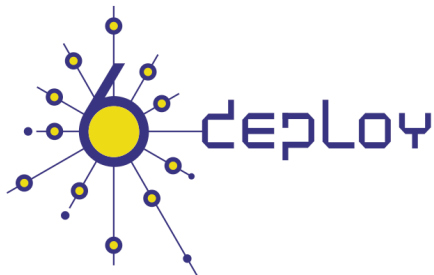# IPv6 Startup

## APNIC Workshop
## Paro, Bhutan
### July 2010

**Jordi Palet, Consulintel**

**(jordi.palet@consulintel.es)**

# Agenda

1. IPv6 Installation and Setup in Several Platforms (Windows 2000/XP/2003/Vista/7, Linux, BSD, Mac OS X)

2. Basic Configuration, Stateless/Stateful Autoconfiguration, Privacy, Static Routes

3. Transition Mechanisms Configuration

4. Examples of Applications

# Part 1

# IPv6 Setup in several Platforms (Windows 2000/XP/2003/Vista/7, Linux, BSD, Mac OS X)

# IPv6 in Windows

- Full Support
  - Windows 7, Vista, XP SP1 and later
  - Windows Server 2003 and 2008

- Technology preview
  - Windows XP without SP
  - Windows 2000 (not compatible with SP2 or later)

- Developer edition
  - Windows NT 4.0

- Third part products without official support
  - Windows 95/98/ME
  - Windows 2000 with SP2 and later

- Supported Characteristics (latest versions)
  - Autoconfiguration, 6in4 tunnels, 6to4 tunnels, 6to4 relay , TEREDO tunnels, ISATAP tunnels, IPSec (manual keys)

# IPv6 Setup: 2000 (1)

- Non-production stack available (originally developed by Microsoft Research)

- Download the "Microsoft IPv6 Technology Preview for Windows 2000":
  - `Available at http://www.ipv6tf.org/using/connectivity/guides.php?cid=1`
  - Note that Windows 2000 IPv6 isn't supported anymore by Microsoft

- Install Procedure:
  - Log on to the Windows 2000 with local administrator privileges
  - Extract IPv6 Technology Preview files, for example in C:\IPv6Kit
  - Follow the procedure in SPn & IE6 fixed.txt in order to change /setup/hotfix.ini file
  - Run the Setup.exe or hotfix.exe
  - From the Windows 2000 desktop, click Start, point to Settings, and then click Network and Dial-up Connections. As an alternative, you can right-click My Network Places, and then click Properties
  - Right-click the Ethernet-based connection to which you want to add the IPv6 protocol, and then click Properties (typically, this connection is named Local Area Connection
  - Click Install)
  - In the Select Network Component Type dialog box, click Protocol, and then click Add
  - In the Select Network Protocol dialog box, click Microsoft IPv6 Protocol and then click OK
  - Click Close to close the Local Area Connection Properties dialog box

- In a DOS Prompt:
  - **ipv6 if** to check if IPv6 has been installed
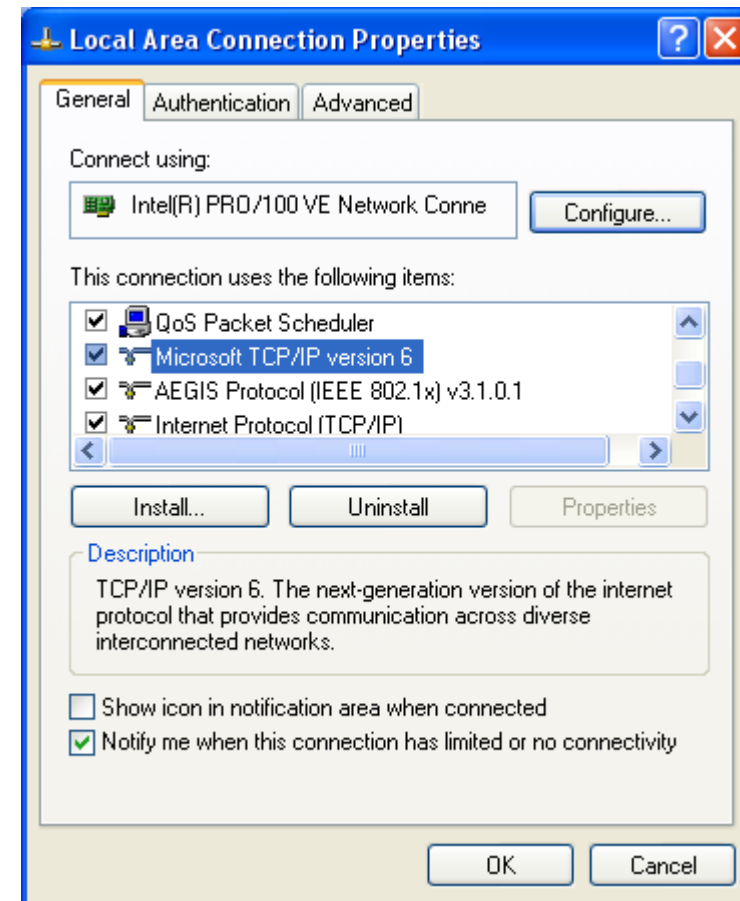
# IPv6 Setup: 2000 (2)

