réseaux
2. Ping et scanning de ports
3. Scanner les ports vulnérables

```
[tick:/var] scott# nmap -sP 10.1.1.0/24
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host (10.1.1.0) seems to be a subnet broadcast ...
Host (10.1.1.1) appears to be up.
Host (10.1.1.12) appears to be up.
Host (10.1.1.22) appears to be up.
Host (10.1.1.23) appears to be up.
Host (10.1.1.101) appears to be up.
Host (10.1.1.255) seems to be a subnet broadcast ...
Nmap run completed -- 256 IP addresses (7 hosts up)
scanned in 4 seconds
```

# Reconnaissance en IPv6

## Différence de taille de subnet

- En IPv6 le sous-réseau de base a  $2^{64}$  adresses  
10 Mpps = plus de 50 000 ans
- NMAP ne marche pas sur IPv6

# Reconnaissance en IPv6

## Changement de méthode

- Les serveurs publics seront toujours accessibles via le DNS
  - ⇒ Plus d'informations collectées via Google...
  - ⇒ Cf SensePost BiDiBLAH
- Avec l'augmentation des déploiements IPv6 et l'utilisation du DNS dynamique
  - => Plus d'informations seront disponibles dans le DNS
- Les administrateurs fainéants vont utiliser des plans d'adressages faciles (:::10,:::20,:::F00D, :::C5C0 ou simplement mettre l'IPv4 dans l'adresse IPv6)
- Dès qu'un hôte est compromis, il est facile de connaître les autres hôtes.

# Reconnaissance en IPv6

## Nouvelles Adresses Multicast

- Tous les routeurs (FF05::2) et tous les serveurs DHCP (FF05::1:3)

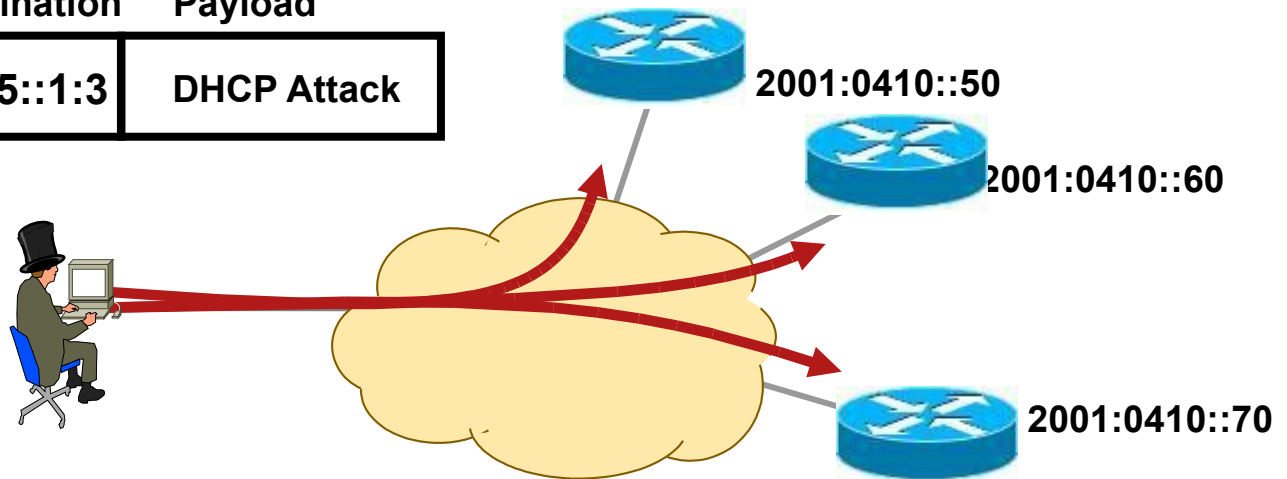
Pas besoin de reconnaissance

- Quelques adresses obsolètes site-local encore utilisées

FEC0:0:0:FFFF::1 DNS server

- **Prévoir un filtrage de ces adresses au bord du réseau**

Source	Destination	Payload
Attacker	FF05::1:3	DHCP Attack



# Virus et Worms (vers) en IPv6

- Virus et email worms: pas de changement en IPv6

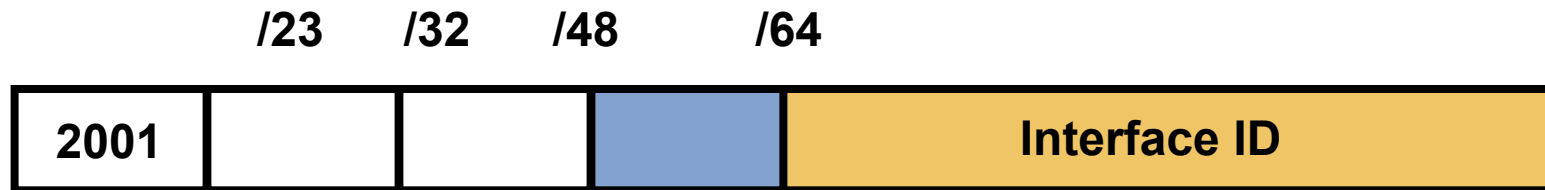
- Autres Vers:

