# Passer à IPv6

Laurent Toutain - GET/ENST Bretagne

Laurent.Toutain@enst-bretagne.fr

# Agenda

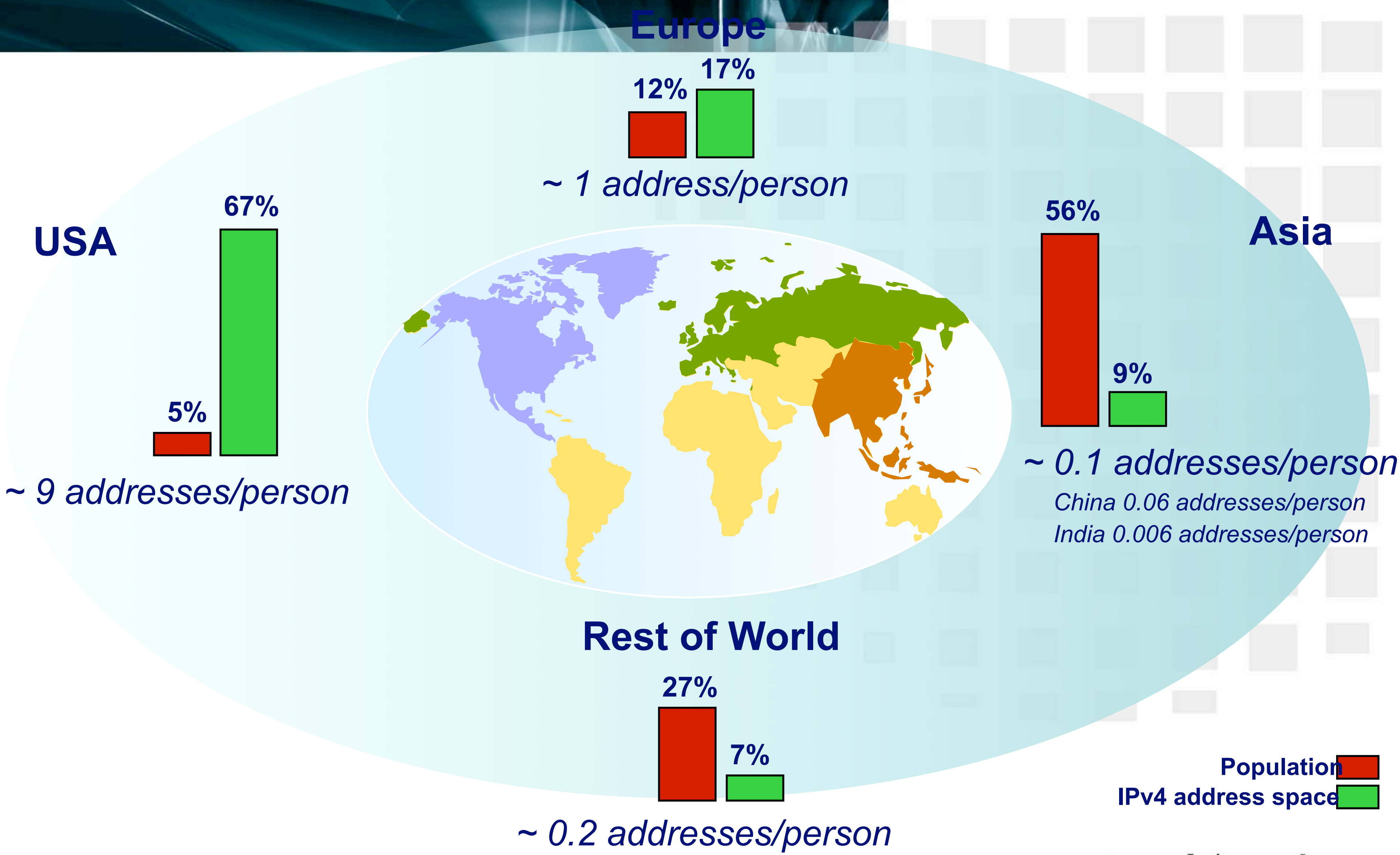
- Les risques liés a IPv4
- Petit tutorial sur IPv6
- Perspective de recherche
- L'ENST Bretagne et IPv6

# l'arrivée des problèmes avec IPv4

# L'Internet actuel

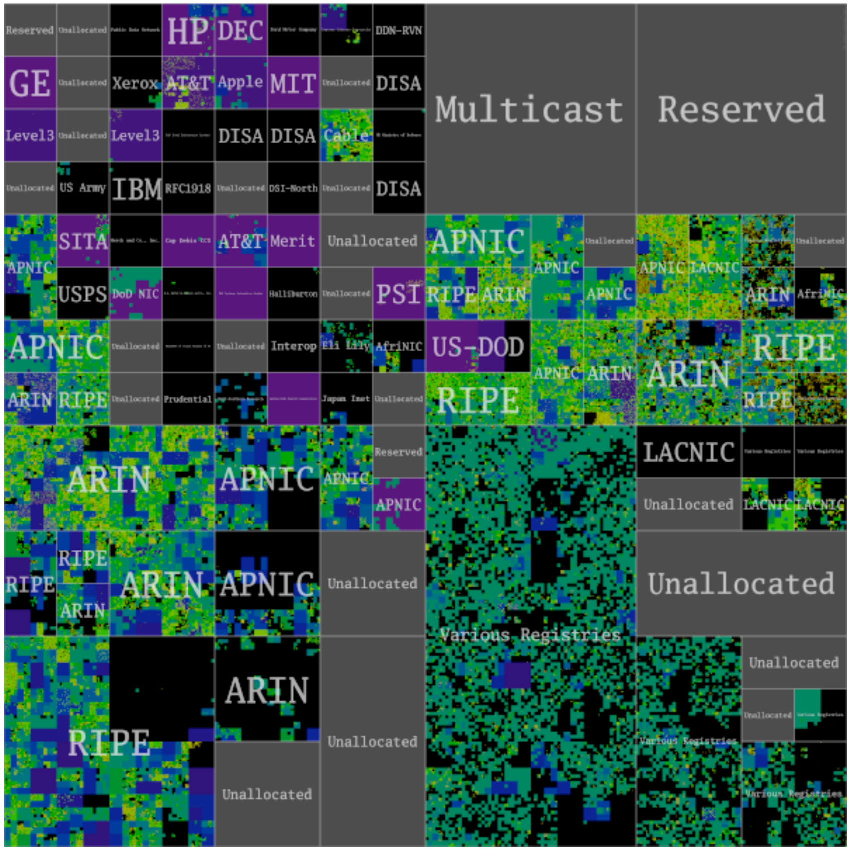
- **Croissance massive**
- **En 1983 environs 100 ordinateurs sur le réseau**
- **Les choix orignaux sont toujours d'actualité :**
  - L'espace d'adressage est saturé à très court terme
  - La mobilité des équipements n'a pas été prise en compte
    - Les ordinateurs pesaient plusieurs tonnes dans les années 1970