- Uninstall Procedure:
  - Log on to the Windows 2000 with local administrator privileges
  - From the Windows 2000 desktop, click Start, point to Settings, and then click Network and Dial-up Connections. As an alternative, you can right-click My Network Places, and then click Properties
  - Right-click the connection to which you want to remove the Microsoft Research IPv6 protocol, and then click Properties (typically, this connection is named Local Area Connection)
  - Click MSR IPv6 Protocol and then click Uninstall
  - In the Uninstall MSR IPv6 Protocol dialog box, click Yes
  - In the Local Network dialog box, click Yes to restart your computer

- In a DOS Prompt:
  - **ipv6 if** to check if IPv6 was uninstalled

# IPv6 Setup: XP/2003 (1)

- In a DOS Prompt:
  - **ipv6 install** to install IPv6 as Network Protocol

    - **ipconfig** or **ipv6 if** to check if IPv6 was installed

THE IPv6 PORTAL

Consul inTel
Consultores Integrales en Telecomunicaciones

# IPv6 Setup: XP/2003 (2)

- Another option to check if IPv6 was installed
  - Network Connections > Local Area Connection > Properties

- Also it is possible to install/uninstall IPv6 from here

# IPv6 Setup: XP/2003 (3)

In a Command Prompt:

- **ipv6 uninstall** to delete IPv6 as Network Protocol

  - **ipconfig** or **ipv6 if** to check if IPv6 was uninstalled

THE IPv6 PORTAL

Consulintel
Consultores Integrales en Telecomunicaciones

- 9

# IPv6 Setup: Vista (1)

- Nothing to do!!!
  - IPv6 is installed and enabled by default ☺
  - The configuration is based on GUI and netsh commands
- Other new features in Vista regarding IPv6
  - Full IPsec support
  - MLDv2
  - Link-Local Multicast Name Resolution (LLMNR)
    - It doesn't need DNS server. IPv6 nodes in a segment ask the name to a multicat IPv6 address. It's similar to the NetBIOS working.
  - IPv6 address in URLs support
  - IPv6 over PPP
  - DHCPv6, not only in the client but also in the server
  - Random IDs by default for the IPv6 address
    - Similar to the Privacy Extension Address but allows to be included in DNS
    - It prevents the user be tracked by using the EUI-64 part of the IPv6 address
  - TEREDO supports symetric NATs
    - It is enabled by default but no actvie. It becomes active automatically if any application needs IPv6 support and it is no natively available in the network