IPv4: scan réseau

IPv6: pas si facile (**voir reconnaissance**) => autres techniques

- Les développeurs de Vers vont s'adapter à IPv6
- Ce qui a été dit en IPv4 pour combattre les Vers reste valide
- Attaques de CPU en cas de scan réseau  
Les routeurs feront du ND (Neighbor Discovery)

# IPv6 Privacy Extensions (RFC 3041)



- Adresses IPv6 pour les noeuds, ex. web browser

On ne peut plus localiser les utilisateurs

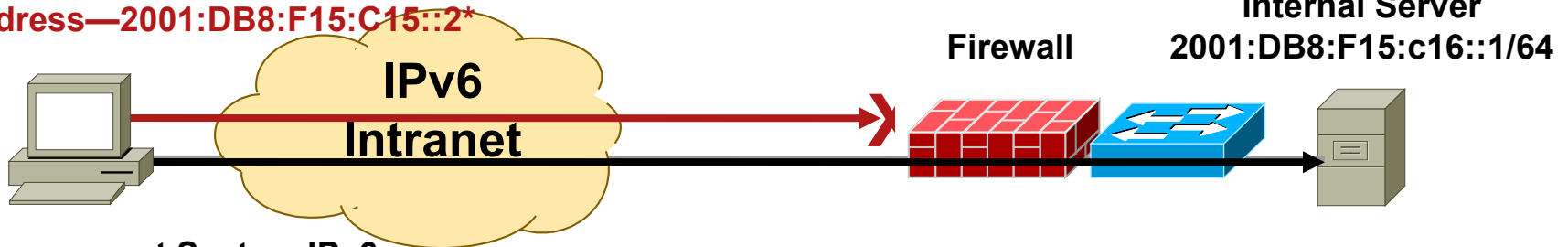
64 bit interface ID au hasard , puis Duplicate Address Detection  
L'adresse peut changer à la vitesse nécessaire

**Recommandation: utiliser Privacy Extensions pour la communication externe mais pas interne (difficile à diagnostiquer)**

# Contrôle d'accès pour IPv6 Privacy Extension

- Bon pour protéger l'intimité d'un noeud ou d'un utilisateur
- Difficile de définir une politique quand les adresses changent tout le temps

Management System New IPv6  
Address—2001:DB8:F15:C15::2\*



Management System IPv6  
Address—2001:DB8:F15:C15::1\*

Action	Src	Dest	Src Port	Dst Port
Permit	2001:DB8:F15:C15::1	2001:DB8:F15:c16::1	Any	80
Deny	Any	Any		

\*Not real RFC3041 derived addresses

# Supprimer Privacy Extension

- Microsoft Windows

Utiliser Group Policy Object (GPO)

Ou

```
netsh interface ipv6 set global randomizeidentifiers=disabled  
netsh interface ipv6 set global randomizeidentifiers=disabled store=persistent  
netsh interface ipv6 set privacy state=disabled store=persistent
```

- Alternatives

Utiliser DHCP avec un pool spécifique (voir plus loin)

Prendre ce pool comme base de filtrage

# L3-L4 Spoofing en IPv4

- Le spoofing niveau 4 peut être fait en même temps qu'au niveau 3 (ex. UDP, i.e. SNMP, Syslog, etc.)
- Une partie des adresses IPv4 ne sont pas allouées ou réservées (RFC3330) ; cela rend facile le filtrage en entrée avec les filtres bogons.