# The IPv4 world view



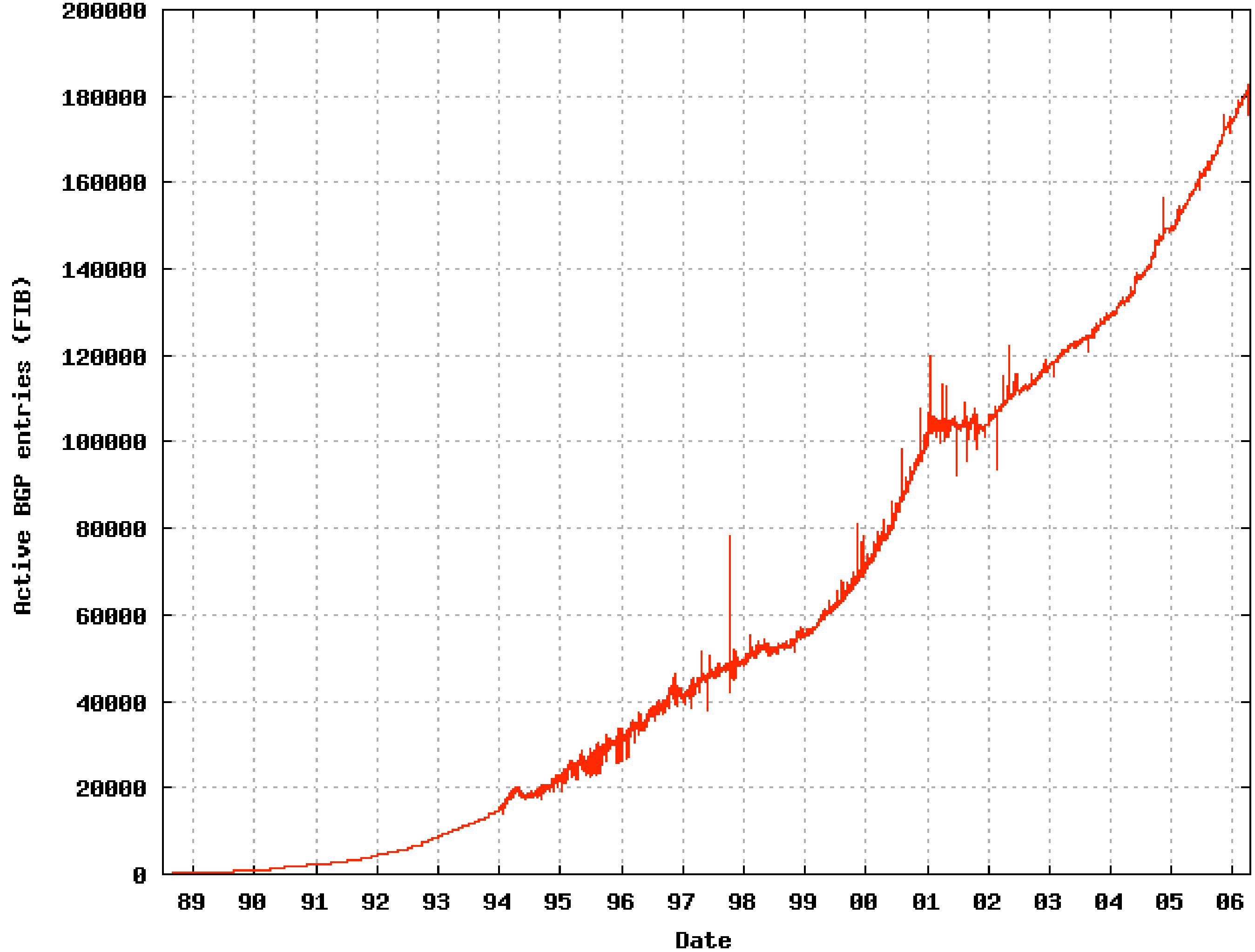

Slide: Commission Européenne Ulf Dahlsten

# Epuisement de l'espace d'adressage IPv4



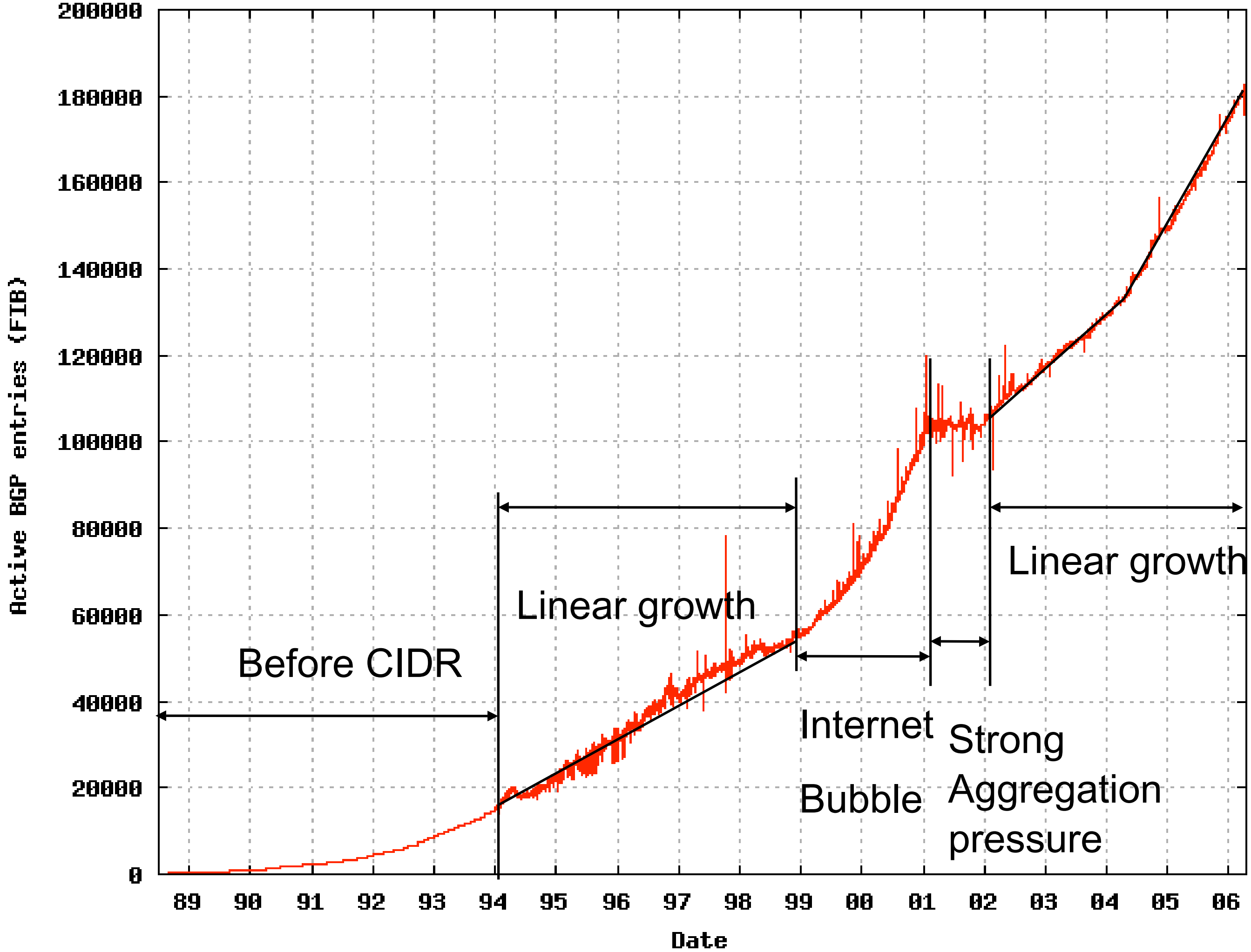

- http://maps.measurement-factory.com/gallery/Routeviews/

# Saturation des tables de routage




Source: Geoff Huston - (`http://bgp.potaroo.net/`)

# Saturation des tables de routage




Source: Geoff Huston - (`http://bgp.potaroo.net/`)

# Risques

- **Le coût d'exploitation de l'Internet v4 va augmenter :**
  - Davantage de niveaux de NAT
  - Davantage de mémoire dans les routeurs
- **La qualité de l'Internet v4 va diminuer :**
  - Temps de convergence des tables de routage
  - Agrégations sauvages
  - Suppression des préfixes longs
- **Année pivot : 2011**

# Avant 2011

- **Depuis 5 ans:**
  - Ecrire du code compatible IPv4 et IPv6
    - Possible en C, Java, python, ...
    - Attention aux libraries incompatibles
- **Maintenant:**
  - Préciser dans les appels d'offre la compatibilité à IPv6
    - Routeur, switchs, firewall, serveurs
  - Démarrer des plates-formes de test IPv6
    - Familiarisation avec la nouvelle version du protocole
    - Tests de compatibilité des équipements

# En 2011

- **Offrir une vision vers l'extérieur en double pile**
  - Permet de contrer les problèmes d'interconnectivité d'IPv4
  - Intégrer des adresses AAAA dans le DNS
  - mail (MX), web
    - Solutions simples avec proxys SSL et HTTP
    - Connectivité IPv6 par tunnel ou native
- **Le back-office reste v4**
- **Progressivement déployer IPv6 dans l'entreprise**

# IPv6 Ready Logo

# Incert IPv6 Courses

ENST bretagne

November 13, 2007

# IPv6 addresses



```
F2C:544:9E::2:EF8D:6B7 F692:: A:1455::A:6E0 D:63:D::4:3A:55F B33:C::F2 7:5059:3D:C0::
9D::9BAC:B8CA:893F:80 1E:DE2:4C83::4E:39:F35:C875 2:: A:FDE3:76:B4F:D9D:: D6::
369F:9:F8:DBF::2 DD4:B45:1:C42F:BE6:75:: 9D7B:7184:EF::3FB:BF1A:D80 FE9::B:3
EC:DB4:B:F:F11::E9:090 83:B9:08:B5:F:3F:AF:B84 E::35B:8572:7A3:FB2 99:F:9:8B76::BC9
D64:07:F394::BDB:DF40:08EE:A79E AC:23:5D:78::233:84:8 F0D:F::F4EB:0F:5C7
E71:F577:ED:E:9DE8:: B::3 1D3F:A0AA:: 70:8EA1::8:D5:81:2:F302 26::8880:7 93:: F::9:0
E:2:0:266B:: 763E:C:2E:1EB:F6:F4:14:16 E6:6:F4:B6:A888:979E:D78:09
9:754:5:90:0A78:A1A3:1:7 2:8:: 97B:C4::C36 A40:7:5:7E8F:0:32EC:9A:D0 8A52::575
D::4CB4:E:2BF:5485:8CE 07:5::41 6B::A9:C 94FF:7B8::D9:51:26F 2::E:AE:ED:81 8241:: 5F97::
AD5B:259C:7DB8:24:58:552A:: 94:4:9FD:4:87E5:: 5A8:2FF:1::CC EA:8904:7C::
7C::D6B7:A7:B0:8B DC:6C::34:89 6C:1::5 7B3:6780:4:B1::E586 412:2:5E1:6DE5:5E3A:553:3::
7F0:: B39::1:B77:DB 9D3:1F1:4B:3:B4E6:7681:09:D4A8 61:520::E0 1:28E9:0:095:DF:F2::
1B61:4::1DE:50A 34BC:99::E9:9EFB E:EF:: BDC:672A:F4C8:A1::4:7:9CB7 C697:56AD:40:8:0::62
```

# Don't Worry



Addresses are not random numbers, ... they are quite easy to remember and manipulate

# Notation

- Base format (a 16 byte Global IPv6 Address):
  - `2001:0660:3003:0001:0000:0000:6543:210F`
- Compact Format:

2001:0660:3003:0001:0000:0000:6543:210F

1. remove 0 on the left of each word
2. substitute one sequence of zeros by `::`

- an IPv4 address may also appear : `::FFFF:123.12.34.56`

**Warning:**

`2001:660:3::/40` is in fact `2001:660:0003::/40` and not `2001:660:0300::/40`

# Notation



- Base format (a 16 byte Global IPv6 Address):
  - `2001:0660:3003:0001:0000:0000:6543:210F`
- Compact Format:

2001:0660:3003:0001:0000:0000:6543:210F

1. remove 0 on the left of each word
2. substitute one sequence of zeros by `::`

- an IPv4 address may also appear : `::FFFF:123.12.34.56`

**Warning:**

`2001:660:3::/40` is in fact `2001:660:0003::/40` and not `2001:660:0300::/40`

# Notation

- Base format (a 16 byte Global IPv6 Address):
  - `2001:0660:3003:0001:0000:0000:6543:210F`
- Compact Format:

**2001:660:3003:1:0:0:6543:210F**

1. remove 0 on the left of each word
2. substitute one sequence of zeros by `::`

- an IPv4 address may also appear : `::FFFF:123.12.34.56`

**Warning:**

`2001:660:3::/40` is in fact `2001:660:0003::/40` and not `2001:660:0300::/40`

# Notation



- Base format (a 16 byte Global IPv6 Address):
  - `2001:0660:3003:0001:0000:0000:6543:210F`
- Compact Format:

2001:660:3003:1:0:0:6543:210F

1. remove 0 on the left of each word
2. substitute one sequence of zeros by `::`

- an IPv4 address may also appear : `::FFFF:123.12.34.56`

**Warning:**

`2001:660:3::/40` is in fact `2001:660:0003::/40` and not `2001:660:0300::/40`

# Notation



- Base format (a 16 byte Global IPv6 Address):
  - `2001:0660:3003:0001:0000:0000:6543:210F`
- Compact Format:

2001:660:3003:1::6543:210F

1. remove 0 on the left of each word
2. substitute one sequence of zeros by `::`

- an IPv4 address may also appear : `::FFFF:123.12.34.56`

**Warning:**

`2001:660:3::/40` is in fact `2001:660:0003::/40` and not `2001:660:0300::/40`

# Is it enough for the future ?



- Address length
  - Between 1 564 and 3 911 873 538 269 506 102 addresses by $m^2$
  - 60 000 trillion trillion addresses per inhabitant of the earth
  - Addresses for every grain of sands in the world
- Justification of a fix address length

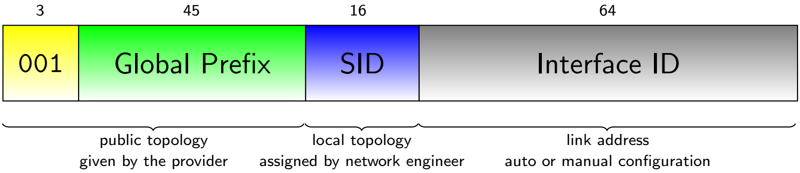
**Warning:**

- An address for everything on the network and not an address for everything
- No addresses for whole life:
  - Depend of your position on the network
  - ISP Renumbering may be possible

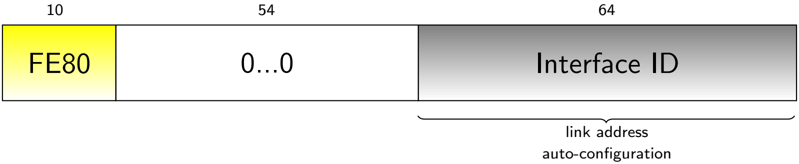
# Address Format

# Address Format

Global Unicast Address:




Link-Local Address:



# SID Values



- 16 bit length up to 65 535 subnets
  - Large enough for most companies
  - Too large for home network ?
  - May be an /56 or /60 GP will be allocated
- There is no strict rules to structure SID:
  - sequencial : 1, 2, ...
  - use VLAN number
  - include usage to allow filtering, for instance, Rennes 1 University:

| **4bits** : Community | **8bits** | **4bits** |
|---|---|---|
| **0** : Infrastructure | *Specific addresses* | |
| **1** : Tests | *Specific addresses* | |
| **6** : Point6 | *Managed by Point6* | |
| **8** : Wifi guests | *Specific addresses* | |
| **A** : Employees | Entity | Sub-Network |
| **E** : Students | Entity | Sub-Network |
| **F** : Other (Start up, etc.) | *Specific addresses* | |

# Interface Identifier

**Interface ID can be selected differently**

- Derived from a Layer 2 ID (I.e. MAC address) :
  - for Link Local address
  - for Global Address : plug-and-play hosts
- Assigned manually :
  - to keep same address when Ethernet card or host is changed
  - to remember easily the address
    - 1, 2, 3, ...
    - last digit of the v4 address
    - the IPv4 address (for nostalgic system administrators)
    - ...

# Interface Identifier



**Interface ID can be selected differently**

- Random value :
  - Changed every day to guaranty anonymity
- Hash of other values (experimental) :
  - To link address to other properties
  - Public key
  - List of assigned prefixes
  - ...

