IPv6 Course

© G6 Association
Presentation : Jean-Luc Richier

4 Novembre 2008

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some Ressources

Table of Contents

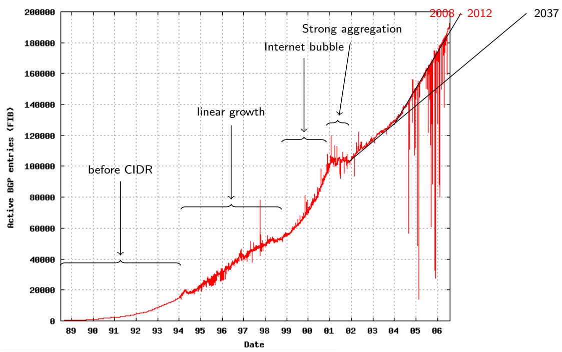
Historical facts

- 1983 : Research network for about 100 computers
- 1992 : Commercial activity
  - Exponential growth
- 1993 : Exhaustion of the class B address space
  - Allocation in the class C space
  - Require more information in routers memory

Historical facts

- Forecast of network collapse for 1998!
  - 1999 : Bob Metcalfe ate his Infoworld 1995 paper where he made this prediction
- RIRs statistics (May 2004)
  - http://www.ripe.net/ripe/meetings/ripe-48/presentations/ripe48-ap-rir-stats.pdf


Current Usage



Source:
http://www.cidr-report.org (August 2006)

The Facts
IPv6 Protocol
IPv6 Header
IPv6 Extensions
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some Ressources

IPv6 packet Format

IPv6 Header : Simpler

Definition

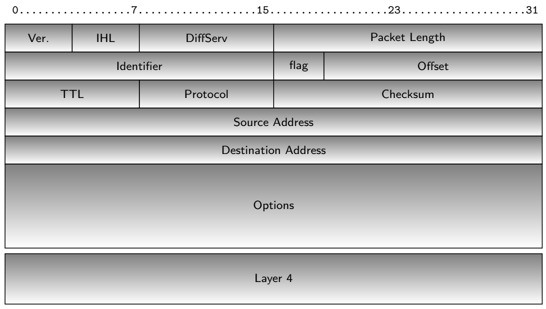
- IPv6 header follows the same IPv4 principle:
  - fix address size ... but 4 times larger
  - alignment on 64 bit words (instead of 32)
- Functionalities never used in IPv4 are supressed

Goal :

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

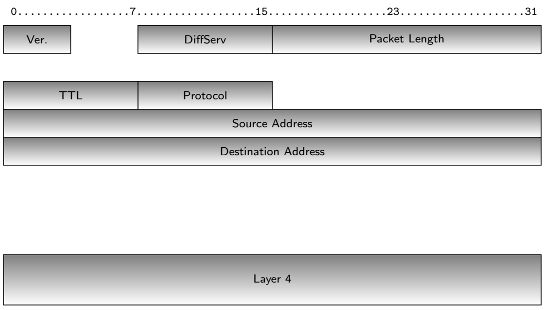
IPv6 Header

The Facts
IPv6 Protocol
IPv6 Header
IPv6 Extensions
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some Ressources



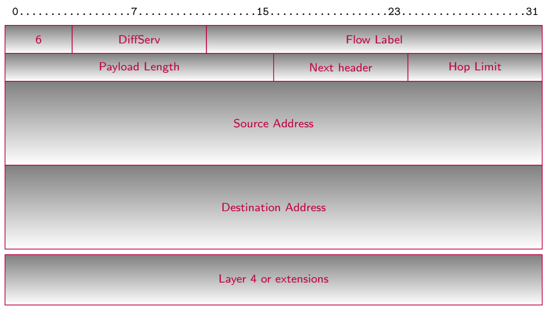
IPv6 Header

The Facts
IPv6 Protocol
IPv6 Header
IPv6 Extensions
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some Ressources



IPv6 Header

The Facts
IPv6 Protocol
IPv6 Header
IPv6 Extensions
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some Ressources



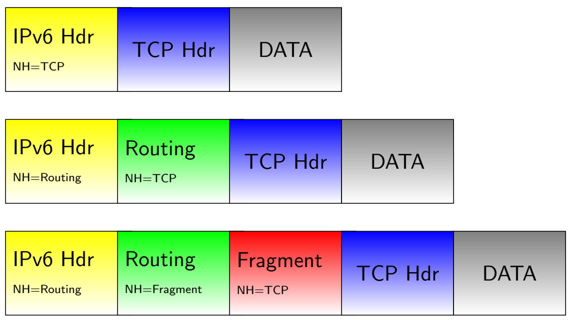
Extensions

The Facts
IPv6 Protocol
IPv6 Header
IPv6 Extensions
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some Ressources

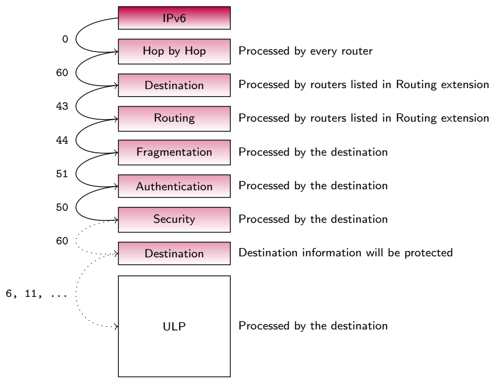
- Seen as a L4 protocol
- Processed only by destination
  - Except Hop-by-Hop processed by every router
  - Equivalent of option field in IPv4
- No size limitation
- Several extension can be linked to reach L4 protocol
- Processed only by destination
  - Destination (mobility)
  - Routing (loose source routing, mobility)
  - Fragmentation
  - Authentication
  - Security

Extensions in packets

The Facts
IPv6 Protocol
IPv6 Header
IPv6 Extensions
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some Ressources



Extension Order is Important



Addresses

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:

  2001:660:3003:1::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`
- CIDR prefix notation for nets `addr/lg, e.g. 2001:660:3::/40`

Warning:

`2001:660:3::/40` is `2001:660:0003::/40`, that is `2001:660::/40`, and not `2001:660:0300::/40`

Is it enough for the future ?

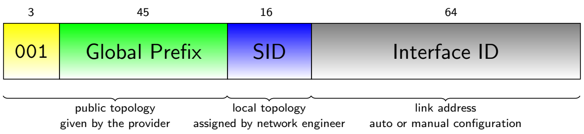
- Address length
  - Between 1 564 and 3 911 873 538 269 506 102 addresses by m^2
  - 60 000 trillion trillion addresses per inhabitant of the earth
  - Addresses for every grain of sands in the world
- Justification of a fix address length

Warning:

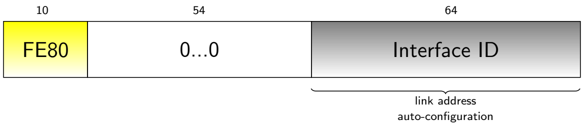
- An address for everything on the network and not an address for everything
- No addresses for whole life:
  - Depend of your position on the network
  - ISP Renumbering may be possible

Address Format

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)
    - ...

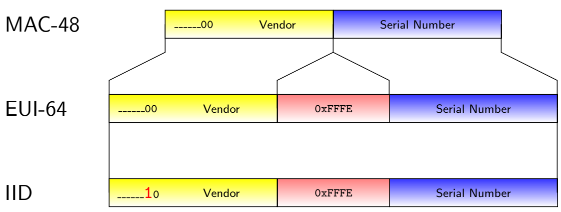
Interface Identifier

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 (Secure ND – SEND)
  - List of assigned prefixes
  - ...

How to Construct an IID from MAC Address

- 64 bits is compatible with EUI-64 (i.e. IEEE 1394 FireWire, ...)
- IEEE propose a way to transform a MAC-48 to an EUI-64
- U/L changed for numbering purpose

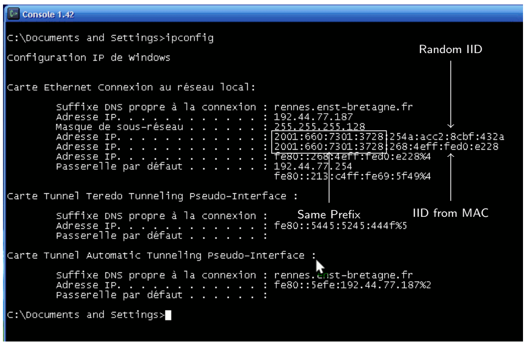


- There is no conflicts if IID are manually numbered: 1, 2, 3, ...