# IPv6 Setup: Vista (2)

- GUI Configuration
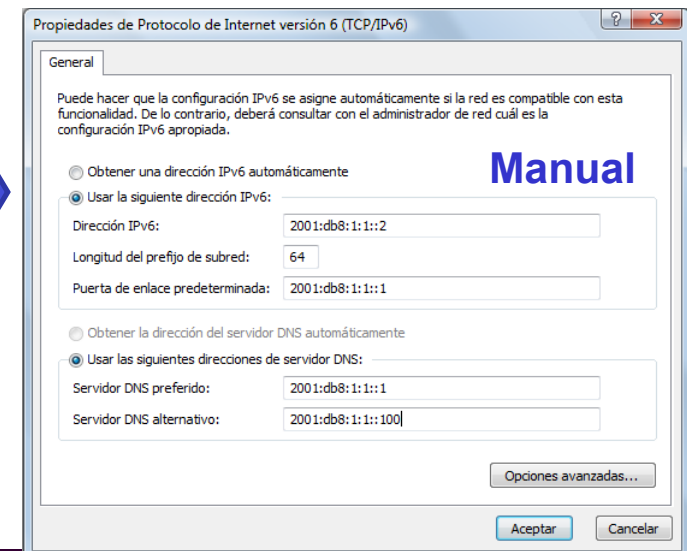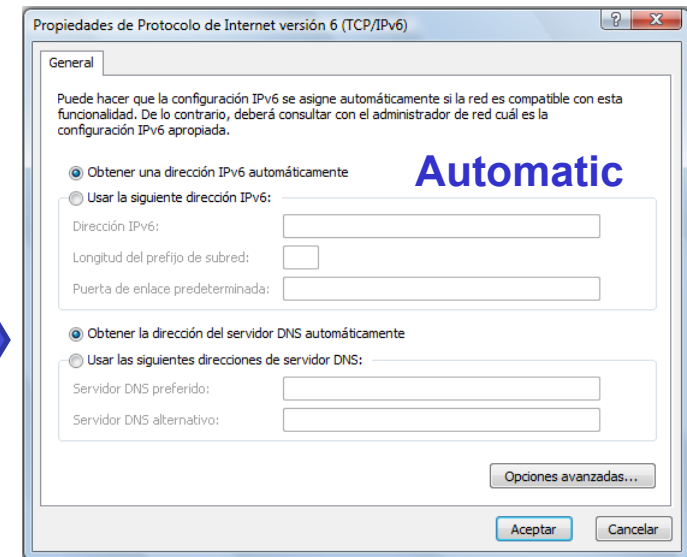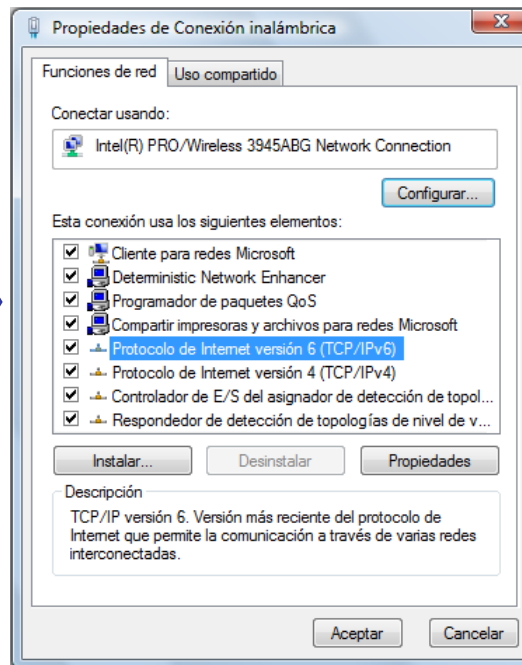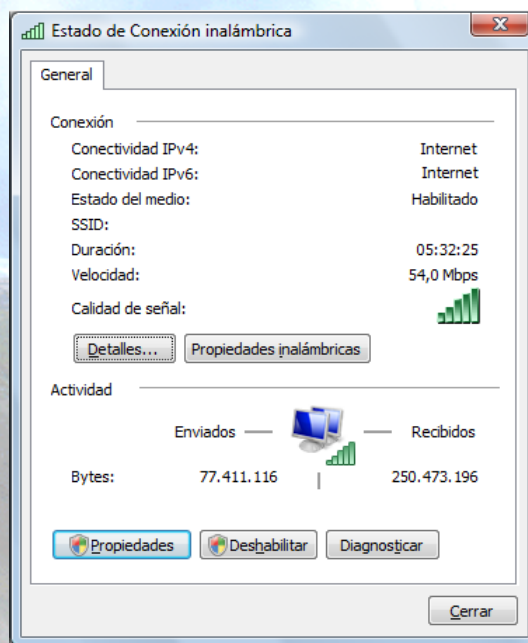
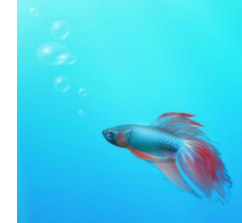Uncheck it to disable the protocol in the interface

# IPv6 Setup: Vista (3)

- Un-installation
  - It cannot be un-installed because the IPv6 stack is completely integrated in the operating system as the IPv4 one
- It can be disabled for one specific network interface
  - Through the GUI "Network Connections" and disabling the IPv6 stack
- It is possible the use of netsh commands. It needs DOS with administrator permissions
- Different IPv6 components can be customized through the registry:
  - The following record (type DWORD) needs to be created: HKEY_LOCAL_MACHINE\SYSTEM \CurrentControlSet\Services\tcpip6\Parameters\DisabledComponents
  - The DisabledComponents registry value is a bit mask that controls the following series of flags, starting with the low order bit (Bit 0 = enabled):
  - Bit 0 ➔ Set to 1 to disable all IPv6 tunnel interfaces, including ISATAP, 6to4, and TEREDO tunnels. Default value is 0
  - Bit 1➔ Set to 1 to disable all 6to4-based interfaces. Default value is 0
  - Bit 2 ➔ Set to 1 to disable all ISATAP-based interfaces. Default value is 0
  - Bit 3 ➔ Set to 1 to disable all Teredo-based interfaces. Default value is 0
  - Bit 4 ➔ Set to 1 to disable IPv6 over all non-tunnel interfaces, including LAN interfaces and Point-to-Point Protocol (PPP)-based interfaces. Default value is 0
  - Bit 5 ➔ Set to 1 to modify the default prefix policy table to prefer IPv4 to IPv6 when attempting connections. Default value is 0
  - Some values of **DisabledComponents** for disabling some components:

    | | |
    |---|---|
    | Disable all tunnel interfaces | 0x1 |
    | Disable 6to4 | 0x2 |
    | Disable ISATAP | 0x4 |
    | Disable TEREDO | 0x8 |
    | Disable TEREDO and 6to4 | 0xA |
    | Disable all LAN and PPP interfaces | 0x10 |
    | Disable all LAN, PPP, and tunnel interfaces | 0x11 |
    | Prefer IPv4 over IPv6 | 0x20 |
    | Disable IPv6 over all interfaces and prefer IPv4 to IPv6 | 0xFF |