# Example : Mac / Unix



```
%ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        inet 127.0.0.1 netmask 0xff000000
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500

        inet6 fe80::216:cbff:febe:16b3%en1 prefixlen 64 scopeid 0x5

        inet 192.168.2.5 netmask 0xffffff00 broadcast 192.168.2.255
        inet6 2001:660:7307:6031:216:cbff:febe:16b3 prefixlen 64
        autoconf

        ether 00:16:cb:be:16:b3
        media: autoselect status: active
        supported media: autoselect
```

# IPv6 Header : Simpler



## Definition

- IPv6 header follows the same IPv4 principle:
  - fix address size ... but 4 times larger
  - alignment on 64 bit words (instead of 32)
- Functionalities never used in IPv4 are supressed

## Goal :

- Forward packet as fast as possible
- Less treatments in routers
- More functionalities at both ends

# IPv4 Header



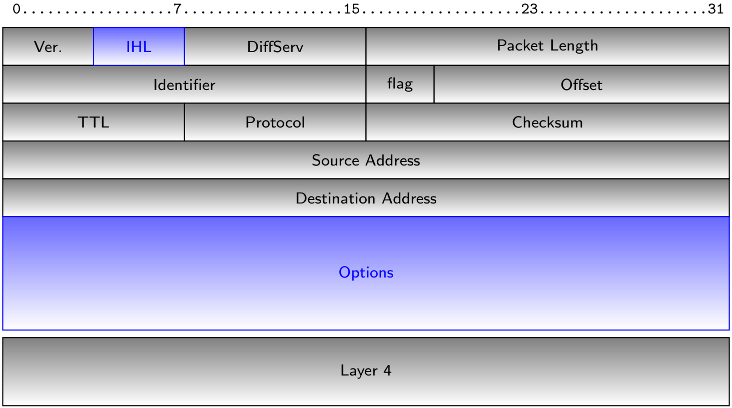
0 . . . . . . . 7 . . . . . . . 15 . . . . . . . 23 . . . . . . . 31

| | | | |
|---|---|---|---|
| Ver. | IHL | DiffServ | Packet Length |
| Identifier | | flag | Offset |
| TTL | | Protocol | Checksum |
| Source Address | | | |
| Destination Address | | | |
| Options | | | |

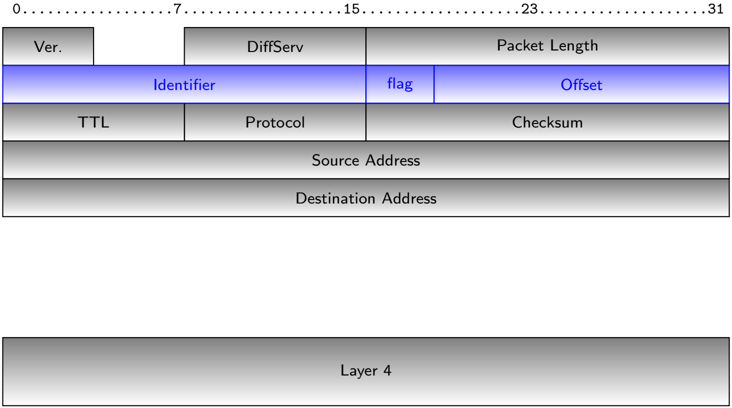
| |
|---|
| Layer 4 |

# IPv4 Header



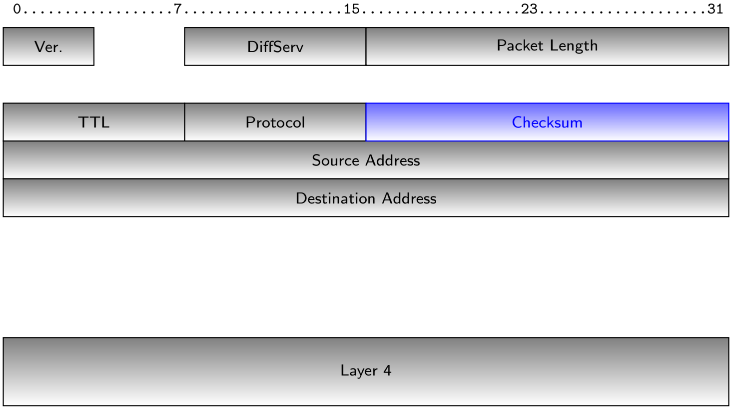

0.................7.................15.................23.................31
Ver.
IHL
DiffServ
Packet Length
Identifier
flag
Offset
TTL
Protocol
Checksum
Source Address
Destination Address
Options
Layer 4

# IPv4 Header


0 7 15 23 31
Ver.
DiffServ
Packet Length
Identifier
flag
Offset
TTL
Protocol
Checksum
Source Address
Destination Address
Layer 4

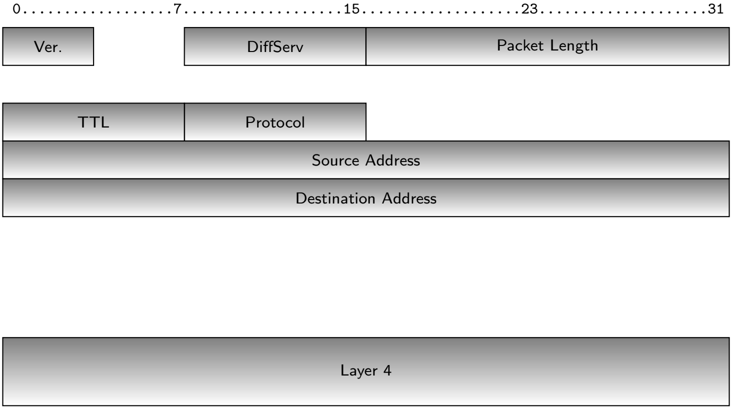
# IPv4 Header




0 7 15 23 31
Ver.
DiffServ
Packet Length
TTL
Protocol
Checksum
Source Address
Destination Address
Layer 4

# IPv4 Header

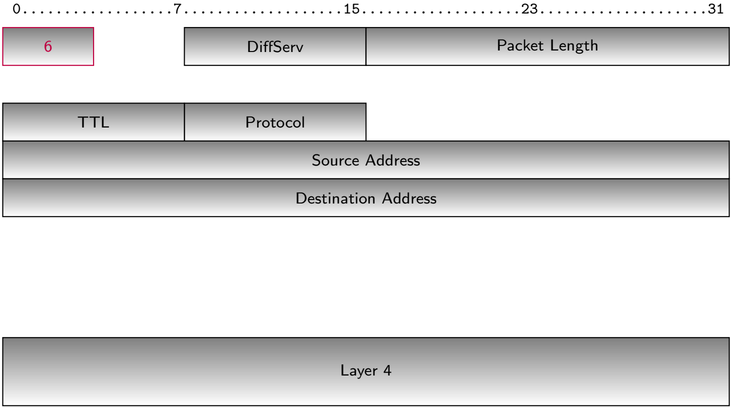




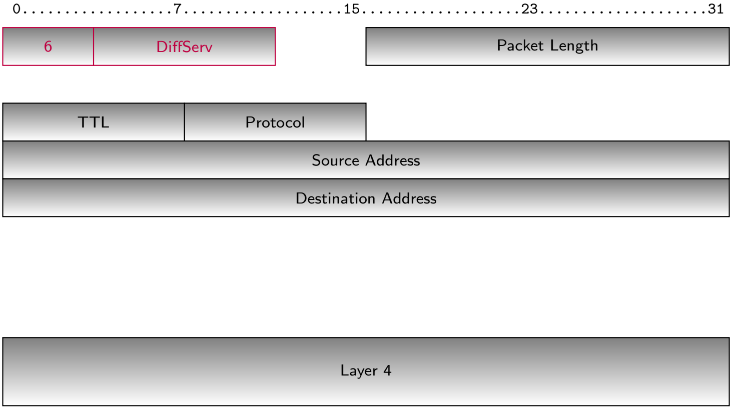
0.................7..................15..................23..................31
Ver.
DiffServ
Packet Length
TTL
Protocol
Source Address
Destination Address
Layer 4

# IPv4 Header



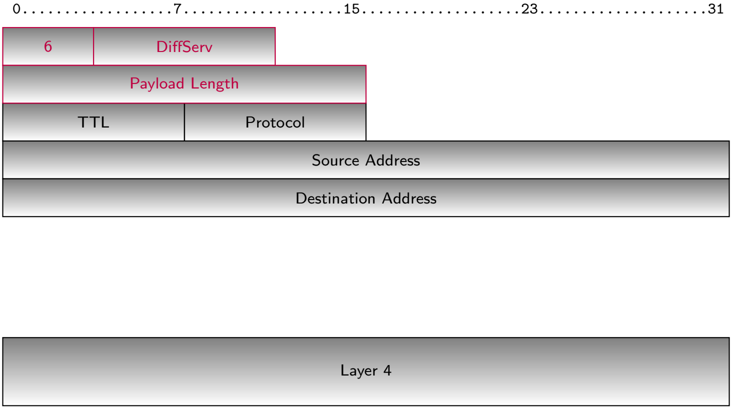

0.................7..................15..................23..................31
6
DiffServ
Packet Length
TTL
Protocol
Source Address
Destination Address
Layer 4

# IPv4 Header


0 . . . 7 . . . 15 . . . 23 . . . 31
6
DiffServ
Packet Length
TTL
Protocol
Source Address
Destination Address
Layer 4

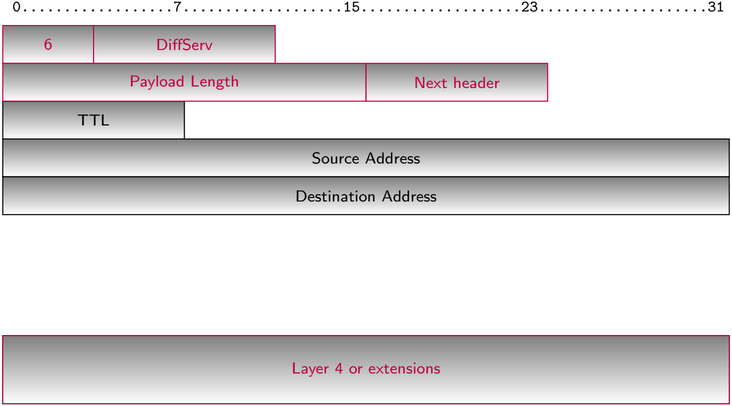
# IPv4 Header




0 . . . 7 . . . 15 . . . 23 . . . 31
6
DiffServ
Payload Length
TTL
Protocol
Source Address
Destination Address
Layer 4

# IPv4 Header

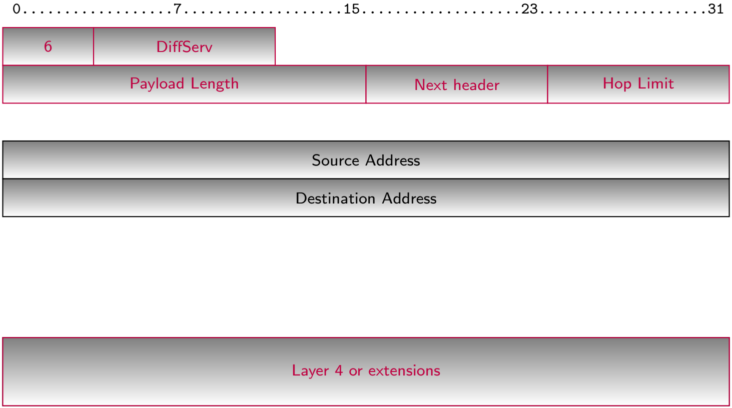




0.................7..................15..................23..................31
6
DiffServ
Payload Length
Next header
TTL
Source Address
Destination Address
Layer 4 or extensions

# IPv4 Header

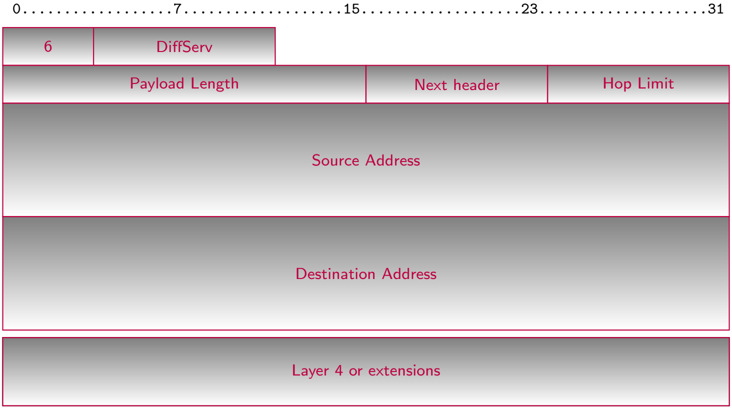




0 . . . . . . . 7 . . . . . . . 15 . . . . . . . 23 . . . . . . . 31
6
DiffServ
Payload Length
Next header
Hop Limit
Source Address
Destination Address
Layer 4 or extensions

# IPv4 Header




0..................7..................15..................23..................31
6
DiffServ
Payload Length
Next header
Hop Limit
Source Address
Destination Address
Layer 4 or extensions