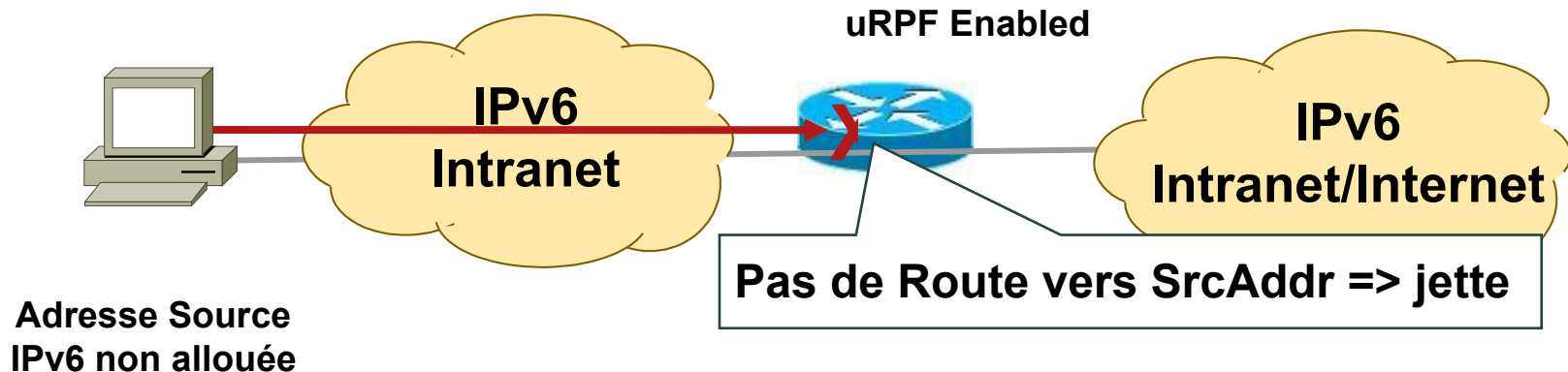
# Filtres d'adresses IPv6 à proscrire

- En IPv4, on bloque les adresses inutiles
- En IPv6, il n'y a que peu d'adresse allouée