THE IPv6 PORTAL

Consulintel
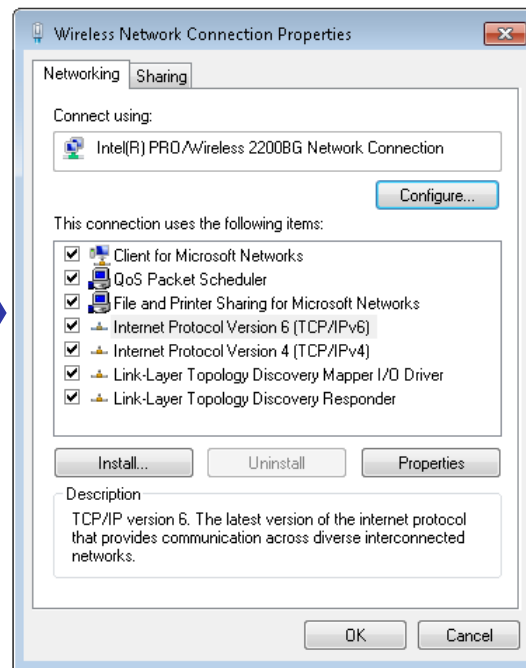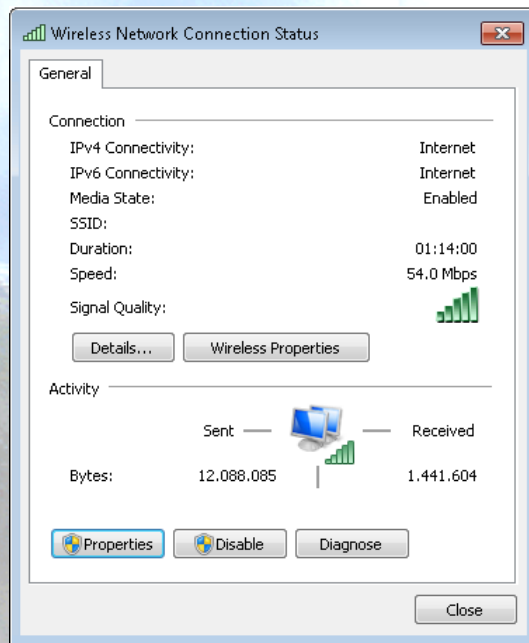Consultores Integrales en Telecomunicaciones