# IPv6 Header



0 . . . . . . . . . . . . . . . . . 7 . . . . . . . . . . . . . . . . . . 15 . . . . . . . . . . . . . . . . . . 23 . . . . . . . . . . . . . . . . . . . 31

| | | | | |
|---|---|---|---|---|
| 6 | DiffServ | Flow Label | | |
| Payload Length | | | Next header | Hop Limit |
| Source Address | | | | |
| Destination Address | | | | |

| |
|---|
| Layer 4 or extensions |

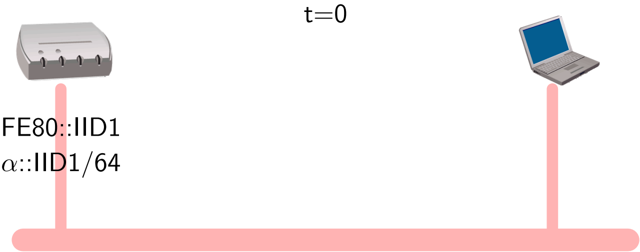
# Stateless Auto-configuration

# Stateless Auto-configuration: Basic Principles



# Stateless Auto-configuration: Basic Principles



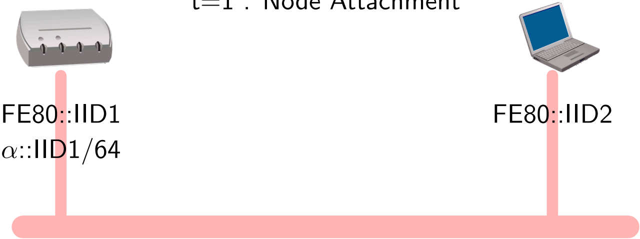



Time t=0: Router is configured with a link-local address and manually configured with a global address ($\alpha$::/64 is given by the network manager)

# Stateless Auto-configuration: Basic Principles



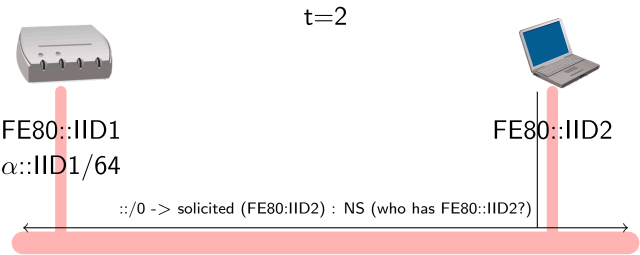
t=1 : Node Attachment




Host constructs its link-local address based on the interface MAC address

# Stateless Auto-configuration: Basic Principles

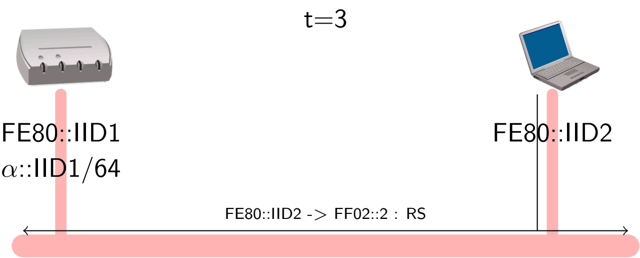





Host does a DAD (i.e. sends a Neighbor Solicitation to query resolution of its own address: no answers means no other host as this value).

# Stateless Auto-configuration: Basic Principles

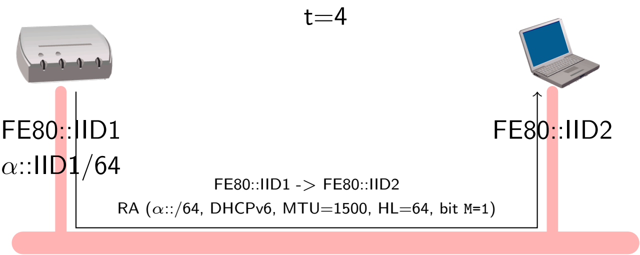





Host sends a Router Solicitation to the All Router Multicast group using the newly link-local configured address.

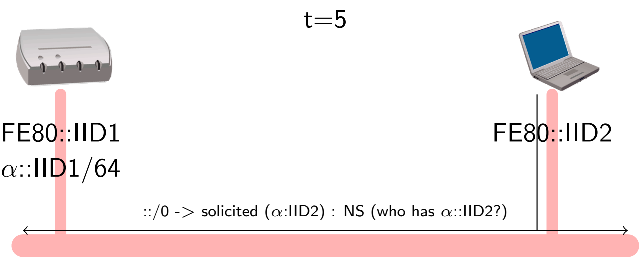
# Stateless Auto-configuration: Basic Principles






Router answer directly to the host using Link-local addresses. The answer may contain a/several prefix(es). Router can also mandate hosts to use DHCPv6 to obtain prefixes (state full auto-configuration) and/or other parameters (DNS servers,...): Bit $\mathtt{M} = 1$.

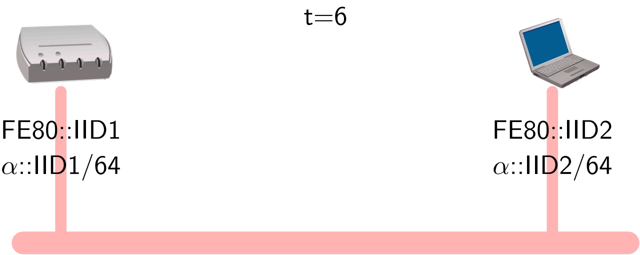
# Stateless Auto-configuration: Basic Principles




Host does a DAD (i.e. sends a Neighbor Solicitation to query resolution of its own global address: no answers means no other host as this value).

# Stateless Auto-configuration: Basic Principles






Host set the global address and takes answering router as the default router.

# Softwires



IETF solution to transport and manage IPvX over IPvY.

Two scenarios discussed in the problem statement :

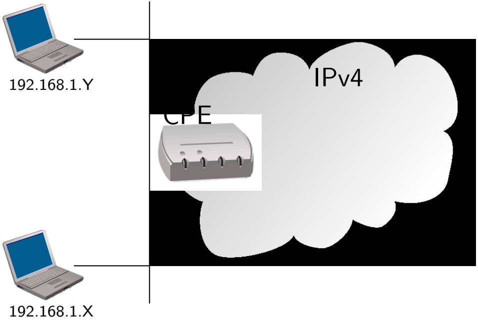
- Mesh problem : IPv4 over IPv6 in core network
  - Problem raised by chinese research network CERNET2
  - Connect IPv4 island across an IPv6 only backbone
  - Solution: MPLS tunnels
- Hub-and-spoke problem : IPv6 over IPv4 for Home Network
  - Problem raised by NTT, Comcast and Point6
  - Connect IPv6 Home network over IPv4 only DSL connection
  - Solution: L2TP tunnels

**Currently deployed by RENATER and Point6**

`http://point6.net/box/`

# Softwires: H&S Architecture



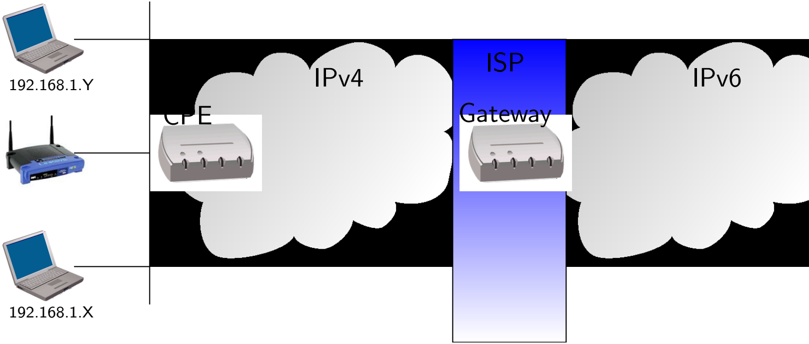



Current DSL ISPs connect Home Network with 1 IPv4 address:

- Clients are behind a NAT Box
- Services hard to deploy at home

# Softwires: H&S Architecture



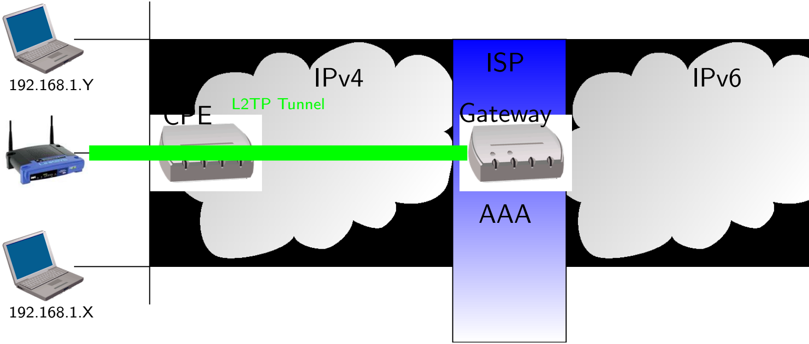



Idea: Build a virtual ISP for IPv6 :

- Provide clients with a non-intrusive CPE box for IPv6
- Deploy a Gateway to connect with IPv6 network

# Softwires: H&S Architecture

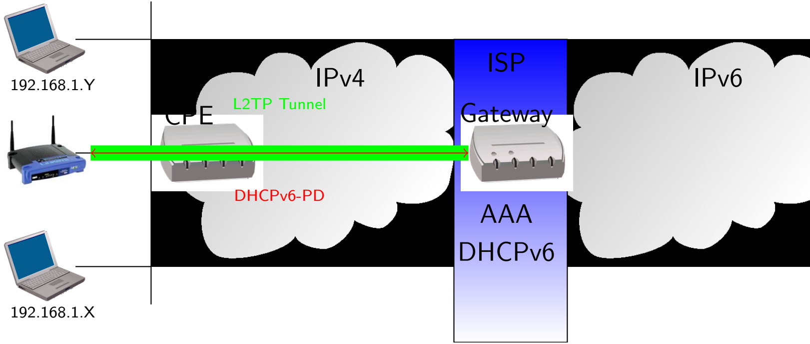





Tunneling with L2TP protocol :

- UDP encapsulation for NAT-traversal
- PPP connection for user authentication using AAA

# Softwires: H&S Architecture

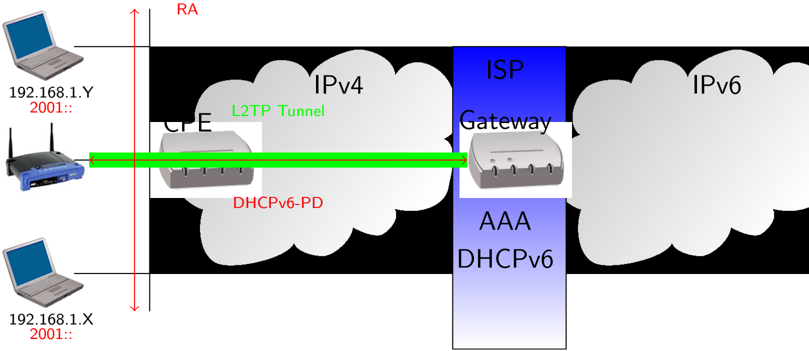



IPv6 prefix for Home Network provided by DHCPv6

- Standard prefix delegation
- Link with AAA for prefix management

# Softwires: H&S Architecture



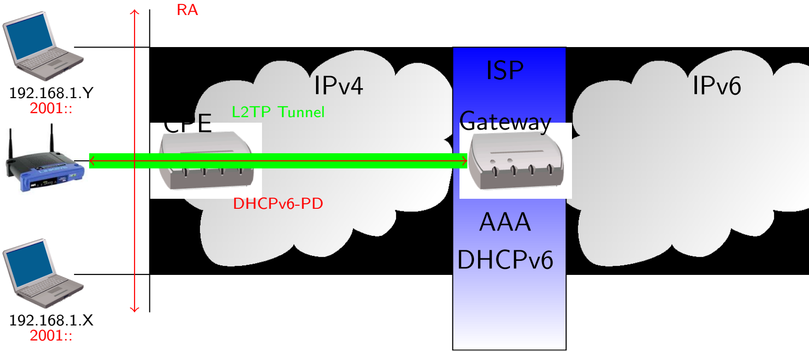



IPv6 addresses distributed with auto-configuration

- Softwire box is the IPv6 default router for the Home Network
- Non-intrusive router