On autorise les adresses déclarées

- IPv6 et IPv4 sont sur un pied d'égalité

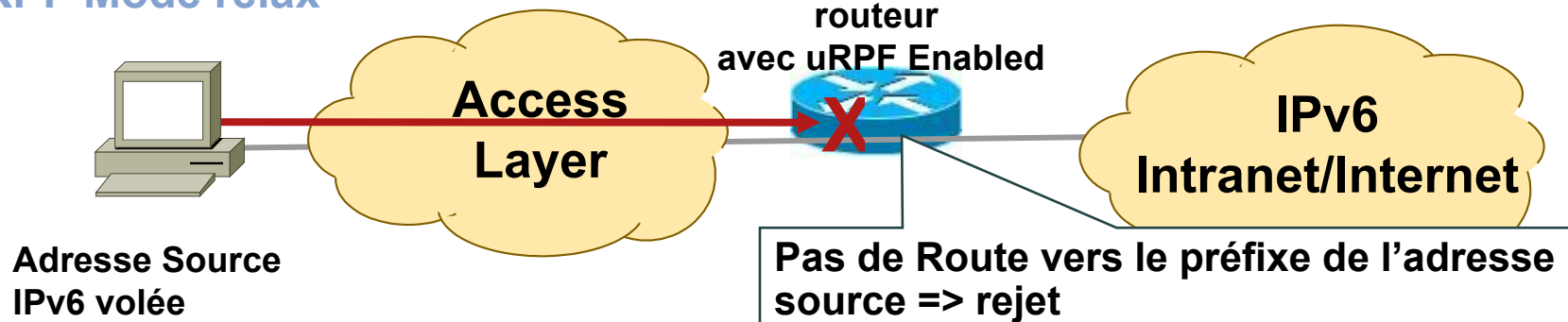
=> Même technique = uRPF



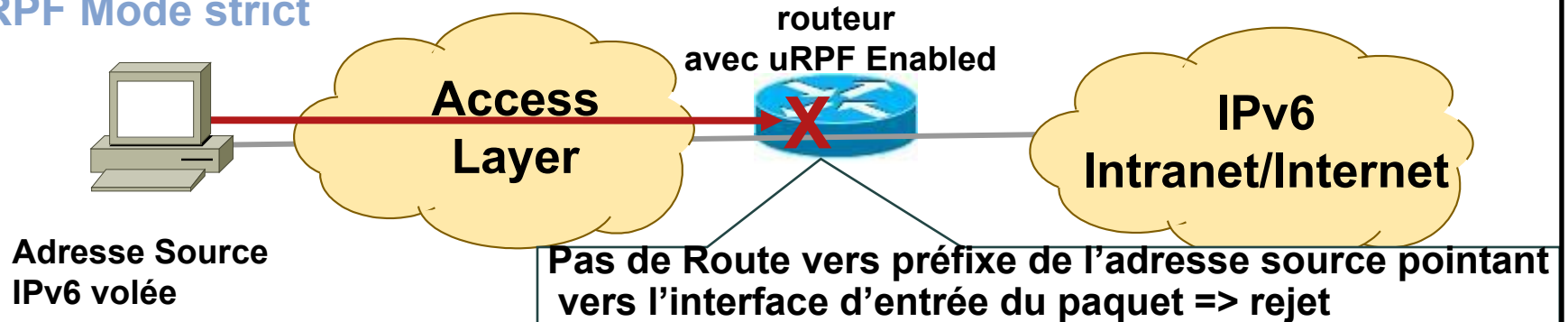
# L3 Spoofing pour IPv6

uRPF est l'outil de base pour contrer cette attaque

## uRPF Mode relax



## uRPF Mode strict



# ICMPv4 vs. ICMPv6

- De grands changements
- Plus au centre du protocole

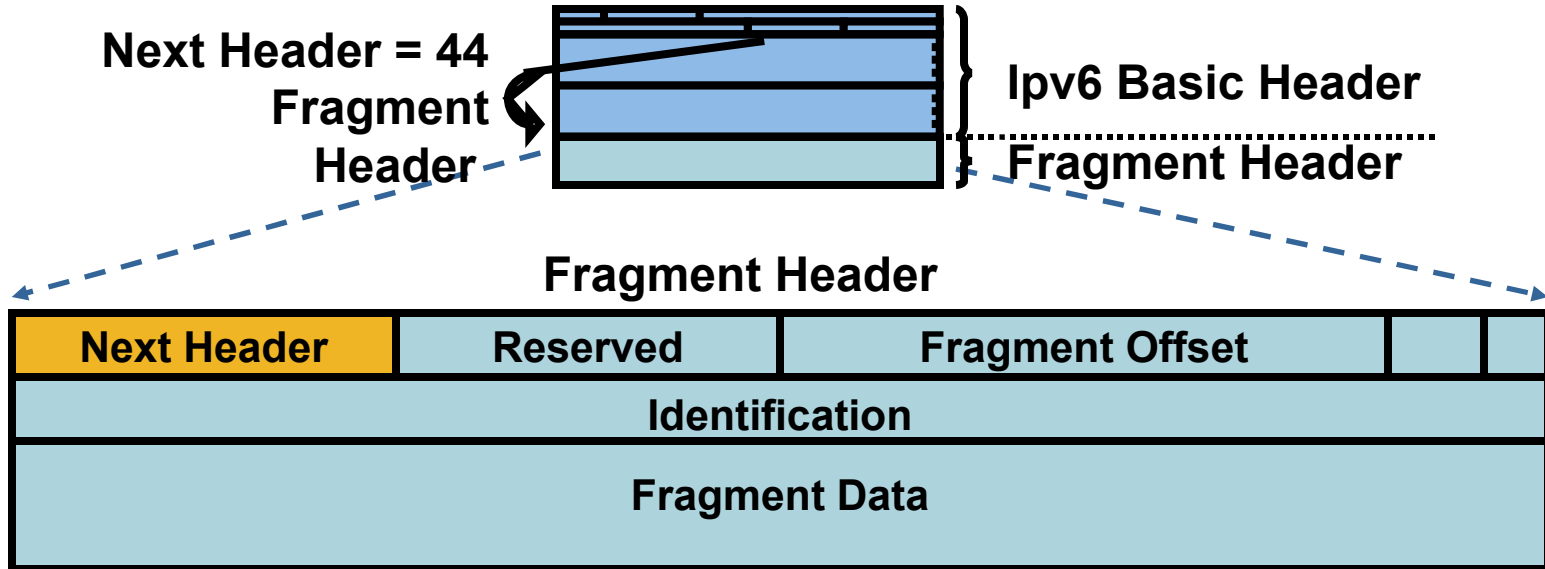
ICMP Message Type	ICMPv4	ICMPv6
Connectivity Checks	X	X
Informational/Error Messaging	X	X
Fragmentation Needed Notification	X	X
Address Assignment		X
Address Resolution		X
Multicast Group Management		X
Mobile IPv6 Support		X

- => La politique ICMP sur les firewalls doit être changée
- Voir RFC 4890

# Fragmentation utilisée dans les attaques IPv4

- Pour éviter les règles de filtrage
- Outils disponibles : whisker, fragrout, etc.
- Rends la tâche des firewalls et des IDS difficile
- Souvent Utiliser dans les attaques DoS contre un nœud mais peut être utilisé dans les attaques qui compromettent les noeuds.