# IPv6 Setup: 7 RC (1)

- Windows 7 Ultimate – Version RC 6.1.7100
- Nothing to do!!!
  - IPv6 is installed and enabled by default ☺
  - The configuration is based on GUI and netsh commands
- IPv6 support similar to Vista and Server 2008
  - IPsec, MLDv2, LLMNR, IPv6 in URLs, IPV6CP, IPv6 over PPP, DHCPv6, TEREDO
  - Change: Random Interface ID is on by default (RFC 3041)
    - It doesn't use EUI-64 by default to get the interface ID in the autoconfigured addresses
      - netsh interface ipv6 set global [[randomizeidentifiers=]enabled|disabled]
- But with some new features
  - IP-HTTPS (IP over Secure HTTP)
    - It enables hosts located behind a proxy or firewall to establish connectivity by tunneling IP traffic inside an HTTPS tunnel. IP-HTTPS does not providing data security—you must use IPsec to provide data security for an IP-HTTPS connection. Further information: http://msdn.microsoft.com/en-us/library/dd358571.aspx
  - DirectAccess
    - It enables to remote users, who attempt to access corporate intranet resources, can access these resources without the need of having to connect to a virtual private network (VPN). Also, it enables administrators to keep the computers of mobile uses in a managed state even when they are off-site. DirectAccess can work over existing IPv4 networks such as the public IPv4 Internet by leveraging IPv4/v6 transition technologies such as 6to4, TEREDO and ISATAP.  It also supports native IPv6 connectivity for clients that have been assigned native IPv6 addresses.
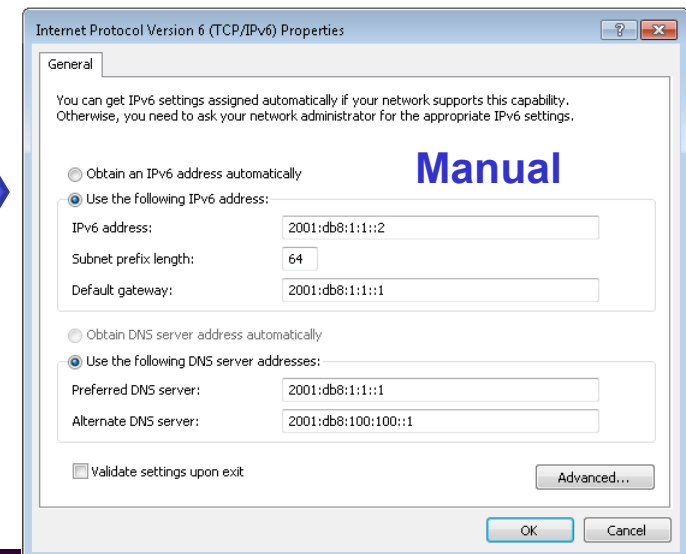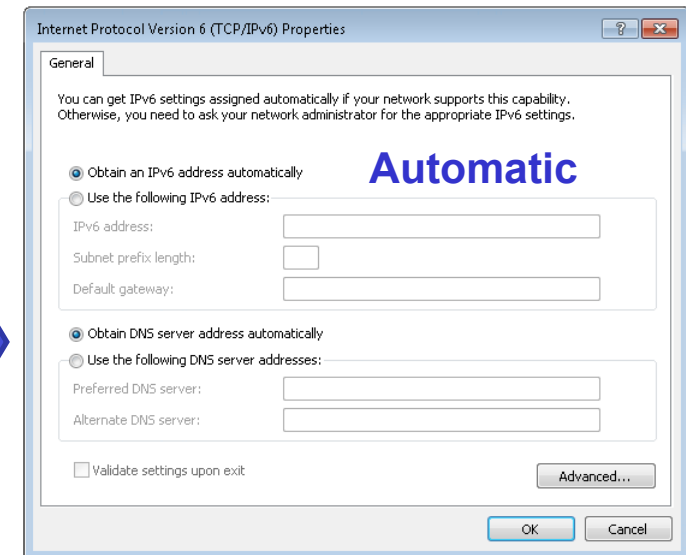    - The client can be Windows 7 or Server 2008. The server can be Server 2008

THE IPv6 PORTAL

ConsulinTel
Consultores Integrales en Telecomunicaciones

# IPv6 Setup: 7 RC (2)

- GUI Configuration



**Automatic**

**Manual**

Uncheck it to disable the protocol in the interface

# IPv6 Setup: 7 RC (3)

- Un-installation
  - It cannot be un-installed because the IPv6 stack is completely integrated in the operating system as the IPv4 one
- It can be disabled for one specific network interface
  - Through the GUI "Network Connections" and disabling the IPv6 stack
- It is possible the use of netsh commands. It needs DOS with administrator permissions
- As well as Vista, different IPv6 components can be customized through the registry
  - http://technet.microsoft.com/en-us/library/bb878057.aspx

# IPv6 Setup: Linux (1)

- Supported from kernel version 2.4.x

  #uname -r

- To check if IPv6 is installed:

  #test -f /proc/net/if_inet6 && echo "Current Kernel supports IPv6"

- Module Installation:

  #modprobe ipv6

- Module check:

  #lsmod |grep -w 'ipv6' && echo "IPv6 module loaded"

- Automatic Load/Unload of Module (/etc/modules.conf o /etc/conf.modules ):

  alias net-pf-10 ipv6 #enables load on demand

  alias net-pf-10 off  #disables load on demand

# IPv6 Setup: Linux (2)

**# ifconfig** to check

```
eth0  Link encap:Ethernet  HWaddr 00:E0:81:05:46:57
      inet addr:10.0.0.3  Bcast:10.0.0.255  Mask:255.255.255.0
      inet6 addr: fe80::2e0:81ff:fe05:4657/64 Scope:Link
      inet6 addr: 2001:800:40:2a05::3/64 Scope:Global
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:2010563 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1700527 errors:0 dropped:0 overruns:2 carrier:0
      collisions:0 txqueuelen:100
      RX bytes:205094215 (195.5 Mb)  TX bytes:247063610 (235.6Mb)
      Interrupt:11 Base address:0xe000 Memory:f8201000-f8201038
lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:1675838 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1675838 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:659846244 (629.2 Mb)  TX bytes:659846244 (629.2 Mb)
```

# IPv6 Setup: Linux (3)

## Persistent Configuration

- Red Hat (from 7.1) and similar "distros":

    Add in /etc/sysconfig/network:

    NETWORKING_IPV6=yes

    Network Restart:

    # service network restart

    Or

    #/etc/init.d/network restart

- SUSE:

    Add in /etc/sysconfig/network/ifcfg-<Interface-Name>:

    SUSE 8.0: IP6ADDR="<ipv6-address>/<prefix>"

    SUSE 8.1: IPADDR="<ipv6-address>/<prefix>"

# IPv6 Setup: Linux (4)

## Persistent Configuration

* Debian:

  Once the IPv6 module is loaded, then edit /etc/network/interfaces, for example:

  ```
  iface eth0 inet6 static
        pre-up modprobe ipv6
        address 3ffe:ffff:1234:5::1:1
        # unable autoconfiguration:
        # up echo 0 > /proc/sys/net/ipv6/conf/all/autoconf
        netmask 64
        # router is autoconfigured and doesn't have static address
        # it finds it because of
        # (/proc/sys/net/ipv6/conf/all/accept_ra).
        # if not, gateway must be configured:
        # gateway 3ffe:ffff:1234:5::1
  ```

  – Reboot or:
  ```
  # ifup --force eth0
  ```

# IPv6 Setup: Linux (5)

- Tools:

1. net-tools package

   # /sbin/ifconfig -? 2>& 1|grep -qw 'inet6' && echo "'ifconfig supports IPv6"

   # /sbin/route -? 2>& 1|grep -qw 'inet6' && echo "'route supports IPv6"

2. iproute package

   # /sbin/ip 2>&1 |grep -qw 'inet6' && echo "'ip supports IPv6"

3. iputils package contains ping6, traceroute6 and tracepath6

# IPv6 Setup: BSD (1)

- To install the Stack (Versions 4.5+)
- Good IPv6 support

# IPv6 Setup: Mac OS X (1)

- IPv6 support from 10.2 Jaguar

- Enabled by default

- From 10.3 Panther is possible to enable/disable and configure IPv6 using a GUI. System Prefenerces > Network
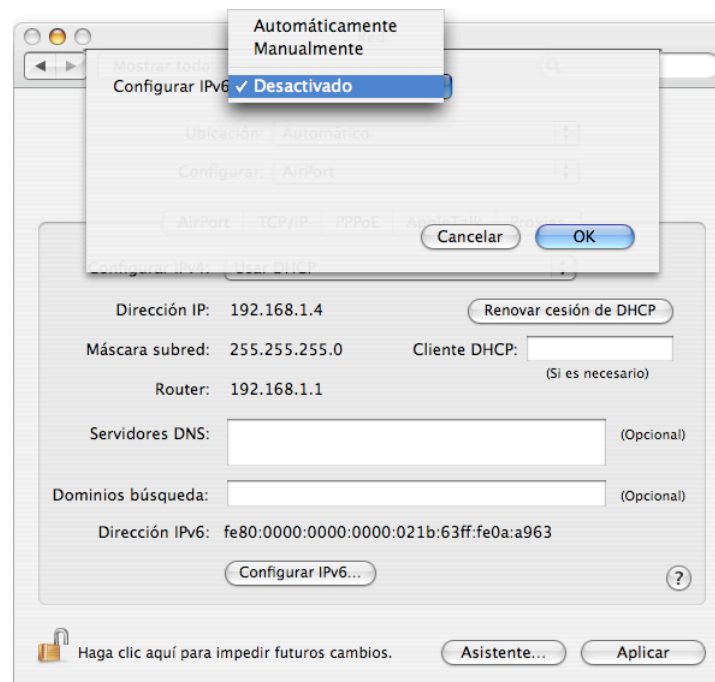
- Configuring IPv6
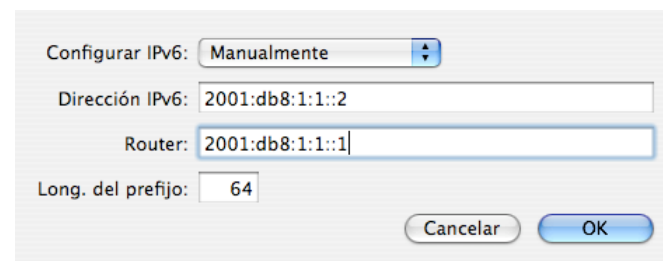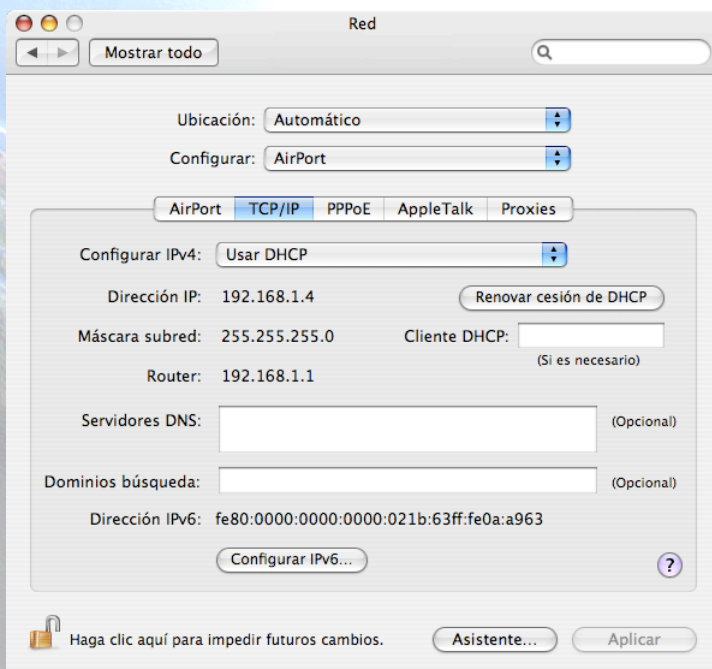
# IPv6 Setup: Mac OS X (2)