# Softwires: H&S Architecture






Transition Plan

- Softwire box features to be merged with IPv4 CPE
- Virtual ISP features to be moved into official ISP
- Tunnel to be replaced by native connection

# ENST Bretagne et IPv6

- **longue longue longue histoire :**
  - **Le 30 mars 1995:**

```
ottawa#          pwd
/usr/src/usr.bin/telnet6
ottawa#          ./telnet ::204.123.39.2
Trying 0:0:0:0:0:0:cc7b:2702...
Connected to ::204.123.39.2.
Escape character is '^]'.



OSF/1 (sipper.pa-x.dec.com) (ttyp5)


login: telnet
Password:
Last login: Tue Mar 30 03:44:10 from ::128.93.1.21
```

# Formation IPv6

- **Inclus par défaut dans la formation des ingénieurs**
- **Formation continue**
  - Stage: construire des réseaux IPv6
  - Stage: programmer des applications pour IPv6
- **PRACOM**
  - Showroom: IPv6
- **Membre du G6, Task Force IPv6, IPv6 Forum**
- **Point6box: accès IPv6**

# Goals for IPv6 showroom

- Demonstrate IPv6 integration in SME network (front-office, back-office)
- Test and experiment integration of systems and software in IPv6 environnement
- Validate inter-operability for protocols and services
- Demonstrate integration scenarios for services and clients

# ·Architecture

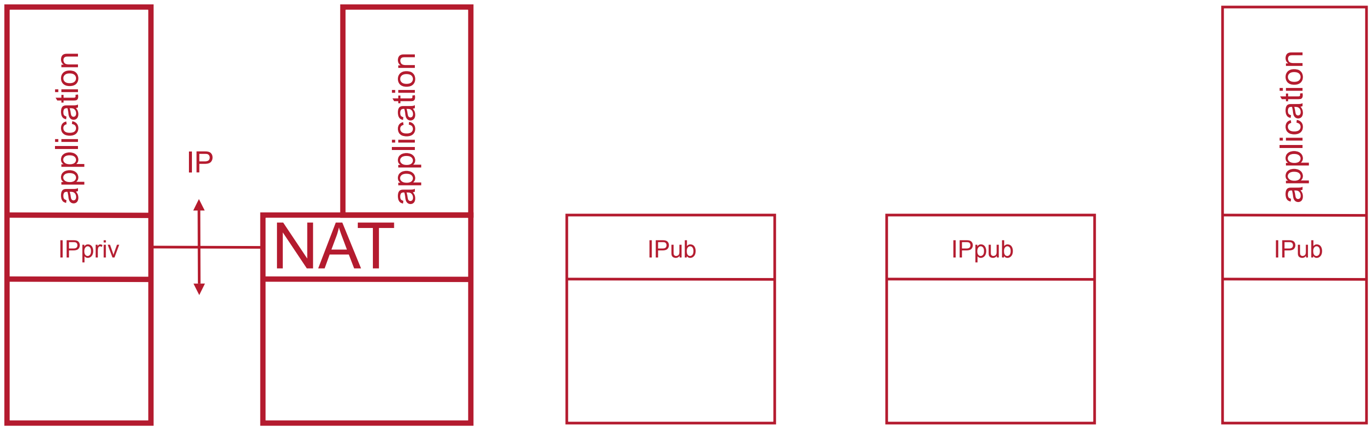
## Differents networks with differents levels of IPv6 integration :

- IPv4-only network
- Dual-Stack (IPv4+IPv6) network
- IPv6-only network

## Differents systems

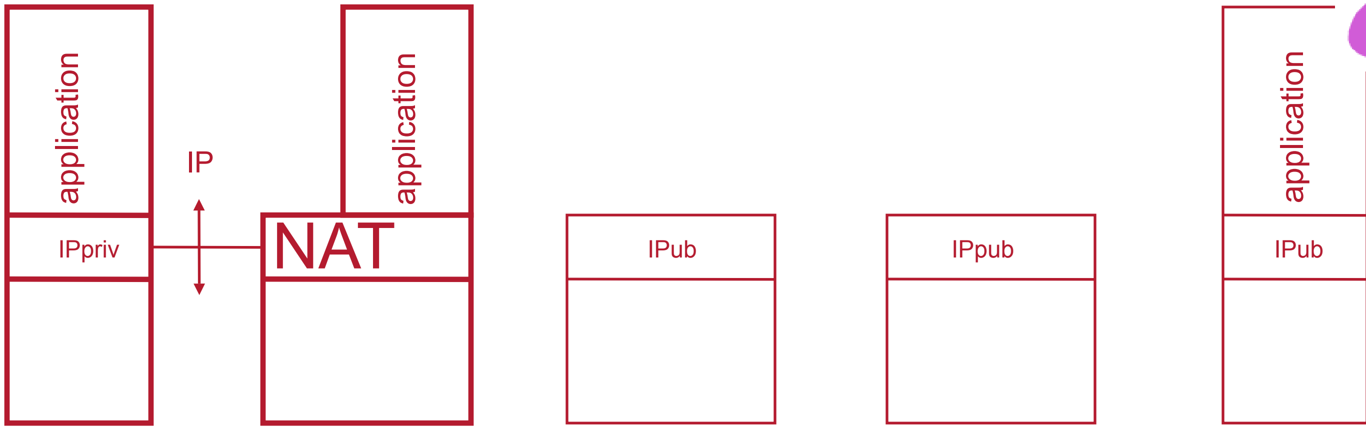
- Clients: Linux, Windows XP and Vista, MacOSX
- Servers: Linux
- Routers/Firewall: Linux, Cisco

# Triple play architecture



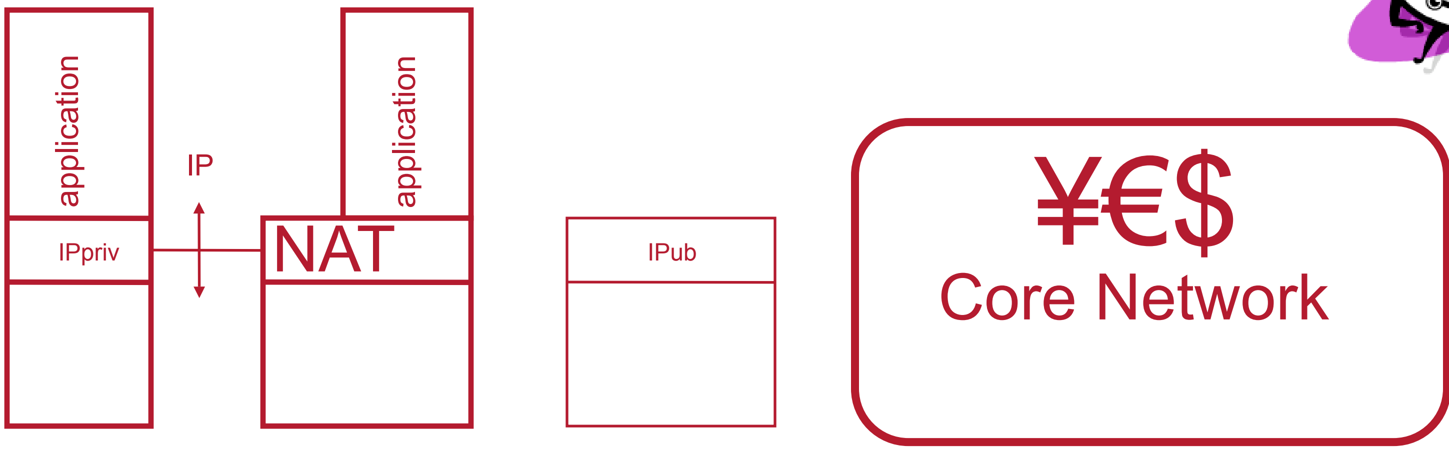

- **Provider services have a public address**
  - They can be managed directly
- **User is behind a NAT so:**
  - He cannot be joined directly
  - He does not know the public address
  - Security feeling
- **Is NAT the provider way to impose its own value added services and block the others ?**

# Adding IPv6

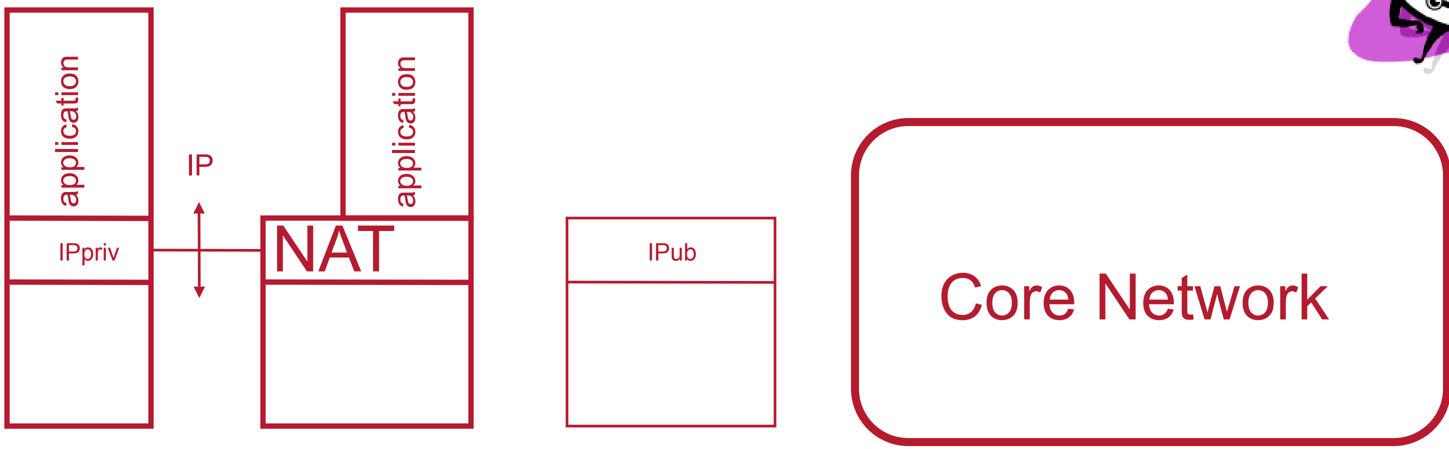

- **IPv4 and IPv6 prefixes are managed the same way**
- **Adapt equipment to IPv6 (routing protocol and forwarding plan)**
  - If not possible with core network elements : use MPLS or 6PE
- **We already have some IPv6 core networks**

# Adding IPv6


- **IPv4 and IPv6 prefixes are managed the same way**
- **Adapt equipment to IPv6 (routing protocol and forwarding plan)**
  - If not possible with core network elements : use MPLS or 6PE
- **We already have some IPv6 core networks**

# Adding IPv6

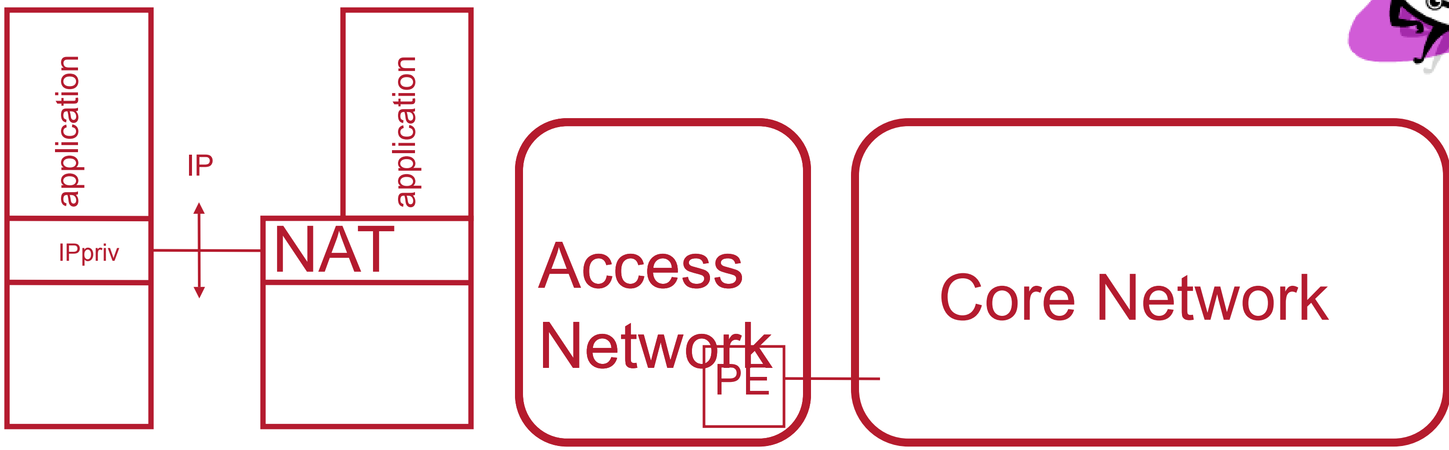




- **V6fication can be a question of investment**
- **But last mile syndrome… may stay IPv4 until new IPv6 based services are developed in home network.**
- **Transition is possible**
  - IETF's Softwires working group