# Entête fragment: IPv6



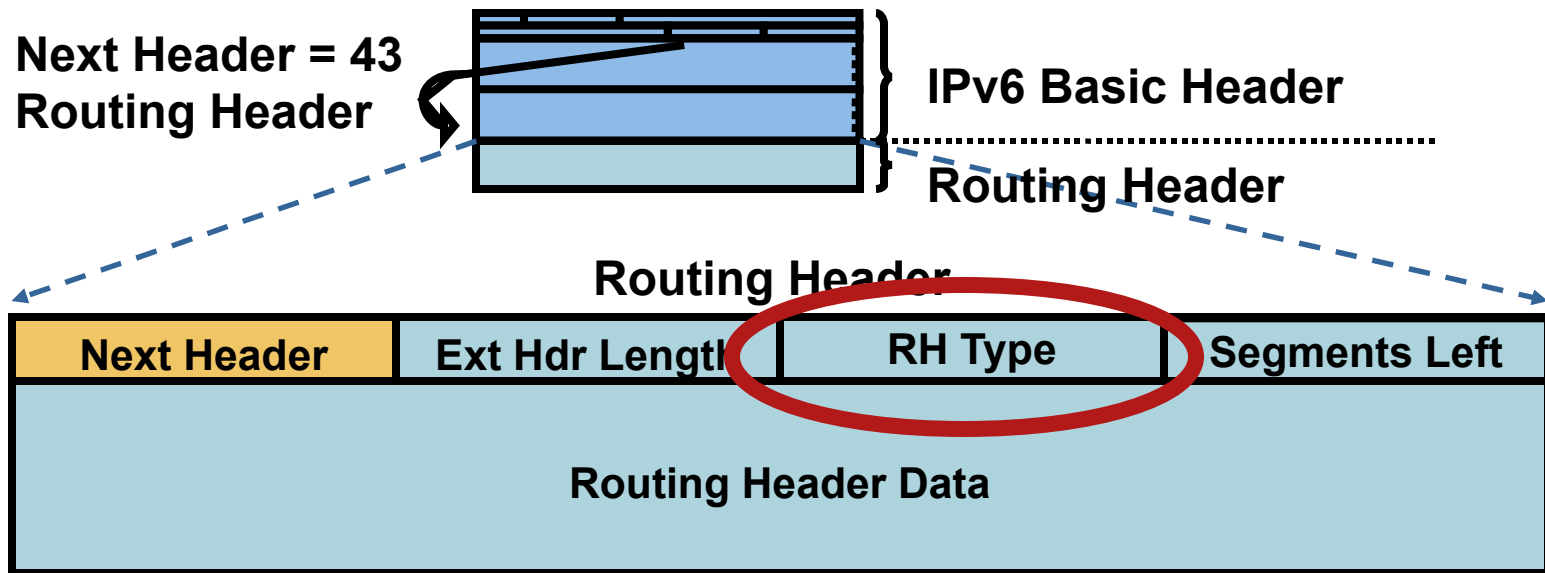
- La fragmentation est réalisée par les extrémités du flux
- Le réassemblage se fait par l'hôte d'extrémité comme enIPv4
- Un pirate peut fragmenter en chemin

# Entête Routage IPv6

- Traité par les intermédiaires cités
- deux types

Type 0: comme IPv4 source routing (routeurs intermédiaires multiples)

Type 2: mobile IPv6



# Prévenir les attaques contre l'entête de routage

- Appliquer la même politique pour IPv6 et IPv4: bloquer l'entête de routage de type 0
- Sur les routeurs du noyau  

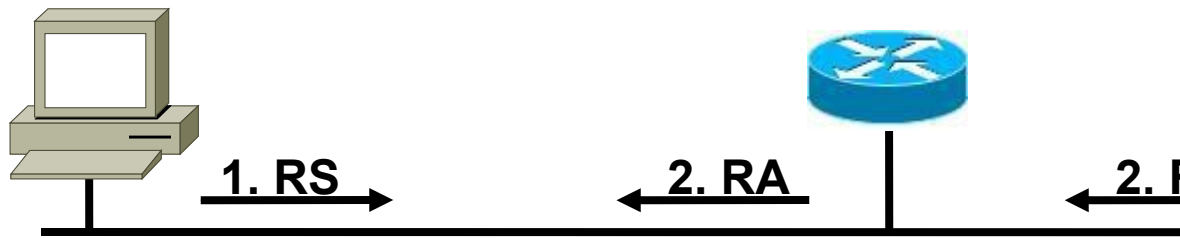
```
no ipv6 source-route
```
- Et à la périphérie

Avec un ACL bloquant les entêtes de routage

# Autoconfiguration sans état

**Router Solicitations** sont envoyés par les hôtes qui démarrent

ICMP et ARP ont le même niveau de sécurité



Outil d'attaque:  
**fake\_router6**

Peut faire de n'importe quelle adresse IPv6 le routeur par défaut

## 1. RS:

Src = ::

Dst = All-Routers  
multicast Address

ICMP Type = 133

Data = Query: please send RA

## 1. RA:

Src = Router Link-local Address

Dst = All-nodes multicast address

ICMP Type = 134

Data= options, prefix, lifetime,  
**autoconfig** flag

# Sollicitation de voisin



Src = A  
Dst = Solicited-node multicast of B  
ICMP type = 135  
Data = link-layer address of A  
Query: what is your link address?