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
```

Windows XP



Local & Global Addresses

Link Local Scoped Addresses

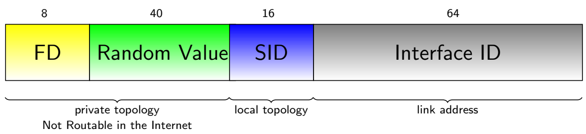
- Global Address, the prefix designates the exit interface
  - An interface can have multiple Global addresses (same prefix or not)
- Link-Local address, the prefix is always fe80::/10
  - The exiting interface is not defined
  - A %iface, can be added at the end of the address to avoid ambiguity.
- Example:

```
Routing tables

Internet6:

Destination        Gateway                      Flags    Netif Expire
default            fe80::213:c4ff:fe69:5f49%en0 UGSc       en0
```


Other kind of addresses : ULA

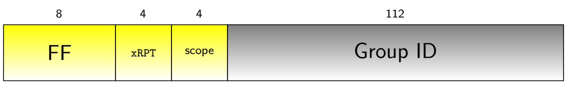
- Equivalent to the private addresses in IPv4
- But try to avoid same prefixes on two different sites:
  - avoid renumbering if two company merge
  - avoid ambiguities when VPN are used
- These prefixes are not routable on the Internet

Unique Local IPv6 Unicast Addresses:



Other kind of addresses: Multicast

Generic Format:



- T (Transient) 0: well known address - 1: temporary address
- P (Prefix) 1 : assigned from a network prefix (T must be set to 1)
- R (Rendez Vous Point) 1: contains the RP address (P & T set to 1)
- Scope :
  - 1 - node-local
  - 2 - link-local
  - 3 - subnet-local
  - 4 - admin-local
  - 5 - site-local
  - 8 - organisation-local
  - E - global

Some Well Known Multicast Addresses


```
FF02:0:0:0:0:0:0:1 All Nodes Address
FF02:0:0:0:0:0:0:2 All Routers Address
FF02:0:0:0:0:0:0:5 OSPFIGP
FF02:0:0:0:0:0:0:6 OSPFIGP Designated Routers
FF02:0:0:0:0:0:0:9 RIP Routers
FF02:0:0:0:0:0:0:FB mDNSv6
FF02:0:0:0:0:0:1:2 All-dhcp-agents
FF02:0:0:0:0:1:FFXX:XXXX Solicited-Node Address
FF05:0:0:0:0:0:1:3 All-dhcp-servers
```

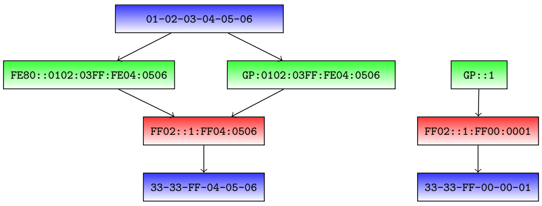
Web:

see : http://www.iana.org/assignments/ipv6-multicast-addresses for all multicast addresses

Solicited Multicast Addresses

The Facts
IPv6 Protocol
IPv6 Addresses
Notation
Format
Kind of addresses
Addressing scheme
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some Ressources

- Create a Multicast Address from a Unicast Address
  - Widely used for stateless auto-configuration
  - Avoid the use of broadcast



Example

The Facts
IPv6 Protocol
IPv6 Addresses
Notation
Format
Kind of addresses
Addressing scheme
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some Ressources

```
Vlan5 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::203:FDFF:FED6:D400
Description: reseau C5
Global unicast address(es):
  2001:660:7301:1:203:FDFF:FED6:D400, subnet is 2001:660:7301:1::/64

Joined group address(es):
  FF02::1  <- All nodes
  FF02::2  <- All routers
  FF02::9  <- RIP
  FF02::1:FFD6:D400  <- Solicited Multicast
```


Other kind of addresses ...

The Facts
IPv6 Protocol
IPv6 Addresses
Notation
Format
Kind of addresses
Addressing scheme
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some Ressources

- unassigned address (in6addr_any ::)
- loopback (::1)
- site local (FC00::/8) Stable site addresses
  Historic, replaced by ULA
- IPv4 mapped (::FFFF:0:0/96)
  Inclusion of IPv4 into IPv6
  (::FFFF:192.1.2.3 == 192.1.2.3)
  only inside applications, not on the net
- IP4 compatible (::/96) For transition - historic (::192.1.2.3)
- Many other, mainly transition proposals

The Facts
IPv6 Protocol
IPv6 Addresses
Notation
Format
Kind of addresses
Addressing scheme
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some Ressources

Addressing scheme
Prefix allocation

Addressing scheme

The Facts
IPv6 Protocol
IPv6 Addresses
Notation
Format
Kind of addresses
Addressing scheme
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some Ressources

- RFC 4291 defines current IPv6 addresses
  - loopback (::1)
  - link local (FE80::/10)
  - global unicast (2000::/3)
  - multicast (FF00::/8)
- Use CIDR principles:
  - Prefix / prefix length notation
  - 2001:660:3003::/48
  - 2001:660:3003:2:a00:20ff:fe18:964c/64
- Interfaces have several IPv6 addresses
  - at least a link local and a global unicast addresses

Addressing Space Utilization

The Facts
IPv6 Protocol
IPv6 Addresses
Notation
Format
Kind of addresses
Addressing scheme
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some Ressources

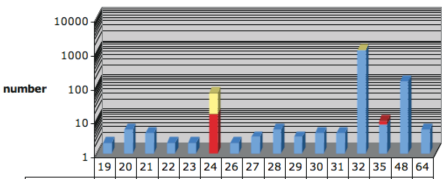
```
0000::/8 Reserved by IETF [RFC4291]
0100::/8 Reserved by IETF [RFC4291]
0200::/7 Reserved by IETF [RFC4048]
0400::/6 Reserved by IETF [RFC4291]
0800::/5 Reserved by IETF [RFC4291]
1000::/4 Reserved by IETF [RFC4291]
2000::/3 Global Unicast [RFC4291]
4000::/3 Reserved by IETF [RFC4291]
6000::/3 Reserved by IETF [RFC4291]
8000::/3 Reserved by IETF [RFC4291]
A000::/3 Reserved by IETF [RFC4291]
C000::/3 Reserved by IETF [RFC4291]
E000::/4 Reserved by IETF [RFC4291]
F000::/5 Reserved by IETF [RFC4291]
F800::/6 Reserved by IETF [RFC4291]
FC00::/7 Unique Local Unicast [RFC4193]
FE00::/9 Reserved by IETF [RFC4291]
FE80::/10 Link Local Unicast [RFC4291]
FEC0::/10 Reserved by IETF [RFC3879]
FF00::/8 Multicast [RFC4291]
```

Source:
http://www.iana.org/assignments/ipv6-address-space

Global Prefix

- Providers allocate a /48 to customers
  Note : may change to /56 for small customers (ADSL)
- A smaller prefix is allocated by RIR to provider depending on usage forecast

Prefix length distribution August 2006



	19	20	21	22	23	24	26	27	28	29	30	31	32	35	48	64
reclaimed						45							23			
returned						13							8	2		
allocated	2	5	4	2	2	1	2	3	5	3	4	4	109	7	132	5

prefix length

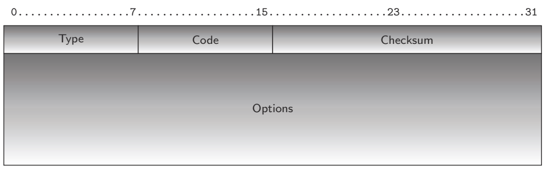
Web:
see : http://www.sixxs.net/tools/grh/dfp/

IPv6 mechanisms and ICMPv6

ICMPv6

- ICMPv6 is different from ICMP for IPv4
- Functionalities are extended and organized better

Format :



Precision

The field *type* code nature of the message ICMPv6
The field *code* specifies the cause of the message ICMPv6
The *checksum* used to verify the integrity of ICMP packet

ICMPv6 : Two Functions

- Error occurs during forwarding (*value* < 128)

Type	Signification
1	Destination Unreachable
2	Packet Too Big
3	Time Exceeded
4	Parameter Problem

- Management Applications (*value* > 128)

Type	Signification
128	Echo Request
129	Echo Reply
130	Group Membership Query
131	Group Membership Report
132	Group Membership Reduction
133	Router Solicitation
134	Router Advertisement
135	Neighbor Solicitation
136	Neighbor Advertisement
137	Redirect

Neighbor Discovery

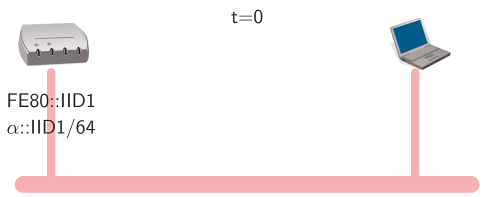
Neighbor Discovery [RFC 2461]

- IPv6 nodes sharing the same physical medium (link) use Neighbor Discovery (ND) to:
  - determine link-layer addresses of their neighbors
    - IPv4 : ARP
  - Address auto-configuration
    - Layer 3 parameters: IPv6 address, default route, MTU and Hop Limit
    - Only for hosts !
    - IPv4 : impossible, mandate a centralized DHCP server
  - Duplicate Address Detection (DAD)
    - IPv4 : gratuitous ARP
  - maintain neighbors reachability information (NUD)
- uses mainly multicast addresses but take into account NBMA Networks
- and ICMPv6 messages :
  - 133: Router Solicitation, 134: Router Advertisement, 135: Neighbor Solicitation, 136: Neighbor Advertisement, 137: Redirect Message.

Stateless Auto-configuration

Stateless Auto-configuration: Basic Principles

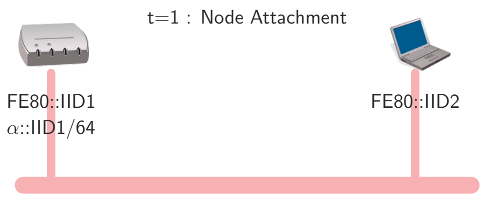
The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some



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

Stateless Auto-configuration: Basic Principles

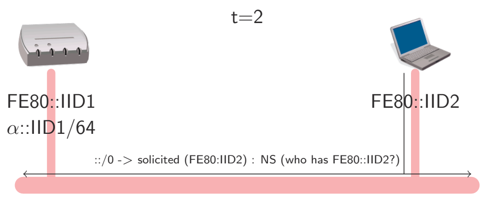
The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some



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

Stateless Auto-configuration: Basic Principles

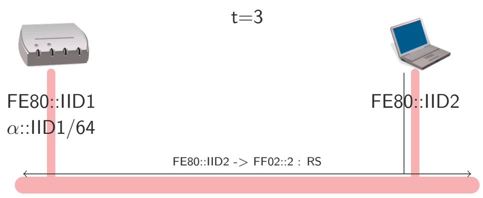
The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some



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

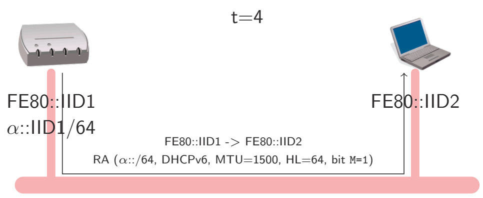
The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some



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

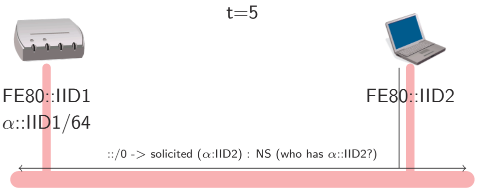
Stateless Auto-configuration: Basic Principles

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some



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 `M` = 1.

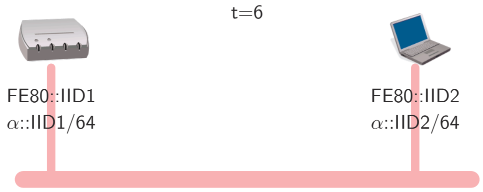
Stateless Auto-configuration: Basic Principles



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

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some



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

Note : Site parameters (DNS ...) are still missing, but connectivity is OK.

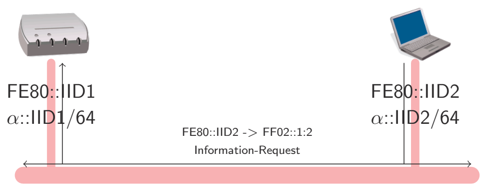
Router Configuration Example

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some