# Adding IPv6


- **V6fication can be a question of investment**
- **But last mile syndrome… may stay IPv4 until new IPv6 based services are developed in home network.**
- **Transition is possible**
  - IETF's Softwires working group

# Adding IPv6


- **V6fication can be a question of investment**
- **But last mile syndrome… may stay IPv4 until new IPv6 based services are developed in home network.**
- **Transition is possible**
  - IETF's Softwires working group

# Adding IPv6



- **V6fication can be a question of investment**
- **But last mile syndrome… may stay IPv4 until new IPv6 based services are developed in home network.**
- **Transition is possible**
  - IETF’s Softwires working group

# Softwires' tunnels



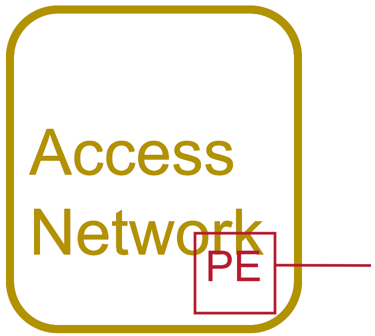
- **During first phase : L2TP**
  - L2TP uses UDP => NAT Traversal
  - PPP is encapsulated in L2TP :
    - User authentication
    - Keep alive messages to maintain NAT contexts
    - Link Local addresses configuration
- **Study prefix delegation**
  - Interaction with DHCPv6 PD
  - Interaction with AAA

# Softwires' tunnels


Access Network

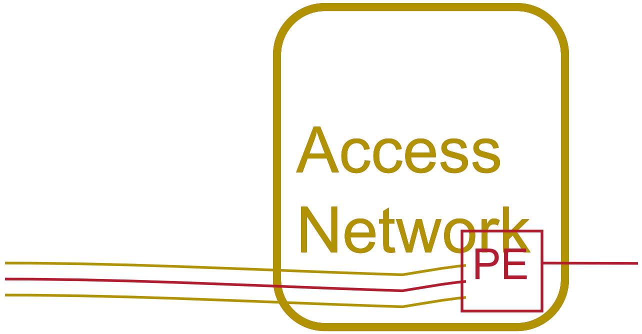
- **During first phase : L2TP**
  - L2TP uses UDP => NAT Traversal
  - PPP is encapsulated in L2TP :
    - User authentication
    - Keep alive messages to maintain NAT contexts
    - Link Local addresses configuration
- **Study prefix delegation**
  - Interaction with DHCPv6 PD
  - Interaction with AAA

# Softwires' tunnels


- **During first phase : L2TP**
  - L2TP uses UDP => NAT Traversal
  - PPP is encapsulated in L2TP :
    - User authentication
    - Keep alive messages to maintain NAT contexts
    - Link Local addresses configuration
- **Study prefix delegation**
  - Interaction with DHCPv6 PD
  - Interaction with AAA

# Softwires' tunnels


- **During first phase : L2TP**
  - L2TP uses UDP => NAT Traversal
  - PPP is encapsulated in L2TP :
    - User authentication
    - Keep alive messages to maintain NAT contexts
    - Link Local addresses configuration
- **Study prefix delegation**
  - Interaction with DHCPv6 PD
  - Interaction with AAA

# Softwires' tunnels



- **Three possibilities in Home Network :**
  - CPE on hosts: One IPv6 address per hosts
  - CPE on special devices :
    - Prefiguration of IPv6 service : always-on, not computer centric
    - Point6box experimentation
  - CPE on Home Gateway
    - Last step before dual stack Access Network
- **Challenge :**
  - Low cost CPE
  - PE architecture

# Softwires' tunnels

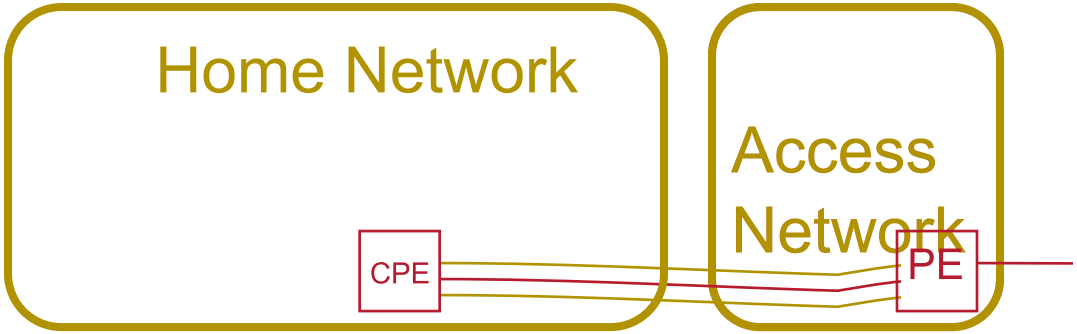


- **Three possibilities in Home Network :**
  - CPE on hosts: One IPv6 address per hosts
  - CPE on special devices :
    - Prefiguration of IPv6 service : always-on, not computer centric
    - Point6box experimentation
  - CPE on Home Gateway
    - Last step before dual stack Access Network
- **Challenge :**
  - Low cost CPE
  - PE architecture


# Softwires' tunnels

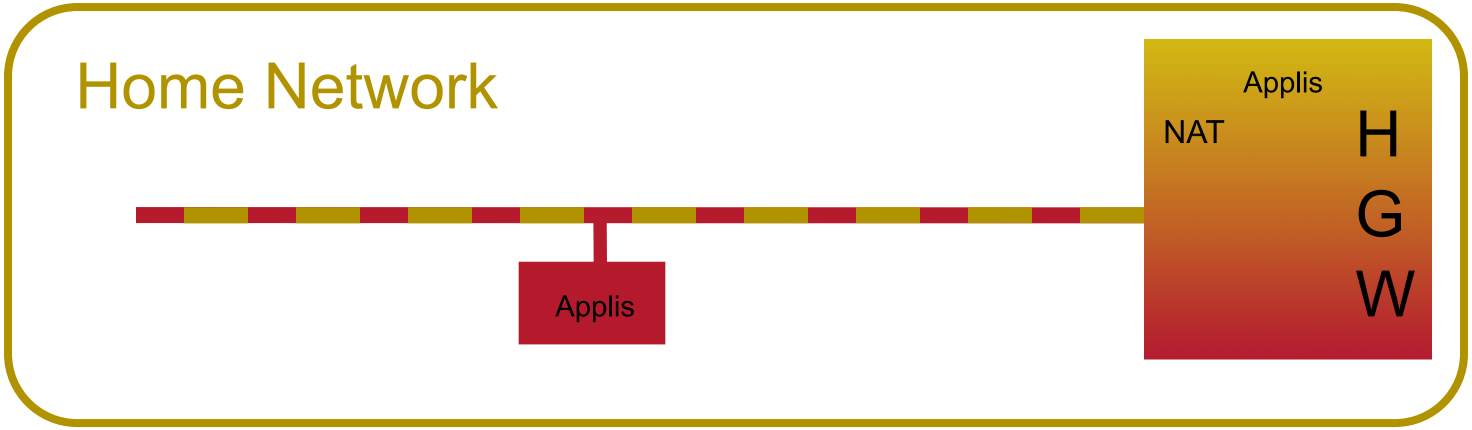

- **Three possibilities in Home Network :**
  - CPE on hosts: One IPv6 address per hosts
  - CPE on special devices :
    - Prefiguration of IPv6 service : always-on, not computer centric
    - Point6box experimentation
  - CPE on Home Gateway
    - Last step before dual stack Access Network
- **Challenge :**
  - Low cost CPE
  - PE architecture

# Softwires' tunnels

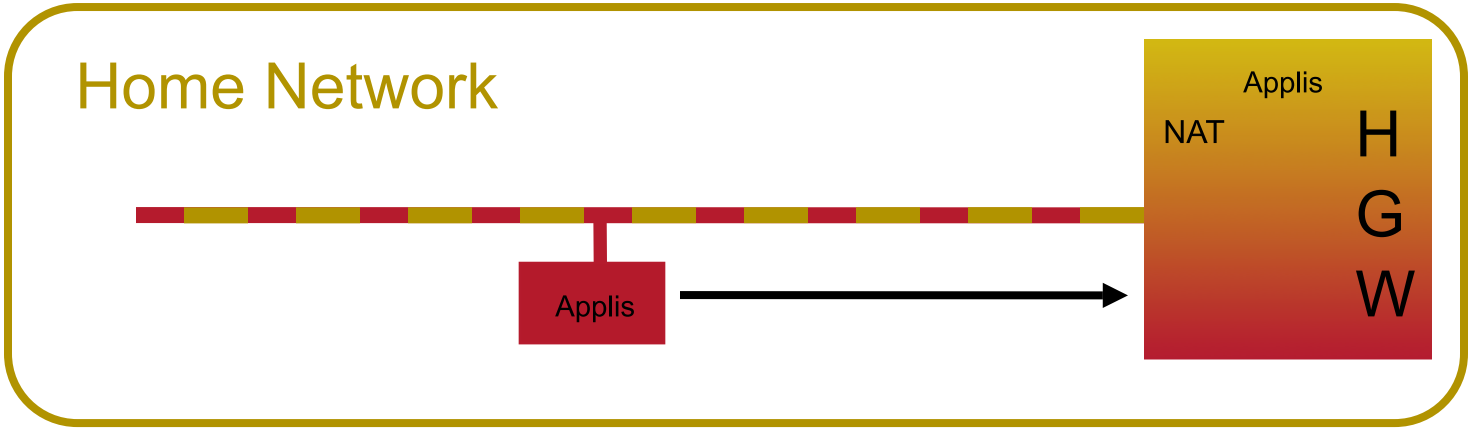

- **Three possibilities in Home Network :**
  - CPE on hosts: One IPv6 address per hosts
  - CPE on special devices :
    - Prefiguration of IPv6 service : always-on, not computer centric
    - Point6box experimentation
  - CPE on Home Gateway
    - Last step before dual stack Access Network
- **Challenge :**
  - Low cost CPE
  - PE architecture

# Home Network Architecture



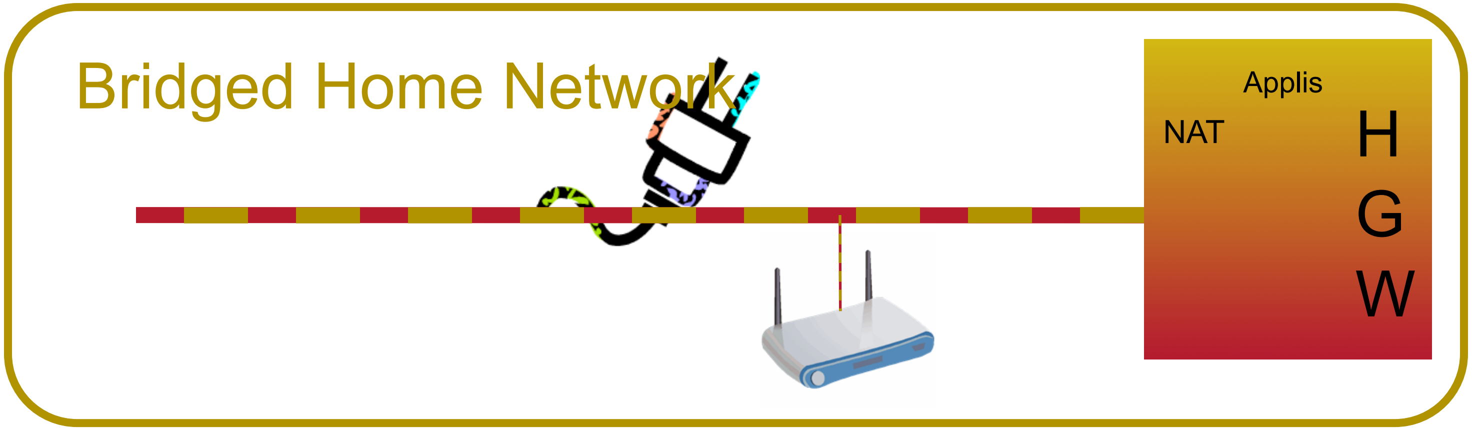

- **Have some dedicated applications outside of the gateway**
  - Managed by the provider ?
  - Security is a key element