- Enable/disable IPv4
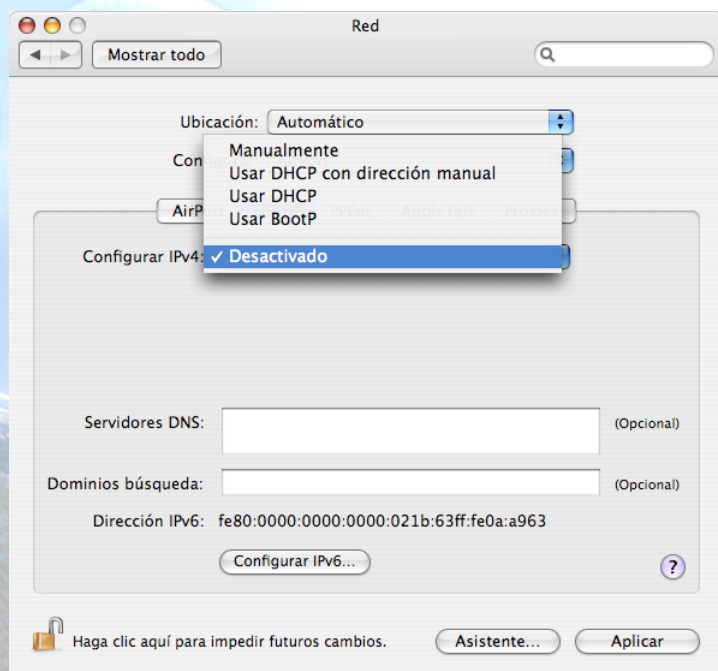
- Enable/disable IPv6



- Enable/disable IPv6 in all interfaces
  - Enable # ip6 –a
  - Disable # ip6 -x

# Part 2

# Basic Configuration Stateless/Stateful Autoconfiguration, Privacy, Static Routes

# Basic Configuration: Windows (1)

- Basic Configuration in Windows

- Useful to obtain information about the status and to configure interfaces, addresses, caches, routes, and so on

- Two groups of commands:
  - **ipv6.exe** (covers up to Windows XP SP1)
    - Some changes are not persistent (values lost with each reboot). It is possible to execute a configuration in a script in each boot.
  - **netsh interface ipv6** (for Windows XP SP2 and later, 2003,2008, Vista and 7)
    - Option **store=active|persistent** to save changes

- Equivalences at:

  http://technet.microsoft.com/en-us/library/bb726950.aspx

# Basic Configuration: Windows (2)

- **"ipv6" Commands**
- **Some of them needs administration permissions**
  - ipv6 [-p] [-v] if [ifindex]
  - ipv6 [-p] ifcr v6v4 v4src v4dst [nd] [pmld]
  - ipv6 [-p] ifcr 6over4 v4src
  - ipv6 [-p] ifc ifindex [forwards] [-forwards] [advertises] [-advertises] [mtu #bytes] [site site-identifier] [preference P]
  - ipv6 rlu ifindex v4dst
  - ipv6 [-p] ifd ifindex
  - ipv6 [-p] adu ifindex/address [life validlifetime[/preflifetime]] [anycast] [unicast]
  - ipv6 nc [ifindex [address]]
  - ipv6 ncf [ifindex [address]]
  - ipv6 rc [ifindex address]
  - ipv6 rcf [ifindex [address]]
  - ipv6 bc
  - ipv6 [-p] [-v] rt
  - ipv6 [-p] rtu prefix ifindex[/address] [life valid[/pref]] [preference P]  [publish] [age] [spl SitePrefixLength]
  - ipv6 spt
  - ipv6 spu prefix ifindex [life L]
  - ipv6 [-p] gp
  - ipv6 [-p] gpu [parameter value] ... (try -?)
  - ipv6 renew [ifindex]
  - ipv6 [-p] ppt
  - ipv6 [-p] ppu prefix precedence P srclabel SL [dstlabel DL]
  - ipv6 [-p] ppd prefix
  - ipv6 [-p] reset
  - ipv6 install
  - ipv6 uninstall