Src = B  
Dst = A  
ICMP type = 136  
Data = link-layer address of B

**A et B peuvent maintenant se parler sur ce lien**

**Pas de mécanisme de sécurité**

**=> qu'est ce qu'on fait pour démarrer**

**Outil d'attaque:**  
**Parasite6**  
**Répond a tous les NS et prétend être ce noeud**

# Détection de duplication d'adresse

**Duplicate Address Detection (DAD)** utilise la sollicitation de voisin pour s'assurer de l'unicité de l'adresse



Src = :: 

Dst = Solicited-node multicast of **A**

ICMP type = 135

Data = link-layer address of A

Query = what is your link address?



**From RFC 2462:**  
« Si une adresse dupliquée est découverte elle ne peut pas être assignée »  
⇔???: si vous utilisez l'adresse MAC de la victime pour fabriquer son adresse IPv6

**Outil d'attaque:**  
**Dos-new-ipv6**

# Attaques IPv6 très proche de IPv4

- **Sniffing**

Sans IPsec, IPv6 est aussi vulnérable aux attaques de sniffing que IPv4.

- **Attaque de la couche applicative**

Ipssec ne peut rien contre une attaque applicative

- **Hôte malicieux**

L'insertion d'hôte malicieux est aussi facile qu'en IPv4

- **Attaque Man-in-the-middle**

Sans IPsec, une attaque Man-in-middle à la même chance de réussir en IPv6 qu'en IPv4.

- **Inondation**

Identique

# C'est réel ☹️

## Outils d'attaque IPv6

- Sniffers/packet capture

Snort

TCPdump

Sun Solaris snoop

COLD

Ethereal

Analyzer

Windump

WinPcap

NetPeek

Sniffer Pro

- Vers

Slapper

- Avertissements

<http://www.cisco.com/warp/public/707/cisco-sa->

<http://www.kb.cert.org/vuls/id/658859>

- Scanners

IPv6 security scanner

Halfscan6

Strobe

Netcat

- DoS Tools

6tunneldos

4to6ddos

Imps6-tools

- Packet forgers

SendIP

Packit

Spak6

- Outil complet

<http://www.thc.org/thc-ipv6/>

# Problèmes spécifiques à IPv6



# Manipulation d'entête IPv6

- Un chaînage illimité des entêtes peut rendre le filtrage difficile
- Attaques DoS potentielle pour les implémentations IPv6 non robustes

Plus de conditions de limites à exploiter

Peut saturer les buffers avec beaucoup d'entêtes d'extension?

⊕ Frame 1 (423 bytes on wire, 423 bytes captured)

⊕ Raw packet data

⊕ Internet Protocol Version 6

⊕ Hop-by-hop Option Header

⊕ Destination Option Header

⊕ Routing Header, Type 0

⊕ Hop-by-hop Option Header

⊕ Destination Option Header

⊕ Routing Header, Type 0

⊕ Destination Option Header

⊕ Routing Header, Type 0

⊕ Transmission Control Protocol, Src Port: 1024 (1024), Dst Port: bgp (179), Seq: 0, Ack: 0, Len: 51

⊕ Border Gateway Protocol

**Paquet IPv6 de syntaxe valide selon le sniffer**

**Ne doit apparaître qu'une fois**

**Pas plus de 2 "destination header"**

**Destination Options Header doit être le dernier**

# Le mythe IPsec :

## IPsec de bout en bout est notre sauveur

- IPv6 exige l'implémentation de IPsec
- IPv6 n'a pas besoin IPsec
- Certains pensent que IPsec doit être utilisé pour sécuriser tous les trafics...

Intéressant problème de **scalability** ( $n^2$  avec IPsec)

On doit **faire confiance aux hôtes et aux utilisateurs finaux** parce que le réseau ne peut pas sécuriser le trafic: pas de IPS, ACL, firewall

La **télémetrie réseau est aveugle** : NetFlow ne sert à rien

Les services réseaux obstrués : QoS?

**Recommandation:** Ne pas utiliser IPsec end-to-end dans un domaine administratif

Usage résidentiel probablement recommandé.