```
interface Vlan5
 description reseau C5
 ip address 192.108.119.190 255.255.255.128
...
 ipv6 address 2001:660:7301:1::/64 eui-64
 ipv6 enable
 ipv6 nd ra-interval 10
 ipv6 nd prefix-advertisement 2001:660:7301:1::/64 2592000
 604800 onlink autoconfig
```


Stateless DHCPv6 : get static parameters

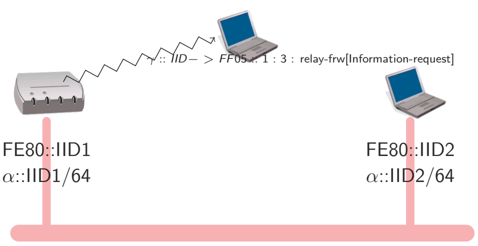
The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some



Host needs only static parameters (DNS, NTP,...). It sends a Information-Request message to All_DHCP_Agents group. The scope of this address is link-local.

Stateless DHCPv6 : get static parameters

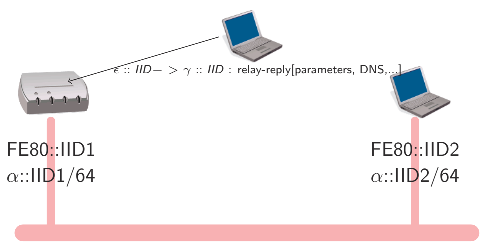
The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some



A relay (generally the router) encapsulate the request into a forward message and send it either to the All_DHCP_Servers site local multicast group or to a list of pre-defined unicast addresses.

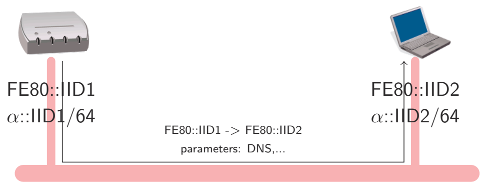
Stateless DHCPv6 : get static parameters

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some



Server answers to the relay

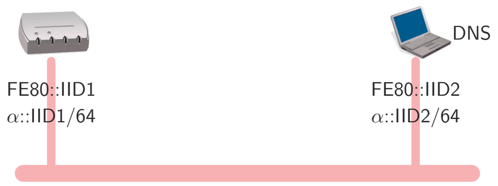
Stateless DHCPv6 : get static parameters



Router extracts information from the message to create answer and send information to the host

Stateless DHCPv6 : get static parameters

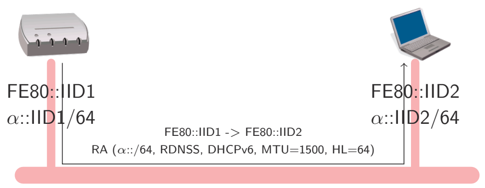
The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some



Host is now configured to solve name through DNS

Alternative Solution : new option for RA

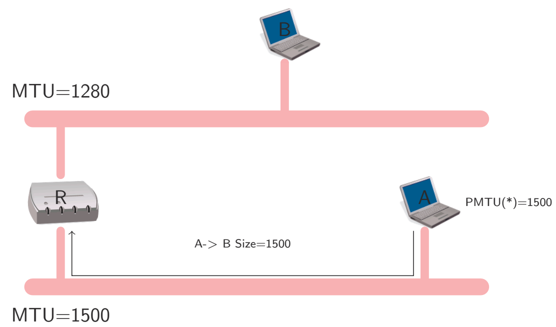
The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some



RFC 5006 : IPv6 Router Advertisement Option for DNS Configuration proposes a new option for RA. RDNSS (Recursive DNS Server) returns the IPv6 address of a resolver.

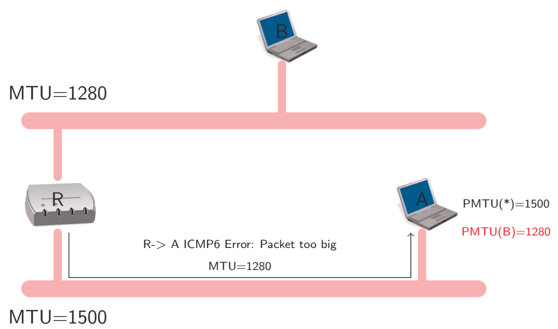
Path MTU discovery

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some



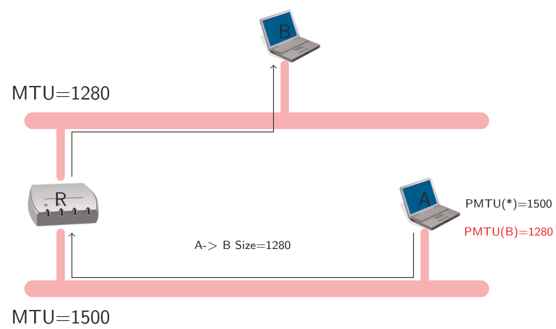
Path MTU discovery

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some



Path MTU discovery

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some



Security issues with Neighbor Discovery

From an attacker point of view, IPv6 attacks are:

- **Difficult** from remote network:
  - Scanning IPv6 network is hard (2∧64 addresses)
  - No broadcast address
  - Remote attacks will target hosts exposed in DNS
- **Easy** from local network:
  - Neighbor Discovery is not (yet) secured
  - Attacks inspired by ARP flaws + new attacks
  - Implementations not (yet) heavily tested

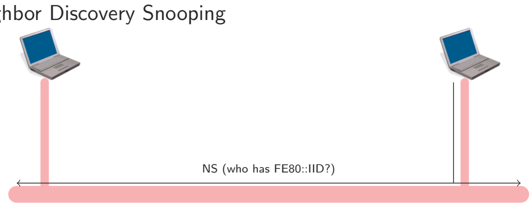
Attacker toolkits already available !

See `http://www.thc.org/thc-ipv6/`

Examples of attacks using ND

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some

Neighbor Discovery Snooping



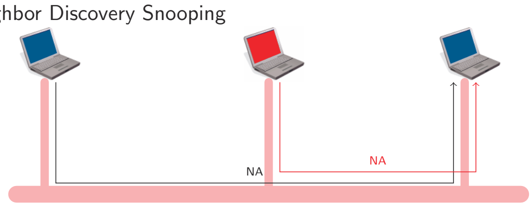
Host use Neighbor discovery for two cases :

- To get MAC address of another host (ARP-like)
- To verify address uniqueness (DAD)

Examples of attacks using ND

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some

Neighbor Discovery Snooping



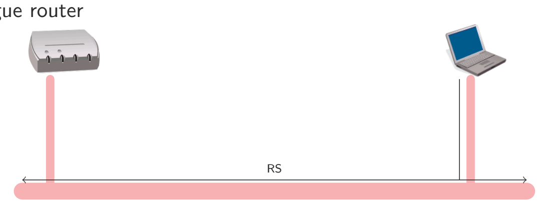
An attacker on the LAN can perfom attack by responding to ND messages

- ARP-like: Pretend to be any host on the LAN => Man in the Middle
- DAD: Pretend to have any address on the LAN => Deny of Service

Examples of attacks using ND

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some

Rogue router

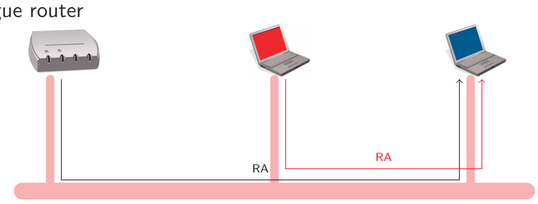


Host use Router Sollicitation to get address of the exit router and prefix used on the LAN.

Examples of attacks using ND

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some

Rogue router



An attacker on the LAN can perfom attack by responding to RS messages

- Pretend to be the exit router => Man in the Middle
- Pretend to route another prefix on the LAN => Deny of Service

Solutions to reduce or prevent attacks ?

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
ICMPv6
Neighbor Discovery
Path MTU discovery
ND Security
DHCPv6
Stateless vs Statefull
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Some

Prevention of attacks:

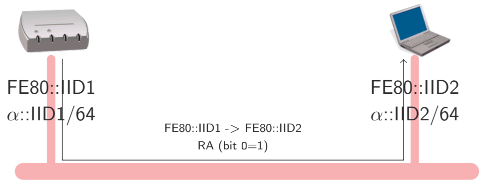
- SEND (Secure Neighbor Discovery)
  - IETF solution (Work in Progress)
  - Use signed ND messages, with a trust relationship
- Level-2 Filtering
  - Filter ND on switch port (ex. only one port allowed to send RA)
  - A few switch still implements it ... (Cisco ?)

Detection of attacks: ndpmon

- Similar to ARP-watch
- Detect Snooping and Denial of Services
- `http://ndpmon.sf.net`

Stateful Auto-configuration

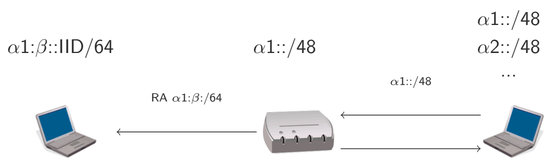
DHCPv6 : Stateful Auto-Configuration



Router answers to RS with a RA message with bit O set to 1.
Host should request its IPv6 address to a DHCP server.

DHCPv6 : Prefix Delegation

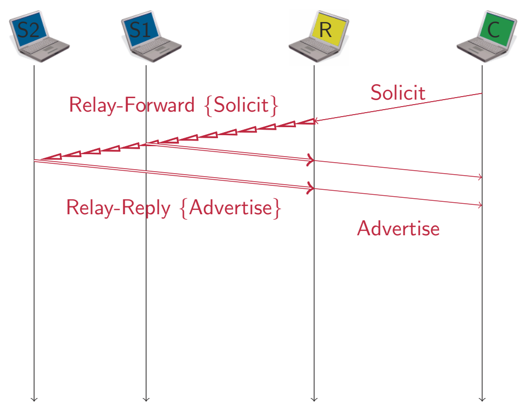
- Dynamic configuration for routers
- ISP solution to delegate prefixes over the network



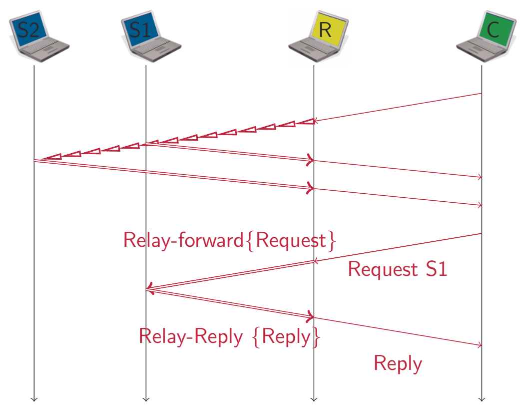
DHCPv6 Full Features

- For address or prefix allocation information form only one DHCPv6 must be taken into account. Four message exchange :
  - Solicit : send by clients to locate servers
  - Advertise : send by servers to indicate services available
  - Request : send by client to a specific server (could be through relays)
  - Reply : send by server with parameters requested
- Addresses or Prefixes are allocated for certain period of time
  - Renew : Send by the client tells the server to extend lifetime
  - Rebind : If no answer from renew, the client use rebind to extend lifetime of addresses and update other configuration parameters
  - Reconfigure : Server informs availability of new or update information. Clients can send renew or Information-request
  - Release : Send by the client tells the server the client does not need any longer addresses or prefixes.
  - Decline : to inform server that allocated addresses are already in use on the link

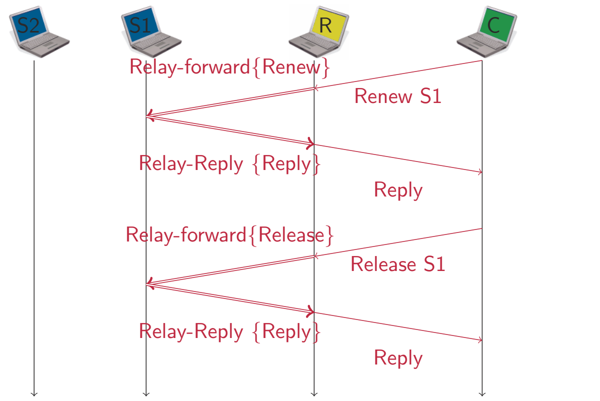
DHCPv6 Scenarii



DHCPv6 Scenarii



DHCPv6 Scenarii



DHCPv6 Identifiers

- DHCPv6 defines several stable identifiers
- After a reboot, the host can get the same information.
- DUID (DHCPv6 Unique IDentifier) :
  - Identify the client
  - Variable length:
    - Link-layer address plus time
    - Vendor-assigned unique ID based on Enterprise Number
    - Link-layer address
- For instance:

```
>od -x /var/db/dhcp6c_duid
0000000 000e 0100 0100 5d0a 5233 0400 9e76 0467
```


DHCPv6 Identifier : IA and IA_PD

- IA and IA_PD are used to link Request and Reply
  - IA is used for Address Allocation and is linked to an Interface
  - IA_PD is used for Prefix Delegation and can be shared among interfaces
- They must be stable (e.g. defined in the configuration file)

Auto-configuration: Stateless vs. Statefull

Stateless

Pro:

- Reduce manual configuration
- One server (the router)

Cons:

- Non-obvious addresses
- No control on addresses on the LAN

Statefull (DHCPv6)

Pro:

- Control of addresses on the LAN
- Control of address format

Cons:

- Require an extra server
- Still need RA mechanism
- Clients to be deployed

- Stateless: For Plug-and-Play networks (Home Network)
- Statefull: For administrated networks (enterprise, institution)

Routing protocols

Routing Protocols for IPv6

- **RIPng** (RFC 2080, 2081) : Extension of RIP for IPv6 (exclusive)
- **OSPFv3** (RFC 2740) : OSPF for IPv6 (exclusive)
- **ISIS** : Can manage both routing plans
- **MP-BGP (BGP4+)** (RFC 2545) : Multi-protocol extension

Conclusion

- No major differences with IPv4
- always needs explicit prefix length and aggregation information

IPv6 & User protocols

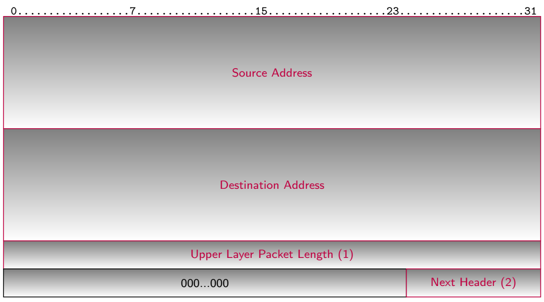
Links with ULP

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
IPv6 Impact on ULP
Header Compression
IPv6 Multihoming
IPv6 in DNS
IPv6 Firewall
IPSec
QOS & Mobility
Programming IPv6
Applications
IPv6 Integration
Some

- IPv6 mandate the use of checksum at layer 4.
  - There is no checksum in IPv6 header
- The checksum includes information for layer 3
- Algorithm to calculate checksum
  1. Layer 4 create a pseudo-header structure
  2. Pseudo-header is filled with addresses, protocol, length, etc.
  3. Checksum is computed
  4. Layer 4 sends packet to layer 3
- If layer 3 adds extensions, they are excluded from checksum
  - In pseudo-header, Next Header = L4 protocol

IPv6 pseudo-header

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
IPv6 Impact on ULP
Header Compression
IPv6 Multihoming
IPv6 in DNS
IPv6 Firewall
IPSec
QOS & Mobility
Programming IPv6
Applications
IPv6 Integration
Some



(1) Extended to 32 bits for Jumbograms

(2) Does not refer to extension

Consequences

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
IPv6 Impact on ULP
Header Compression
IPv6 Multihoming
IPv6 in DNS
IPv6 Firewall
IPSec
QOS & Mobility
Programming IPv6
Applications
IPv6 Integration
Some

- No change for TCP or ICMP
- For UDP, in IPv4 a zero-value checksum means that checksum is not computed
- This behavior is forbidden in IPv6, Checksum is mandatory
- For audio and video coding this may lead to some bigger errors :
  - Codec can deal with errors in data payoad
  - In IPv4, if no UDP checksum is set, this error will not be detected, so the application will receive some data (with errors)
  - In IPv6, since checksum is mandatory, the packet will be discarded by UDP, so the application will loose more information.
- New protocol needed for A/V transport

UDP-Lite

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
IPv6 Impact on ULP
Header Compression
IPv6 Multihoming
IPv6 in DNS
IPv6 Firewall
IPSec
QOS & Mobility
Programming IPv6
Applications
IPv6 Integration
Some

- In UDP the length field is redundant with IP length field
- UDP-lite use that field to indicate the scope of L4 checksum
  - 8 : only on headers
  - Full length : same behavior as UDP
- UDP-Lite is now standard (RFC3828), and is getting integrated into OS (Linux 2.6.20)

Header Compression (ROHC)

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
IPv6 Impact on ULP
Header Compression
IPv6 Multihoming
IPv6 in DNS
IPv6 Firewall
IPSec
QOS & Mobility
Programming IPv6
Applications
IPv6 Integration
Some

- The header compression for IPv6 is more efficient than for IPv4
  - IPv4 identifier field evolution is impredictable
  - Every IPv6 field are almost fixed
- In UDP, since the checksum is mandatory, this field must be sent
  - In normal case, with ROHC, a packet IPv6/UDP/RTP packet may be compressed in two bytes
  - Same value for IPv4

Multihoming in the IPv6 : Approaches

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
IPv6 Impact on ULP
Header Compression
IPv6 Multihoming
IPv6 in DNS
IPv6 Firewall
IPSec
QOS & Mobility
Programming IPv6 Applications
IPv6 Integration
Some

A lot of proposals:

- Routing approches
  - IPv6 multihoming with BGP
  - IPv6 Multihoming using Cooperation between Providers
  - IPv6 Multihoming Support at Site Exit Router
- Middle-Box approches
  - IPv6 Multihoming with NAT
  - Multihoming Aliasing and Translation Protocols
- Host-Centric approches
  - Transport layer approche
  - Network layer approach

Source:

Amine Dhraief, Doctorant At the ENST Bretagne, Rennes
Launois Cedric, Bagnulo Marcello, The paths towards IPv6 multihoming, IEEE Communications Surveys and Tutorials, 8, 2, 2006

IPv6 in DNS

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
IPv6 Impact on ULP
Header Compression
IPv6 Multihoming
IPv6 in DNS
IPv6 Firewall
IPSec
QOS & Mobility
Programming IPv6 Applications
IPv6 Integration
Some

IPv6 entries in DNS

DNS entry for a dual-stack host