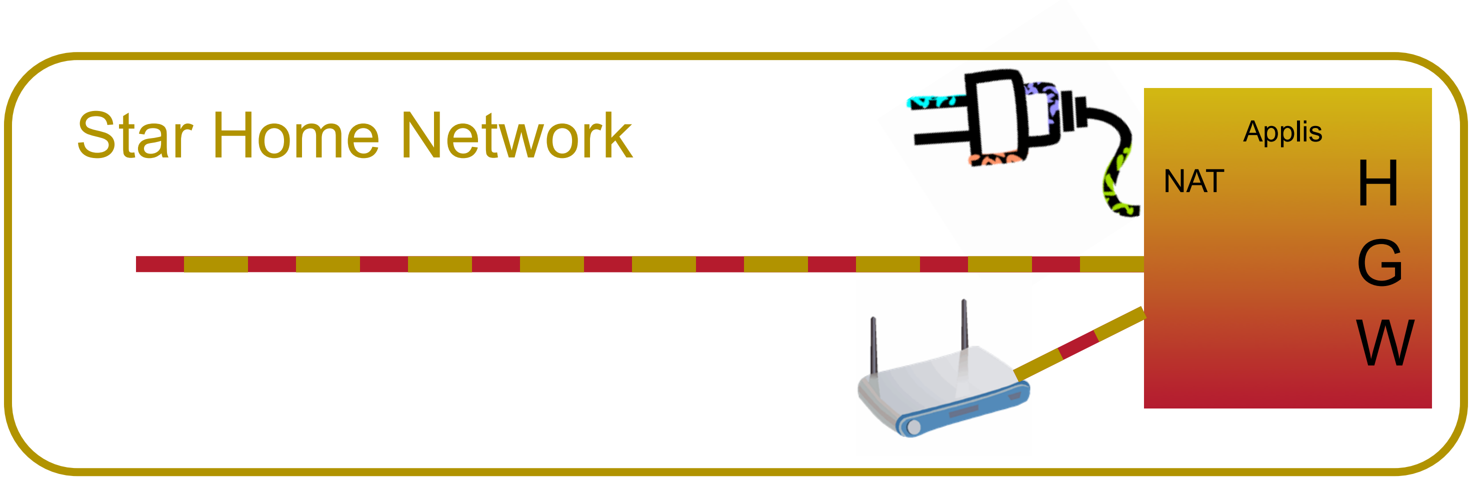
# Home Network Architecture


- **Better security than UPnP NAT context setting**
- **Authentication is a way to maintain links between providers HGW and applications**
  - Standard protocols or pre-registered keys ?

# Home Network Architecture

# Router Auto-configuration

- **User can build complex architecture**
  - If Bridging is used: loops must be detected
  - Spanning Tree is not efficient for Traffic Engineering
  - Traffic will converge on some links
- **Routing will allow more control:**
  - Routers have to be configured

| GP = provider | SID = ? | I-ID = autoconf |
|---|---|---|

# DHCPv6 Prefix Delegation


- **Main idea: The edge router**
  - become the DHCPv6 server for prefixes (/64) for the home network.
  - Get a global prefix for the provider.
  - Create a pool of GP:SID to reach the /64 boundary
  - Allocate these prefixes to routers
- **When a router starts :**
  - Periodically broadcast requests until receiving an answer from a DHCPv6 server
  - When configured act as a DHCPv6 relay.
- **More studies on multi-homing and network stability are needed**

# No Administration Protocol

- `draft-chelius-router-autoconf-00.txt`
- **Main idea:**
  - GP is flooded to routers.
  - One router on a link select randomly a SID to reach /64 limit.
  - SID is flooded to detect collision.
  - If collision smallest Router ID keep it, other routers drawn another value amongst the available ones.
  - When there is no conflict on SID value, Prefix (/64) is announced through Neighbor Discovery.
- **Deal with multi-homing (either is GP length different)**
  - SID will be different for each prefix on a link
- **Algorithm efficiency ?**

# Results

| GP vs #Links | 5 links | 10 links | 15 links | 20 links | 25 links |
|---|---|---|---|---|---|
| /60 | 1.5 | 2.19 | 3.06 | n.a. | n.a |
| /56 | 1.04 | 1.16 | 1.34 | 1.53 | 1.70 |
| /52 | 1.00 | 1.01 | 1.03 | 1.05 | 1.07 |
| /48 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

Fast convergence

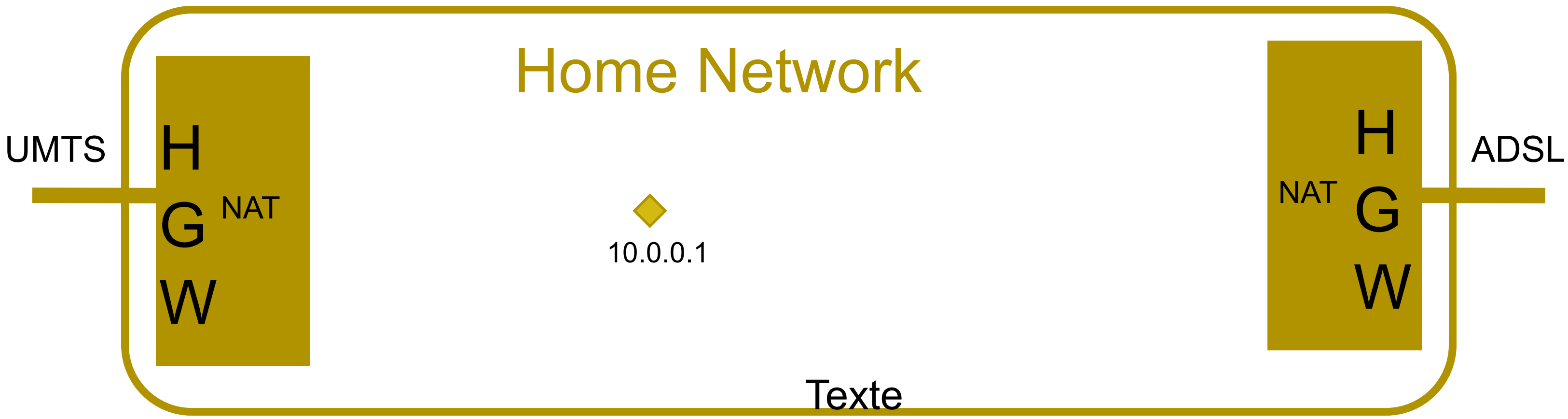
# Results

| GP vs #Links | 5 links | 10 links | 15 links | 20 links | 25 links |
|---|---|---|---|---|---|
| /60 | 1.5 | 2.19 | 3.06 | n.a. | n.a |
| /56 | 1.04 | 1.16 | 1.34 | 1.53 | 1.70 |
| /52 | 1.00 | 1.01 | 1.03 | 1.05 | 1.07 |
| /48 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

Fast convergence

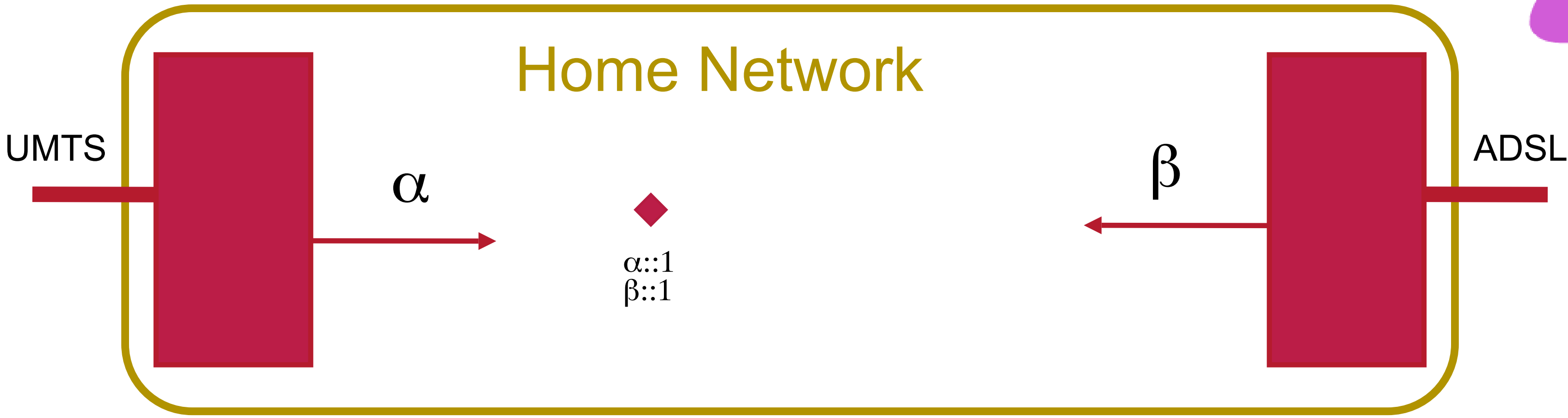
No conflicts

# IPv4 Multi-homing




- **Private addresses for hosts**
- **Packets are routed to the closest exit router**
- **Exit router will change the source address to the provider's address**
- **Applications are not multi-home aware**