# Problèmes de transition IPv4 vers IPv6

- 16+ méthodes, utilisables en combinaison

IP spoofing

- Dual stack

Sécuriser les deux protocoles

Abus croisés v4/v6

Robustesse (ressources partagées)

- Tunnels

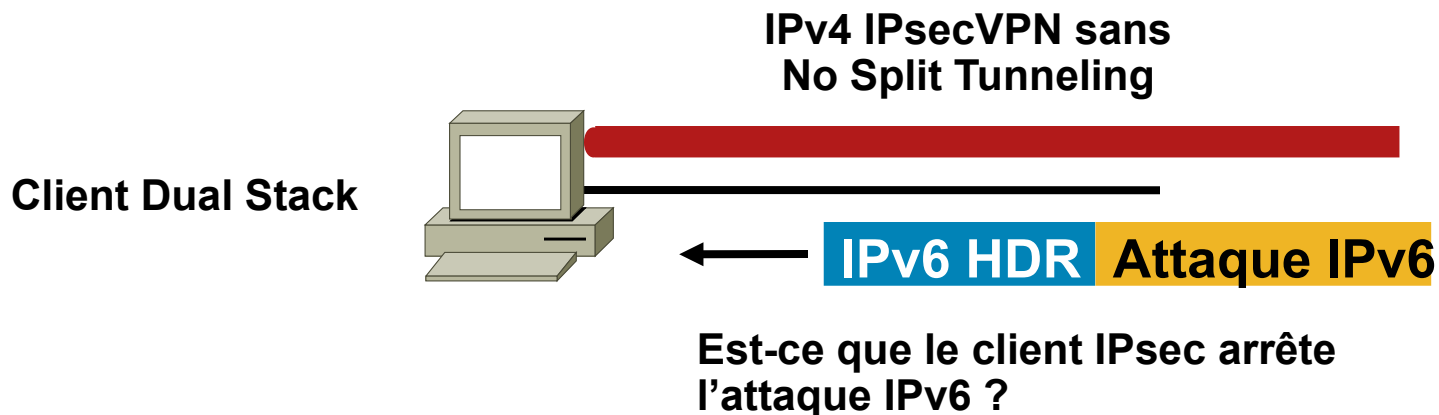
Bypass firewalls (protocole 41)

# Considérations sur les hôtes dual-stack

- Sécurité sur un noeud dual-stack

On peut attaquer les applications sur IPv6 et/ou IPv4

- Les deux versions doivent être sécurisées  
intrusion, firewall personnels, Client VPN, etc.



# Dual Stack avec IPv6 activé par défaut

- Votre noeud:

IPv4 est protégé par votre firewall personnel

IPv6 est activé par défaut (Vista, Linux, MacOS, ...)

- Votre réseau

Ne tourne pas IPv6

- votre hypothèse:

Je suis en sécurité

- Réalité

Vous n'êtes pas en sécurité

Des Router Advertisements sont envoyés par des pirates

Votre hôte prend une adresse IPv6

Vous êtes attaqué par IPv6

- => C'est probablement le moment de configurer IPv6 sur votre réseau

# Bonnes pratiques de sécurité IPv6



# Bonnes pratiques

- Implémenter les privacy extensions avec précaution
- Filtrer les adresses internes à la limite du site
- Filtrer les services inutiles sur le firewall
- Filtrer ICMP avec mesure
- Porter attention à la sécurité des applications et des hôtes
- Déterminer quelle entête ne pas filtrer
- Bloquer les routing header type 0 à l'entrée du réseau
- Déterminer quel messages ICMPv6 sont nécessaires
- Bloquer les fragments destinés aux nœuds intermédiaires quand c'est possible
- S'assurer de ces capacités de filtrage de fragment IPv6 appropriée

# Bonnes pratiques (suite)

- Implémenter des filtres RFC 2827 et demander à votre ISP de faire de même
- Documenter les procédures last-hop traceback
- Utiliser des protections cryptographiques si nécessaires
- Utiliser des entrées statiques dans le Neighbor Cache si nécessaire
- Filtrer les adresses source multicast
- Utiliser IPsec pour sécuriser OSPFv3 et RIPng
- Utiliser des tunnels statiques plutôt que dynamiques
- Laisser passer seulement certains tunnels par votre firewall

# Renforcer les politiques de sécurité



# ACL Cisco IOS IPv6


- Filtrage sur la source et la destination
- Peut filtrer le trafic entrant ou sortant sur une interface spécifique
- Implicite “deny all” à la fin
- Presque comme en IPv4