```
rhadamante    A       192.108.119.134
              AAAA    2001:660:7301:1::1
```

Reverse DNS entry – 4 bits hexadecimal labels

```
$ORIGIN 1.0.0.0.1.0.3.7.0.6.6.0.1.0.0.2.ip6.arpa.
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR rhadamanthe.
```

Bind compatible since version 9.0

IPv6 in DNS

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
IPv6 Impact on ULP
Header Compression
IPv6 Multihoming
IPv6 in DNS
IPv6 Firewall
IPSec
QOS & Mobility
Programming IPv6 Applications
IPv6 Integration
Some

IPv6 Transport for DNS queries: Not mandatory

Make Bind listen on IPv6 :

```
listen-on-v6 { any; };
```

Client support:

- Simply set an IPv6 address for the DNS server
- *BSD, MacOSX: OK
- Linux: OK
- Windows: Problem with SP1 (and SP2 ?)

IPv6 in DNS

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
IPv6 Impact on ULP
Header Compression
IPv6 Multihoming
IPv6 in DNS
IPv6 Firewall
IPSec
QOS & Mobility
Programming IPv6 Applications
IPv6 Integration
Some

Do not forget to restrict recursion !

```
allow-recursion {
      192.108.119.0/24;
      2001:660:7301::/48;
      fe80::/10;
};
```

Link-local may be usefull !

Access control to network

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
IPv6 Impact on ULP
Header Compression
IPv6 Multihoming
IPv6 in DNS
IPv6 Firewall
IPSec
QOS & Mobility
Programming IPv6 Applications
IPv6 Integration
Some

Many sites use IP address allocation as access control to network (static DHCP)

Wrong design !

- Using IP address as User identifier
- Layer 2 access controlled by Layer 3
- Inherant security flaws !

Layer 2 access control should be done at layer 2 !

- 802.1x for Ethernet networks
- 802.11i for 802.11 networks

Set up IPv6 filtering

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
IPv6 Impact on ULP
Header Compression
IPv6 Multihoming
IPv6 in DNS
IPv6 Firewall
IPSec
QOS & Mobility
Programming IPv6 Applications
IPv6 Integration
Some

What do NOT change from IPv4

- Stateless firewall
- Statefull firewall: Possible to set up same security as NAT!

What do change from IPv4

- ICMP filtering: required for MTU discovery, errors, etc.
- Extensions: be carefull when deploying mobility

IPv6 support for firewall plateforms

- Cisco: PIX OS7, IOS 12.4 AdvancedIP (extended ACL)
- BSD Packet Filter
- Linux Netfilter (>2.6.20)

IPSec

- Mandatory in IPv6 item same functions as IPv4 :
  - Both integrity (ESP) and authentication (AH)
  - Both host to host and tunnel
- PKI are the same as in IPv4

However

- The main problem is key management

QOS & Mobility

- Mandatory in IPv6
- QOS
  - Both IntServ and Diffserv
  - Flowlabel field allows 2^{20} IntServ flows
- Mobility
  - Host has 2 addresses (Home and local)
  - Use a router as home agent
  - Use tunelling for direct transmission

However

- Only support in packet fields and basic API
- Real support needs a lot of developpements

IPv6 socket API

Socket API

- Socket Unix API has been extended to IPv6
- New protocol and address family PF_INET6 and AF_INET6
- New structures :
  - `in6_addr`
  - `sockaddr_in6`
  - `sockaddr_storage`
    Big enough to store any sockaddr.
- New functions for names to addresses conversion

Reference

RFC 2553 & Posix 1003.1g

Structure for sockets

Structure in C, C++

```
struct sockaddr_in6 {
  uint8_t         sin6_len;       // structure length if BSD
  sa_family_t     sin6_family;    // AF_INET6
  in_port_t       sin6_port;      // transport layer port
  uint32_t        sin6_flowinfo;  // IPv6 traffic class & flow info
  struct in6_addr sin6_addr;      // IPv6 address
  uint32_t        sin6_scope_id;  // set of interfaces for a scope
};
```

- Similar to sockaddr_in for IPv4
- New fields for scope and flow label

`sizeof(sockaddr_in6) > sizeof(sosckaddr_in)`

- sockaddr_in6 can not be stored in `struct sockaddr`
- Programs have to be modified to be AF-independent !

Managing sockets

- Creation : Same as in IPv4
  - `int s = socket(PF_INET6, SOCK_STREAM, 0);`
- Other functions are not modified
  - `bind, connect, listen, accept, send*, recv*, getpeername, getsockname`
- New functions to manage options
  - `getsockopt, setsockopt`

Sockets and address families

2 options for applications (system and/or application choice) :

- Only use PF_INET6 socket
  - On a IPv4 network, uses IPv4-mapped IPv6 addresses
  - Problem: when IPv6 stack is not available ...
- Use one PF_INET socket and one PF_INET6 socket
  - Client knows which socket to open with `getaddrinfo`
  - Server should wait for packets on both sockets

Examples found with `netstat -taun` (MacOSX)

Proto Rec Send Local Foreign State
tcp46 0 0 *.80 *.* LISTEN ← Apache server uses first option
...
tcp4 0 0 *.22 *.* LISTEN ← SSH server uses second option
tcp6 0 0 *.22 *.* LISTEN ←

Example : Client connection

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
CC++ API
JAVA API
IPv6 Integration
Some Ressources

```
#include <stdio.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netdb.h>
int open_conn(const char *host) {
    int sock = -1, ecode;
    struct addrinfo *res, *r, hints = {
        0, PF_UNSPEC, SOCK_STREAM, 0};

    if ((ecode = getaddrinfo(host, "daytime", &hints, &res)))
        errx(1, "getaddrinfo: %s", gai_strerror(ecode));
    for (r = res; r && sock < 0; r = res->ai_next)
        if ((sock = socket(res->ai_family, res->ai_socktype, res->ai_protocol)) < 0 ||
            connect(sock, res->ai_addr, res->ai_addrlen))
            sock = -1;
    freeaddrinfo(res);
    return sock;
}
```


Example : Server socket

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
CC++ API
JAVA API
IPv6 Integration
Some Ressources

```
#include <stdio.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netdb.h>
int open_serv(const char *serv) {
    int sock, ecode;
    struct addrinfo *res, hints = {
        AI_PASSIVE, PF_UNSPEC, SOCK_STREAM, 0};

    if ((ecode = getaddrinfo(NULL, serv, &hints, &res))
        errx(1, "getaddrinfo: %s", gai_strerror(ecode));
    if ((sock = socket(res->ai_family, res->ai_socktype,res->ai_protocol)) < 0) ||
         bind(sock, res->ai_addr, res->ai_addrlen) ||
        listen(sock, 1))
        err(1, "socket");
    freeaddrinfo(res);
    return sock;
}
```


Example : Server connection

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
CC++ API
JAVA API
IPv6 Integration
Some Ressources

```
main() {
    int sock = open_serv("1000");

    for(;;) {
        struct sockaddr_storage from;
        int s, len = sizeof from;
        char name[NI_MAXHOST];

        if ((s = accept (sock, (struct sockaddr*)&from, &len) < 0)
            err(1, "accept");

        if (getnameinfo((struct sockaddr*)&from, &len, name,
                sizeof name, NULL, 0, NI_NUMERICHOST))
            name[0] = 0;
        printf("connexion %s\n", name);
        /* utiliser socket s ? */
        close (s);
}
```


The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
CC++ API
JAVA API
IPv6 Integration
Some Ressources

Rules to anticipate integration of IPv6 protocol

Generic structure for sockets

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
CC++ API
JAVA API
IPv6 Integration
Some Ressources

- Programs should use `struct sockaddr_storage` to be AF-independent
- Cast depending of AF when needed

Socket containers

```
struct sockaddr_storage ss;
foo((struct sockaddr *)&ss);    // AF independent function

void foo(struct sockaddr *s) {
    // If we need IPv4 socket
    struct sockaddr_in *sin = (struct sockaddr_in *) s;
    // If we need IPv6 socket
    struct sockaddr_in6 *sin6 = (struct sockaddr_in6 *) s;
}
```

Address manipulation : getaddrinfo()

getaddrinfo() Prototype

```
int getaddrinfo(const char *nodename,
                const char *servname,
                const struct addrinfo *hints,
                struct addrinfo **res);
```

- Generic function for name resolution, AF-independent
- Replace function `gethostbyname`
- `servname`: String for protocol name ("http") or port number ("80")
- `hints`: Refine request (IPv4 only, IPv6 only, IPv4 & IPv6)
- May return more than one result (one or more families) !

Address manipulation : getnameinfo()

getnameinfo() Prototype