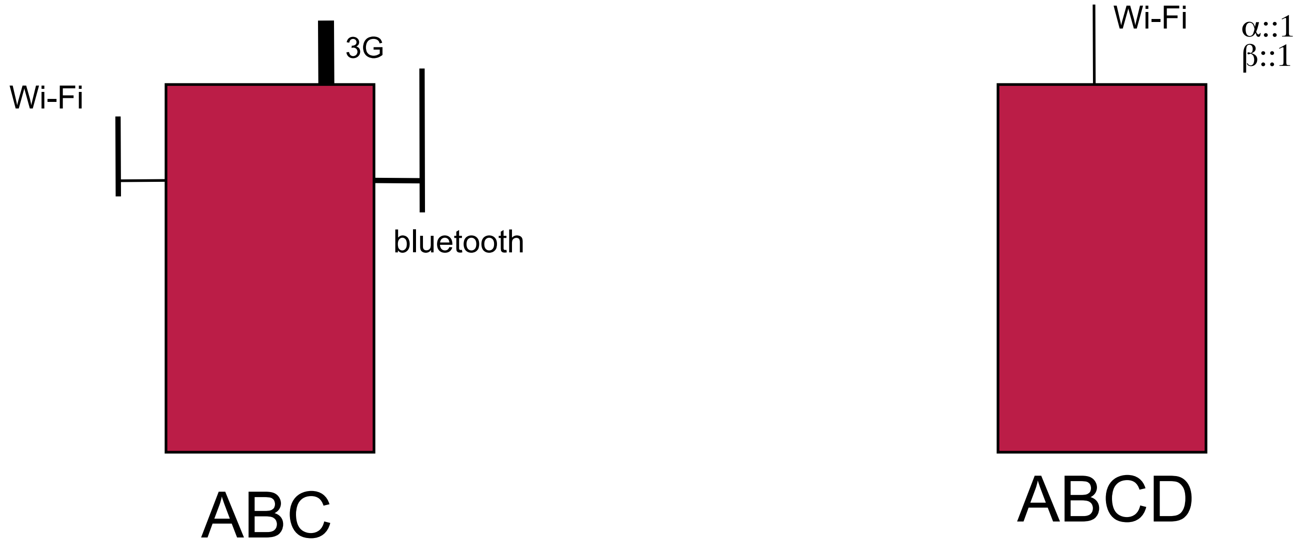
# IPv6 Multi-homing

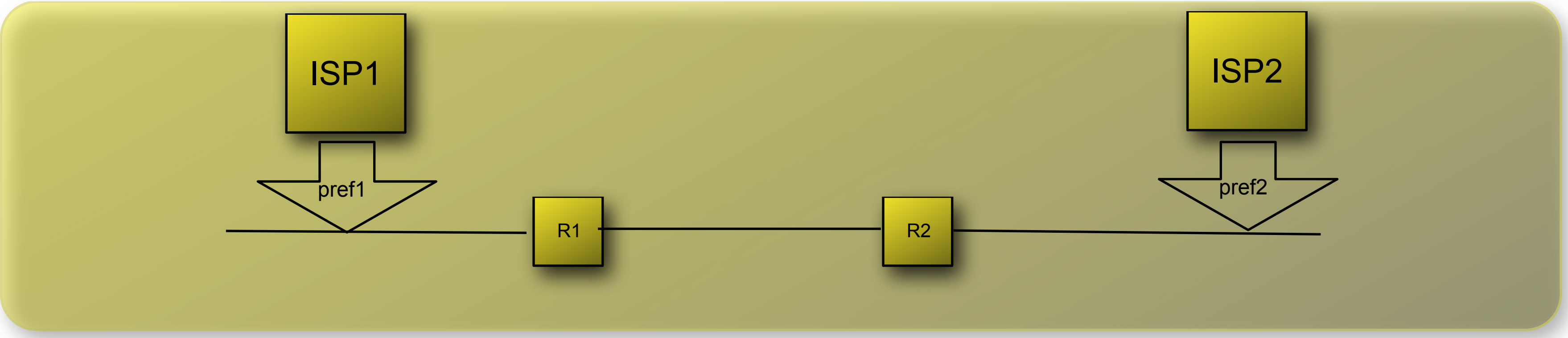

- **Host will have one per providers**
  - Rules to select source address are very simple
- **Routing is based mainly on default route**
  - Packet may led to the wrong provider and discarded
- **Modify IGP to handle source address in default routing ?**

# ABC Extension


- Improve IGP to handle source address properly
- When an equipment selects a provider by selecting the source address

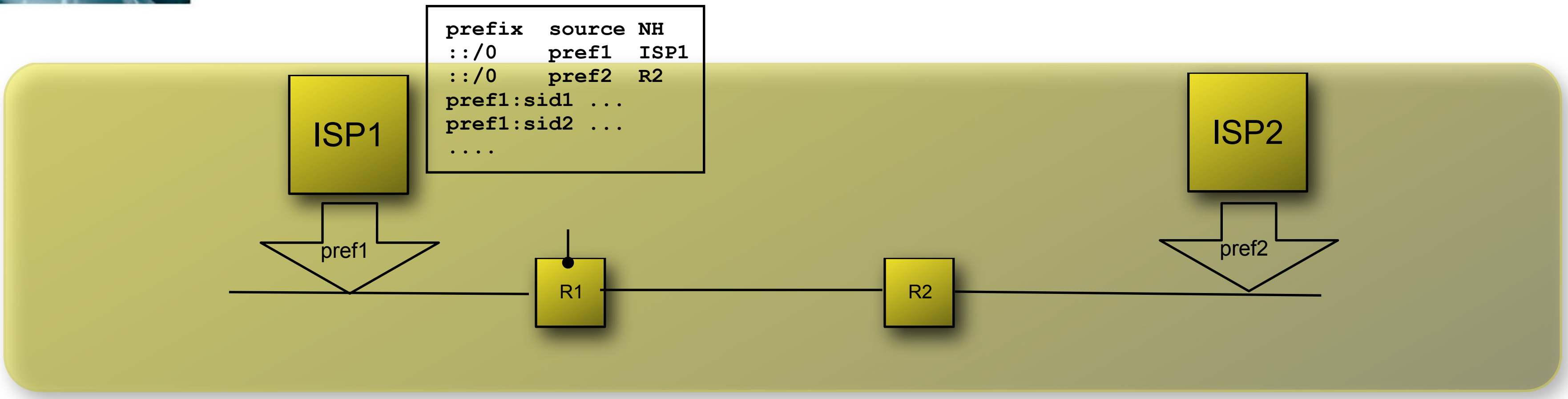


# Multi-homing routing



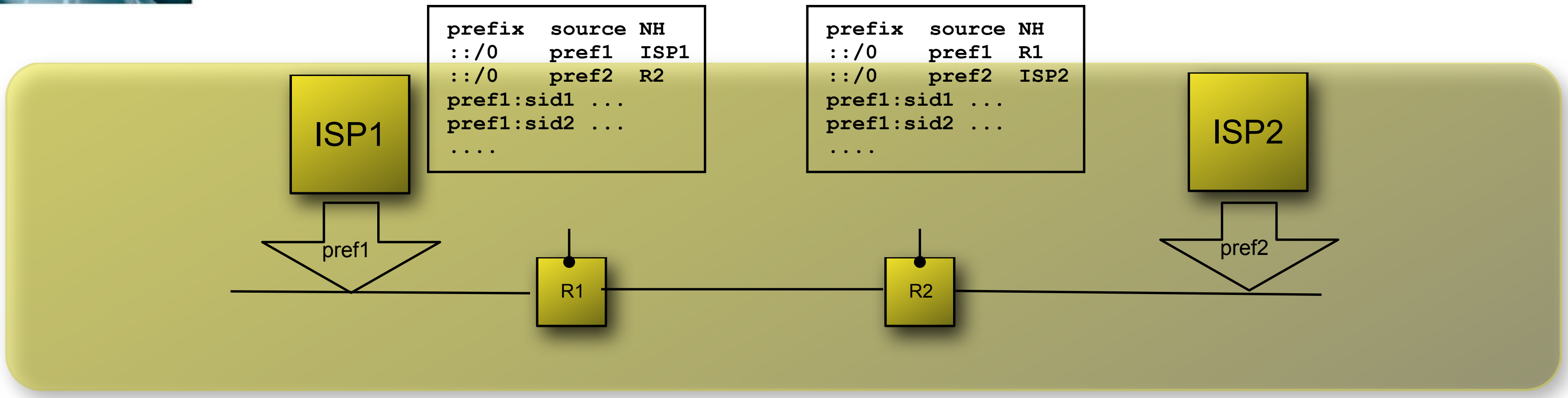

- **For NAP algorithm, prefix must be flooded on the site**
  - change forwarding behavior for default route:
  - ➡ When a default route is selected :
    - ➡ source address is a Global Address
    - ➡ have different entries in routing table for each GP
    - ➡ longest prefix match on the source address
    - ➡ DV cost will avoid loops

# Multi-homing routing



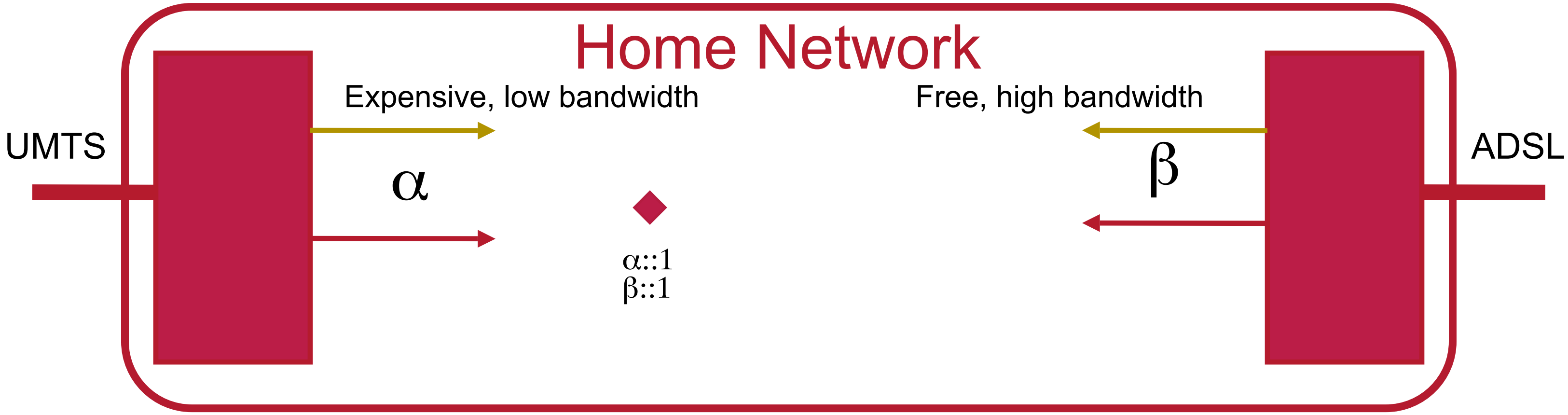

- **For NAP algorithm, prefix must be flooded on the site**
  - change forwarding behavior for default route:
  - ➡ When a default route is selected :
    - ➡ source address is a Global Address
    - ➡ have different entries in routing table for each GP
    - ➡ longest prefix match on the source address
    - ➡ DV cost will avoid loops

# Multi-homing routing




- **For NAP algorithm, prefix must be flooded on the site**
  - change forwarding behavior for default route:
  - ➡ When a default route is selected :
    - ➡ source address is a Global Address
    - ➡ have different entries in routing table for each GP
    - ➡ longest prefix match on the source address
    - ➡ DV cost will avoid loops

# ABCD example




- **Peer to peer application:**
  - Use β prefix - If β fail, wait
- **VoIP application:**
  - Use β prefix - if β fail use α (a multi-homing mechanism will manage address change)
- **Monitoring application:**
  - Use β prefix - if β fail use α and reduce quality

# ABCD

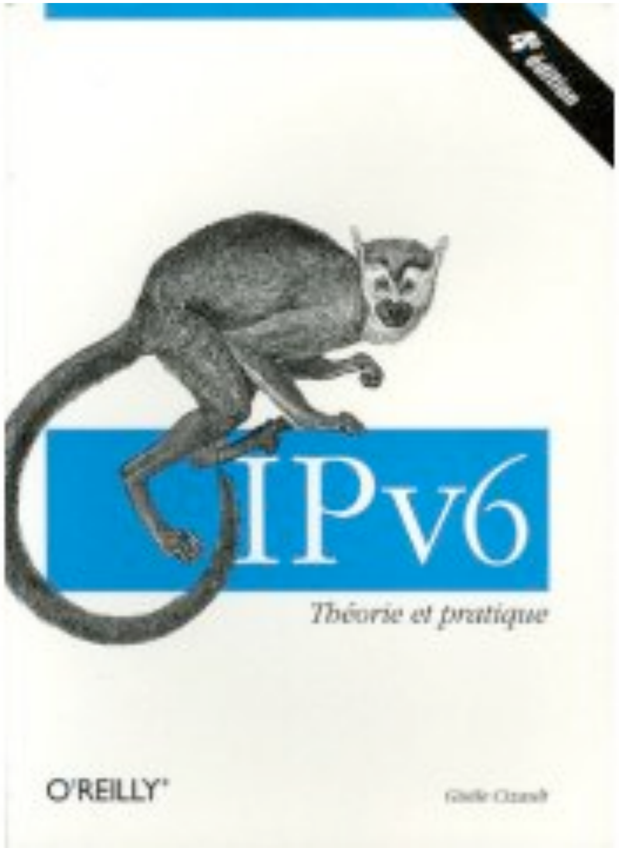

- **Edge routers using service discovery protocol gives information concerning providers network (cost, bandwidth, error rate, prefix…)**
- **Application selects source address regarding edge router information**
- **If one access fails, application decides the appropriate behavior**
  - Wait until network recover
  - Change addresses (source or destination)
- **Compatible with shim6 multi-homing approach**

enst Bretagne | get

# Routing protocol evolution


- **Current IGP:**
  - scalable
  - Traffic converge to high speed links
- **Home network:**
  - Relatively low bandwidth
  - No scalability problems
  - Spread as much as possible traffic to use available bandwidth
  - Manage the available bandwidth through scheduling

# Conclusion




RIPE-55
We recommend that service providers make their services available over IPv6. We urge those who will need significant new address resources to deploy IPv6. We encourage governments to play their part in the deployment of IPv6 and in particular to ensure that all citizens will be able to participate in the future information society. We urge that the widespread deployment of IPv6 be made a high priority by all stakeholders.

http://livre.g6.asso.fr

http://www.youtube.com/watch?v=_y36fG2Oba0