# ACL Cisco IOS IPv6

## Un exemple simple

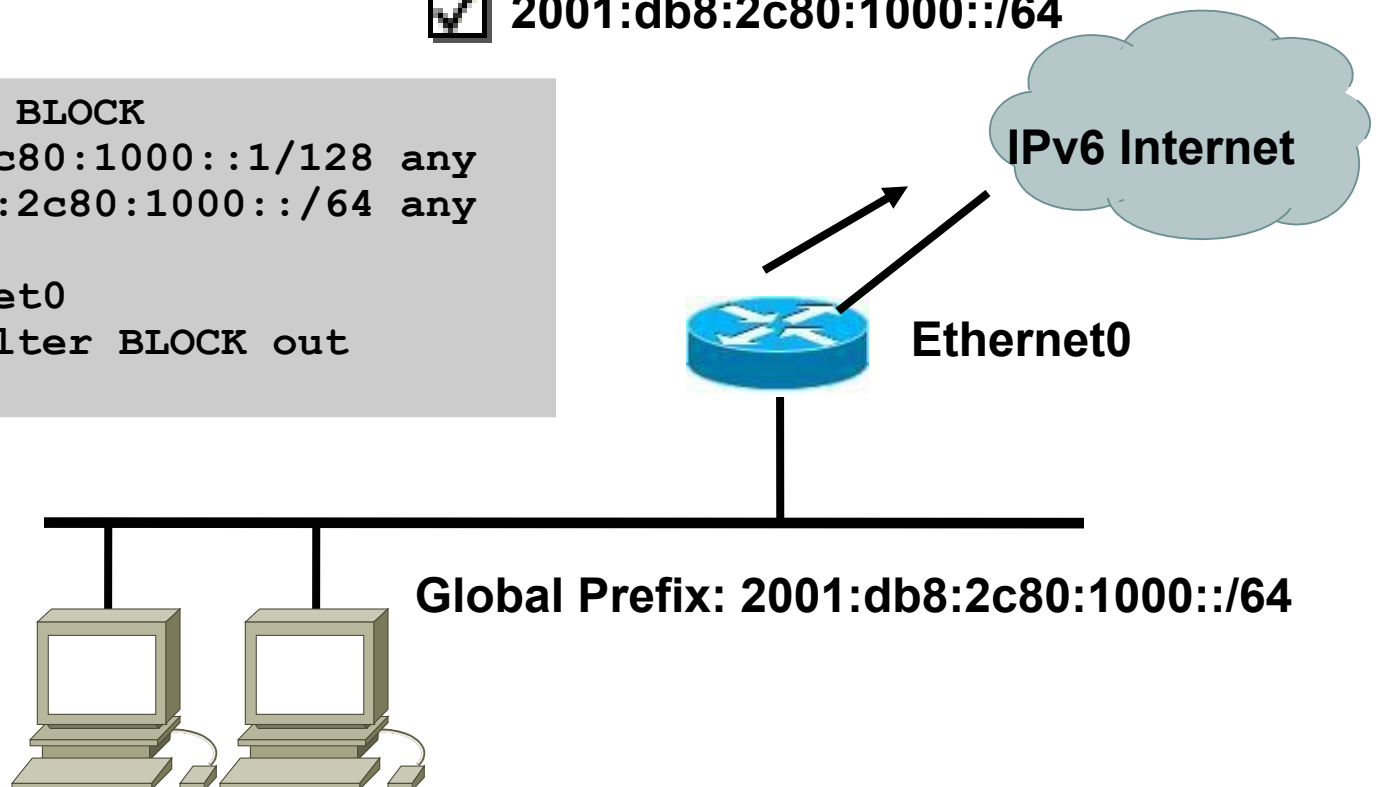
Filtrer le trafic sortant d'une adresse IPv6

 2001:db8:2c80:1000::1

 2001:db8:2c80:1000::/64

```
ipv6 access-list BLOCK
deny 2001:db8:2c80:1000::1/128 any
permit 2001:db8:2c80:1000::/64 any

interface Ethernet0
ipv6 traffic-filter BLOCK out
```



# ACL IPv6 étendues

- Couches supérieures: ICMP, TCP, UDP, SCTP, etc...
- ICMPv6 code and type
- TCP SYN, ACK, FIN, PUSH, URG, RST
- Numéros de ports de niveau 4
- Classe de Trafic (only six bits/8) = DSCP
- Flow label (0-0xFFFFF)
- IPv6 header options

Fragments

Routing header type

Destination header type

# Règles implicites pour ACL IPv6

## Permettre Neighbor Discovery

- Les règles implicites suivantes existent à la fin de chaque ACL IPv6 pour permettre les ICMPv6 neighbor discovery:

```
permit icmp any any nd-na  
permit icmp any any nd-ns  
deny ipv6 any any
```

- **Faites Attention quand vous ajoutez « deny ipv6 any any log » à la fin**

```
permit icmp any any nd-na  
permit icmp any any nd-ns  
deny ipv6 any any log
```

# ACL IPv6 pour protéger les accès virtuels

```
ipv6 access-list VTY
  permit ipv6 2001:db8:0:1::/64 any

line vty 0 4
  ipv6 access-class VTY in
```

# Conclusion



# Key Take Away

- Rien vraiment de nouveau dans IPv6

L'absence d'expérience opérationnelle peut fragiliser la sécurité pendant un moment

- Le renforcement de la sécurité est possible

Contrôlez votre trafic IPv6 comme vous faites pour IPv4

Utiliser IPsec quand c'est possible

- Attention : votre réseau peut être attaqué en utilisant IPv6