```
int getnameinfo(const struct sockaddr *sa,
                socklen_t salen,
                char *host,
                socklen_t hostlen,
                char *serv, socklen_t servlen,
                int flags);
```

- Generic function for reverse resolution, AF-independent
- Replace function `gethostbyaddr`
- Result are either name or number (option choice)

Macros

Macros to test nature of address:

- `IN6_IS_ADDR_UNSPECIFIED (struct in6_addr *);`
- `IN6_IS_ADDR_LOOPBACK (struct in6_addr *);`
- `IN6_IS_ADDR_MULTICAST (struct in6_addr *);`
- `IN6_IS_ADDR_LINKLOCAL (struct in6_addr *);`

Macros to test address equality :

- `IN6_ARE_ADDR_EQUAL (struct in6_addr *, struct in6_addr *);`

Porting applications to IPv6 (in a nutshell)

1: Replace IPv4-only structures and functions with AF-independent version

Generic Structure & Functions

`hostent` → `addrinfo`
`sockaddr_in` → `sockaddr_storage`
`gethostbyname` → `getaddrinfo`
`gethostbyaddr` → `getnameinfo`

2: Look for particular usage of IP address structure `in_addr`

- Applications sometimes use IP addresses as host identifier
- This should be made AF-independent

Porting applications to IPv6 (in a nutshell)

3: Choose a strategy when opening socket (one or two sockets?)

4: Consider one host may have more than one address !

- With `getaddrinfo` you may have one IPv4 and several IPv6 addresses for one host
- To be also considered when using address as host identifier

5: Beware of textual representation of IP addresses

Beware

http://[2001:660:7301:1::1]
scp foo.bar [2001:660:7301:1::1]:/tmp

IPv6 JAVA API

IPv6 Support in Java

- Java support IPv6 since JDK 1.2, extended with JDK 1.4
- Extension have been made for class InetAddress
- Inheritance and polymorphism ensures relative transparency for version of manipulated addresses

Inet6Address

New subclass of `InetAddress` (with `Inet4Address`)

- Class for instanciate IPv6 addresses
- Methods for checking address scope :
  - `isIPv4CompatibleAddress` (for IPv4-mapped addresses)
  - `isLinkLocalAddress`
  - `isMulticastAddress`

InetAddress

`InetAddress` objects may be either IPv4 or IPv6 address
`InetAddress` class extended for DNS resolution

- Method `getByName` returns only IPv4 name resolution
- New method `getAllByName` returns all possible name resolutions (IPv4 and IPv6)
- Reverse resolution unchanged

Changes for IPv6 support

Name resolution using `getByName` should be changed to use `getAllByName` and uses the returned array of addresses

Socket API

- Socket API is based on super-class InetAddress → no major change
- By choosing binding address, change protocol enabled for socket
  - IPv4 binding address → Socket listening for IPv4
  - IPv6 binding address → Socket listening for IPv4 and IPv6

Consequences

- Integration of IPv6 is harmless for IPv4 operations
- IPv6 will be used when correspondant address is IPv6

IPv6 integration in core network

IPv6 in core network

IPv6 Peering

- Global : GEANT, Opentransit
- France : SFINX (academic), PARIX (commercial)

IPv6 Routing

- Dedicated protocol needed

IPv6 Transport

- Native transport
  - Need to upgrade features of all routers !
- Tunnels (6over4, 6PE)
  - Temporary solutions => Need a transition plan from tunnel to native

6over4

Point-to-point tunnel: encapsulate IPv6 packets in IPv4 packets.

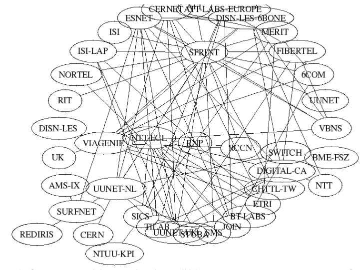
IPv4 Hdr	IPv6 Hdr	Payload

Proto=41 (IPv6)

Use case: Connecting 2 IPv6 islands separated by an IPv4-only network

- Require IPv6-capable routers at both ends
- MTU may be reduced, Performance of encapsulation
- Don't be over-used ... Remember the 6BONE !

6BONE



The 6Bone pTLAs and their tunnels. adamo@dblab.ece.ntua.gr, Fri Jun 9 07:30:28 EEST 2006

Lessons learned from 6BONE

- Too much distributed tunnels are very hard to maintain
- Tunnels should follow network topology

6PE : Provider Edge

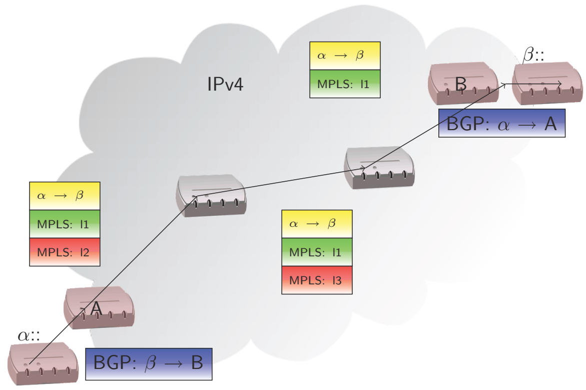
6PE = MPLS Tunnel (LSP) + MP-BGP

- Connect IPv6 islands across IPv4 network
- Automatic LSP tunneling between border routers (Layer 2.5)
- Initiation using MP-BGP announces
- LSP encapsulation to transport IPv6 packets on IPv4 network

Benefits of 6PE

- Specific configuration only on border routers
- Dynamic label allocation with BGP
- No major impact on performance

6PE : Provider Edge



IPv6 integration for ISP

IPv6 integration for ISP

IPv6 Transport: many solutions ...

- Native
- Automatic tunnels (6to4)
- Configured tunnels (Tunnel Broker, Softwire)

Manage IPv6 addresses

- Delegate IPv6 prefix to clients
- ISP will evolve from Address management to Prefix management

6to4 (RFC 3056)

Automatic IPv6 tunnels over IPv4

- Allow sites interconnection through an IPv4-only ISP
- Automatic => less management overhead compared to configure tunnels
- Transport using 6over4 technique

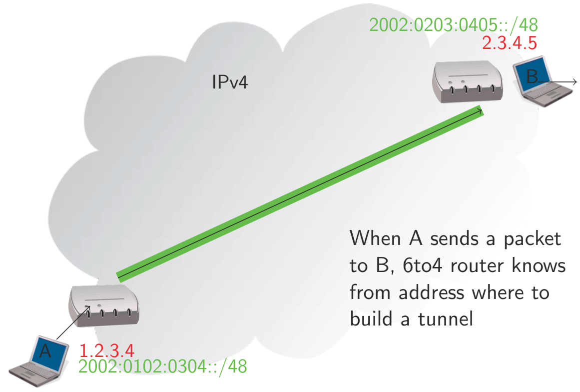
Special Address plan :

- Global IPv6 prefix build from IPv4 public address => No IPv6 provisioning for ISP
- Clients get a /48 prefix to address subnets

0....16............48.....64.......................128

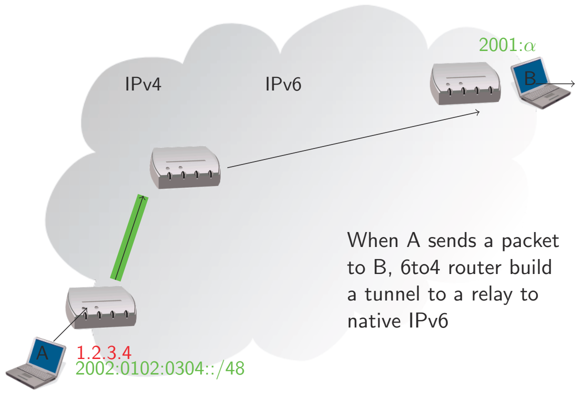
2002	*IPv4 Address*	SID	Interface ID

6to4 Architecture between 2 sites



When A sends a packet to B, 6to4 router knows from address where to build a tunnel

6to4 to connect with native IPv6



When A sends a packet to B, 6to4 router build a tunnel to a relay to native IPv6

6to4 relays

Relays to be reach with anycast addresses

- On IPv4 side: `192.88.99.1`
- On IPv6 side: `2002:c058:6301::/48`

Packets are routed to the closest 6to4 relays

6to4 Issues and limitations

Security issues

- ISP operating the relay should accept all clients ...
- Risk of spoofing
- 6to4 relays can be targeted by DoS attacks

Performance issues

- Long path to reach a relay (17 hops from french ISPs ...)
- 6to4 may lead to assymetrical routing

6to4 is not considered as a global solution.

Tunnel Broker

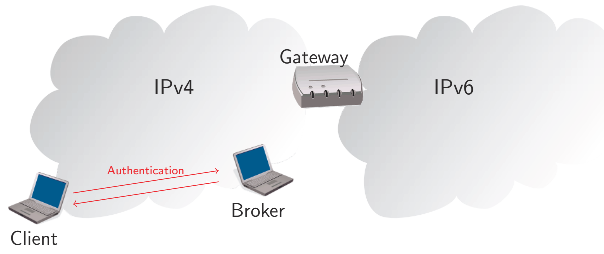
Tunnel Broker = 6over4 with client authentication

- TSP (*Tunnel Setup Protocol*) : XMLRPC-like protocol to set up tunnel parameters (authentication, delegated prefix, etc.)
- can delegate arbitrary prefixes
- use IPv6/IPv4 when public addresses available, IPv6/UDP/IPv4 for NAT-traversal
- IETF draft, commercial implementation by Hexago (free client and account)

Currently deployed by RENATER

`http://tunnel-broker.renater.fr/`

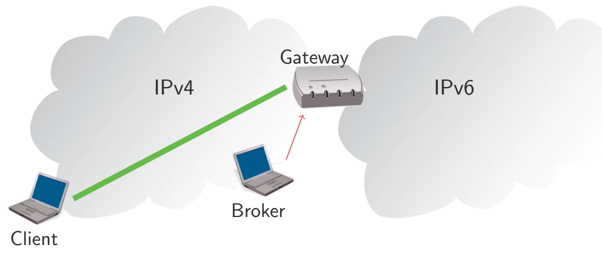
Tunnel Broker Architecture



Client authenticates to Broker and get Gateway parameters

Tunnel Broker Architecture

The Facts · IPv6 Protocol · IPv6 Addresses · IPv6 mechanisms · Routing Protocols · IPv6 & User protocols · Programming IPv6 Applications · IPv6 Integration · Core Network · ISP · Administrated Networks · Some Ressources



Client initiates IPv6 tunnel over IPv4 or UDP to Gateway

Softwires

The Facts · IPv6 Protocol · IPv6 Addresses · IPv6 mechanisms · Routing Protocols · IPv6 & User protocols · Programming IPv6 Applications · IPv6 Integration · Core Network · ISP · Administrated Networks · Some Ressources

IETF solution to transport and manage IPvX over IPvY.
Two scenarios discussed in the problem statement :

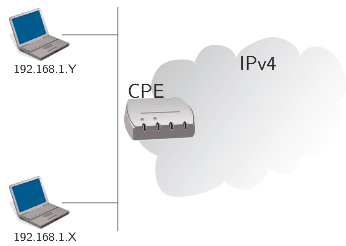
- Mesh problem : IPv4 over IPv6 in core network
  - Problem raised by chinese research network CERNET2
  - Connect IPv4 island across an IPv6 only backbone
  - Solution: MPLS tunnels
- Hub-and-spoke problem : IPv6 over IPv4 for Home Network
  - Problem raised by NTT, Comcast and Point6
  - Connect IPv6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

Currently deployed by RENATER and Point6

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

Softwires: H&S Architecture

The Facts · IPv6 Protocol · IPv6 Addresses · IPv6 mechanisms · Routing Protocols · IPv6 & User protocols · Programming IPv6 Applications · IPv6 Integration · Core Network · ISP · Administrated Networks · Some Ressources

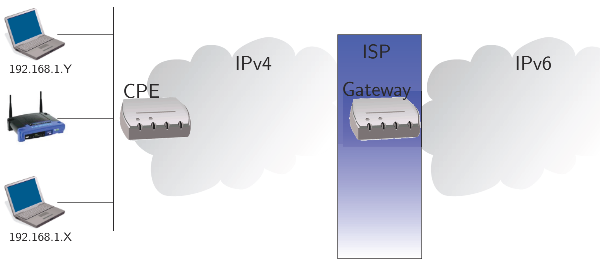


Current DSL ISPs connect Home Network with 1 IPv4 address:

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

Softwires: H&S Architecture

The Facts · IPv6 Protocol · IPv6 Addresses · IPv6 mechanisms · Routing Protocols · IPv6 & User protocols · Programming IPv6 Applications · IPv6 Integration · Core Network · ISP · Administrated Networks · Some Ressources

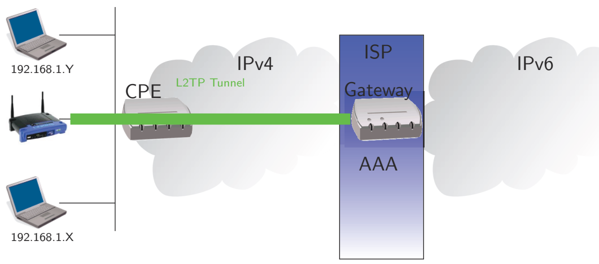


Idea: Build a virtual ISP for IPv6 :

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

Softwires: H&S Architecture

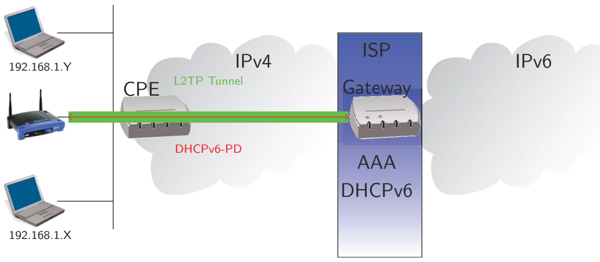
The Facts · IPv6 Protocol · IPv6 Addresses · IPv6 mechanisms · Routing Protocols · IPv6 & User protocols · Programming IPv6 Applications · IPv6 Integration · Core Network · ISP · Administrated Networks · Some Ressources



Tunneling with L2TP protocol :

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

Softwires: H&S Architecture

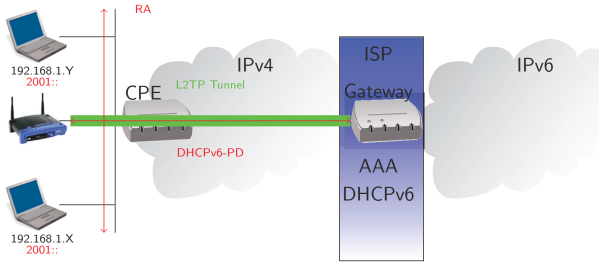


IPv6 prefix for Home Network provided by DHCPv6

- Standard prefix delegation
- Link with AAA for prefix management

Softwires: H&S Architecture

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Core Network
ISP
Administrated Networks
Some Ressources

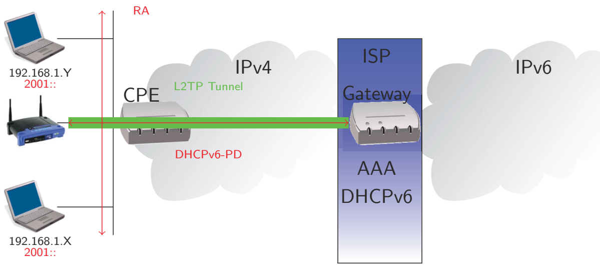


IPv6 addresses distributed with auto-configuration

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

Softwires: H&S Architecture

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Core Network
ISP
Administrated Networks
Some Ressources



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

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Core Network
ISP
Administrated Networks
Some Ressources

IPv6 Integration in administrated networks

IPv6 for administrated networks

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Core Network
ISP
Administrated Networks
Some Ressources

Motivations

- Not necessary shortage of addresses
- Gain experience on new technology
- IPv6 integration can be part of a network re-structuration

Goal

- Dual-Stack deployment
- Same Quality of Service in IPv6 as in IPv4

Problem may come from

- Time and money: resources available ?
- People: System administrators job is focused on IPv4. IPv6 is big changes for them ...

Agenda for IPv6 integration in administrated networks

The Facts
IPv6 Protocol
IPv6 Addresses
IPv6 mechanisms
Routing Protocols
IPv6 & User protocols
Programming IPv6 Applications
IPv6 Integration
Core Network
ISP
Administrated Networks
Some Ressources

- Get IPv6 connectivity and prefix
- Elaborate an address plan
- Deploy IPv6 to servers and users
- Set up IPv6 filtering
- Integrate IPv6 to services
- Monitor IPv6 usage

Get IPv6 connectivity

Native IPv6 provided by ISP

- In France, ask France Telecom, Nerim or RENATER

Tunneled IPv6 provided by ISP

- Does IPv4 ISP deploy 6to4 ? (In France, none)
- Configured tunnels (Tunnel Broker, Softwires)

IPv6 without deployment on access network: ISATAP

The Facts · IPv6 Protocol · IPv6 Addresses · IPv6 mechanisms · Routing Protocols · IPv6 & User protocols · Programming IPv6 Applications · IPv6 Integration · Core Network · ISP · Administrated Networks · Some Ressources

Technique to deploy IPv6 for isolated hosts without intermediate router configuration

Scenario

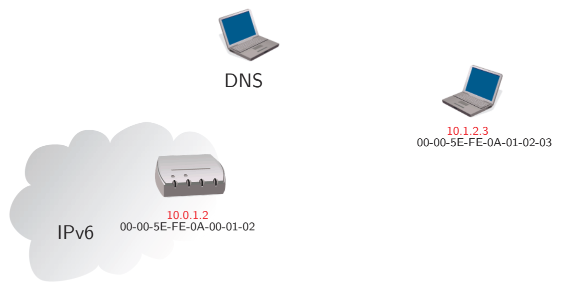
- ISATAP host build an IID with IPv4 address
- One router is designated as ISATAP router and connected to native IPv6 and IPv4.
  - host is register in DNS with standard name `isatap.domain`
  - A prefix is allocated to the router for ISATAP.
- Hosts sending IPv6 packets discover the dedicated router and prefix
- IPv6 packets are encapsulated in IPv4 (6over4)

This technique can be useful for VPN users

ISATAP can use 6to4 prefixes => Minimal deployment

ISATAP scenario

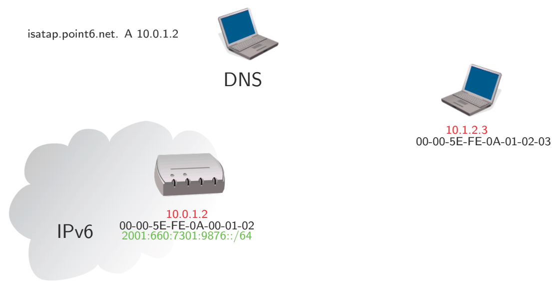
The Facts · IPv6 Protocol · IPv6 Addresses · IPv6 mechanisms · Routing Protocols · IPv6 & User protocols · Programming IPv6 Applications · IPv6 Integration · Core Network · ISP · Administrated Networks · Some Ressources



Hosts and router constructs IID from IPv4 address: `00-00-5E-FE-(IPv4 Address)`

ISATAP scenario

The Facts · IPv6 Protocol · IPv6 Addresses · IPv6 mechanisms · Routing Protocols · IPv6 & User protocols · Programming IPv6 Applications · IPv6 Integration · Core Network · ISP · Administrated Networks · Some Ressources

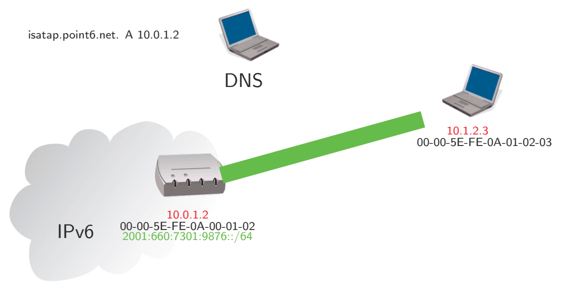


Router get prefix allocated for ISATAP

Standard name is registered to DNS.

ISATAP scenario

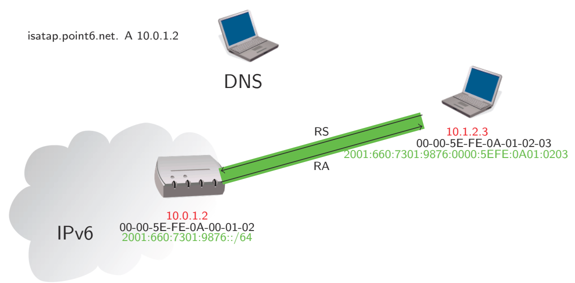
The Facts · IPv6 Protocol · IPv6 Addresses · IPv6 mechanisms · Routing Protocols · IPv6 & User protocols · Programming IPv6 Applications · IPv6 Integration · Core Network · ISP · Administrated Networks · Some Ressources



Host discover ISATAP router and automatically build 6over4 tunnel

ISATAP scenario

The Facts · IPv6 Protocol · IPv6 Addresses · IPv6 mechanisms · Routing Protocols · IPv6 & User protocols · Programming IPv6 Applications · IPv6 Integration · Core Network · ISP · Administrated Networks · Some Ressources



RA mechanism is then used to get prefix

Teredo

Remember NAT traversal technique from Skype ?
Teredo use the same technique to tunnel IPv6

- Some ISP deploying Teredo relays for clients
- Vista offers NAT-traversal and IPv6 at system-level
- Used by IPv6 applications from Vista (Meeting Space)

Potential security threat for sites !

- In a NAT context, Vista automatically setups a Teredo tunnel
- User can access IPv6 internet bypassing site filtering rules !

Address Plan

Sites usually get a /48 prefix from ISP.
How to allocate the 16bit of SID ? Many solutions ...

- Priority to routing
  - Aggregate prefixes by geographic site
  - Aggregation used in routing table
- Priority to filtering
  - Aggregate prefixes by users community
  - Aggregation used in filtering rules
- Mixed solutions
  - Test deployment: Use VLAN number as SID
  - More stable plan: See example from Univ. Rennes 1

Do not fear re-numbering !

Example of University Address Plan

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

- Filtering rules are based on the 4 first bits
- Routing tables are based on geographic prefixes
- Compromise: One filtering rule for all community BUT several routing rules for one geographic entity (one per community)

Deploy IPv6 on access network

- Wired networks
  - Ethernet, VLANs : no problem
    - Switch should all accept 0x86DD ethernet protocol
    - Some old switches may have problems with multicast ethernet addresses
  - ATM : Use LLC/SNAP encapsulation
- Wireless networks
  - 802.11: no problem
  - UMTS: 3GPP considering transition plan, No commercial offer yet

Auto-configuration: Stateless vs Statefull

Stateless

Pro:

- Reduce manual configuration
- One server (the router)

Cons:

- Non-obvious addresses
- No control on addresses on the LAN
- Security flaws

Statefull (DHCPv6)

Pro:

- Control of addresses on the LAN
- Control of address format

Cons:

- Require an extra server
- Still need RA mechanism (still vulnerable)
- Clients to be deployed

- Stateless: For guest and client VLANs ?
- Statefull: For server VLAN ?
- If concerned about security => static configuration !

Access control to network

Many sites use IP address allocation as access control to network (static DHCP)

Wrong design !

- Using IP address as User identifier
- Layer 2 access controlled by Layer 3
- Inherant security flaws !

Layer 2 access control should be done at layer 2 !

- 802.1x for Ethernet networks
- 802.11i for 802.11 networks

IPv6 in DNS

IPv6 entries in DNS

DNS entry for a dual-stack host

```
rhadamante    A      192.108.119.134
              AAAA   2001:660:7301:1::1
```

Reverse DNS entry

```
$ORIGIN 1.0.0.0.1.0.3.7.0.6.6.0.1.0.0.2.ip6.int.
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR rhadamanthe.
```

Bind compatible since version 9.0

IPv6 in DNS

IPv6 Transport for DNS queries: Not mandatory for AAAA queries !

Make Bind listen on IPv6 :

```
listen-on-v6 { any; };
```

Client support:

- *BSD, MacOSX: OK
- Linux: OK
- Windows: Problem with SP1 (and SP2 ?)

IPv6 in DNS

Do not forget to restrict recursion !

```
allow-recursion {
      192.108.119.0/24;
      2001:660:7301::/48;
      fe80::/10;
};
```

Link-local may be usefull !

Set up IPv6 filtering

What do NOT change from IPv4

- Stateless firewall
- Statefull firewall: Possible to set up same security as NAT !

What do change from IPv4

- ICMP filtering: required for MTU discovery, errors, etc.
- Extensions: be carefull when deploying mobility

IPv6 support for firewall plateforms

- Cisco: PIX OS7, IOS 12.4 AdvancedIP (extended ACL)
- BSD Packet Filter
- Linux Netfilter (>2.6.20)

IPv6 support for services

To be fully functionnal in IPv6, a service need support in:

- Access Network
- Operating System
- Service application (Server and Client side)

Applications need explicit support for IPv6

- Network features to be extended/rewritten to support dual stack
- Dual stack may impact on identifier representations, etc...

IPv6 support is coming slowly, but steadily

IPv6 support on Operating Systems

Microsoft Windows:

- < XP: Forget it ...
- XP: SP1 or SP2 OK
- Vista: OK

Unixes:

- *BSD, MacOSX: OK
- Linux: OK (2.6 kernel recommended)
- Solaris: OK (9 or 10)

Mainframe OSes: HPUX, AIX OK
Embedded OSes : WindRiver, Symbian OK

Overview of applications with IPv6 support

Web
- Server: Apache
- Client: Firefox, IE, Safari

Mail
- Server: Sendmail, Postfix, Exim
- Client: Thunderbird, Mail.app

Databases
- MySQL, PostgreSQL

Voice-Over-IP
- Asterisk

Application Inventory (in progress)

http://www.ipv6-to-standard.org/

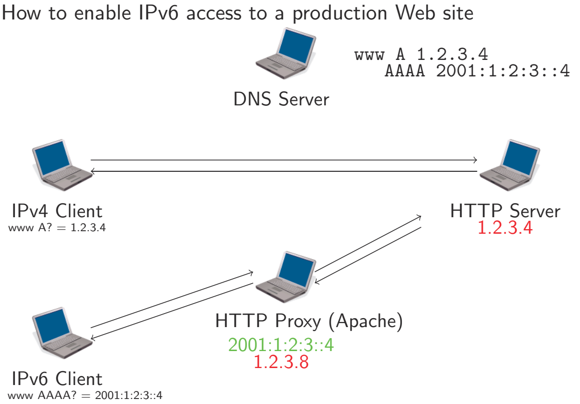
Migrate Application to IPv6 support

Some services are critical for sites (Mail, Web, ...)
IPv6 access to services may impact clients behavior
What to do if upgrade of application is not possible ?

Solutions for incremental deployment exist !
- Application Level Gateway
- SSL Tunnel

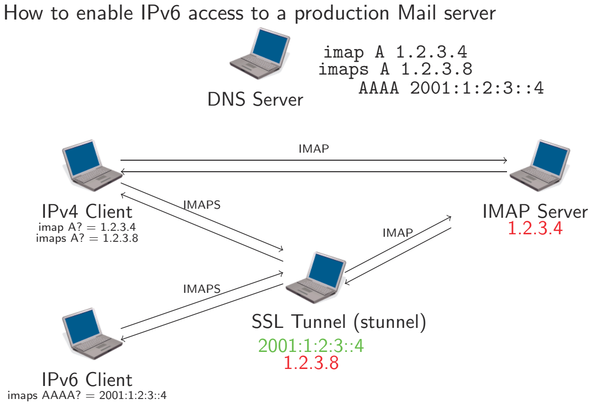
Application Level Gateway

How to enable IPv6 access to a production Web site



SSL Tunnel

How to enable IPv6 access to a production Mail server



Monitor IPv6 usage

Monitoring IPv6 is important for
- See impact of IPv6 deployement
- Ensure same Quality of Service in IPv4 an IPv6

Tools
- Traffic: MRTG/Cacti, Netflow v9
- Services: Nagios

Dual Stack requires dual check !

Need to check service reachability in IPv4 AND in IPv6

Interesting URLs

- G6 association: http://g6bis.point6.net
- IPv6 Task Force France: http://www.g6.asso.fr/tff/
- G6 livre: http://livre.point6.net/
- IPv6 Forum: http://www.ipv6forum.com/
- Ipv6 Task Force: http://www.ipv6tf.org/
- 6DISS (IST project) http://www.6diss.org/
- 6DISS formation slides: http://events.um.edu.mt/6diss/workshop.html
- IPv6 Security infos: http://www.stindustries.net/IPv6/