# Basic Configuration: Windows (3)

- **"netsh interface ipv6" Commands**
- **Some of them needs administration permissions**
  - 6to4        - Changes to the 'netsh interface ipv6 6to4' context
  - ?             - Displays a list of commands
  - add         - Adds a configuration entry to a table
  - delete      - Deletes a configuration entry from a table
  - dump      - Displays a configuration script
  - help       - Displays a list of commands
  - isatap      - Changes to the 'netsh interface ipv6 isatap' context
  - reset      - Resets IPv6 configuration state
  - set         - Sets configuration information
  - show      - Displays information
  - install     - Installs IPv6 (not available in Vista and 7)
  - uninstall    - Uninstalls IPv6 (not available in Vista and 7)
  - renew      - Restarts IPv6 interfaces (not available in Vista and 7)

# Basic Configuration: Windows (4)

- **"netsh interface ipv6 add" Commands**
  - add address    - Adds an IPv6 address on an interface.
  - add dnsserver  - Adds a static DNS server address
  - add neighbors – Add a neighor address
  - add potentialrouter – Add a router in the list fo possibles routers
  - add prefixpolicy - Adds a prefix policy entry.
  - add route      - Adds an IPv6 route over an interface.
  - add v6v4tunnel - Creates an IPv6-in-IPv4 point-to-point tunnel
  - add 6over4tunnel - Creates a 6over4 interface (not available in Vista and 7)

- **"netsh interface ipv6 set" Commands**
  - set address    - Modifies IPv6 address information
  - set global     - Modifies global configuration general parameters
  - set interface  - Modifies interface configuration parameters
  - set prefixpolicy - Modifies prefix policy information
  - set privacy    - Modifies privacy configuration parameters
  - set route      - Modifies route parameters
  - set teredo     - Sets TEREDO state
  - set mobility   - Modifies mobility configuration parameters (not available in Vista and 7)
  - set state      - Sets the state of deprecated functionality (not available in Vista and 7)

# Basic Configuration: Windows (5)

- **"netsh interface ipv6 show" Commands**
    - show address   - Shows IPv6 addresses
    - show destinationcache - Shows destination cache entries
    - show dnsservers      - Displays the DNS server addresses
    - show global    - Shows global configuration parameters
    - show interface - Shows interface parameters
    - show joins     - Shows IPv6 multicast addresses
    - show neighbors - Shows neighbor cache entries
    - show prefixpolicy - Shows prefix policy entries
    - show privacy   - Shows privacy configuration parameters
    - show route     - Shows route table entries
    - show siteprefixes - Shows site prefix table entries
    - show teredo    - Shows TEREDO service state
    - show bindingcacheentries - Shows binding cache entries (not available in Vista and 7)
    - show mobility  - Shows mobility configuration parameters (not available in Vista and 7)
    - show state     - Shows the state of deprecated functionality (not available in Vista and 7)
    - show routes    - Shows route table entries (not available in Vista and 7 – changed for show route)

# Basic Configuration: Windows (6)

- Interface Information
- **ipconfig [/all]**
- **ipv6 [-v] if [IfIndex]**
- Example: ipv6 if 5

```
Interface 5: Ethernet: Local Area Connection
  Guid {F5149413-6E54-4FDA-87BD-24067735E363}
  uses Neighbor Discovery
  uses Router Discovery
  link-layer address: 00-01-4a-18-26-c7
    preferred global 2001:db8::2, life infinite (manual)
    preferred global 2001:db8::4, life infinite (manual)
    preferred global 2001:db8::fde7:a76f:62d5:3bb9, life 6d21h3m20s/21h33s (temporary)
    preferred global 2001:db8::201:4aff:fe18:26c7, life 29d23h51m39s/6d23h51m39s (public)
    preferred link-local fe80::201:4aff:fe18:26c7, life infinite
    multicast interface-local ff01::1, 1 refs, not reportable
    multicast link-local ff02::1, 1 refs, not reportable
    multicast link-local ff02::1:ff18:26c7, 2 refs, last reporter
    multicast link-local ff02::1:ffd5:3bb9, 1 refs, last reporter
    multicast link-local ff02::1:ff00:4, 1 refs, last reporter
    multicast link-local ff02::1:ff00:2, 1 refs, last reporter
  link MTU 1500 (true link MTU 1500)
  current hop limit 64
  reachable time 29000ms (base 30000ms)
  retransmission interval 1000ms
  DAD transmits 1
  default site prefix length 48
```

# Basic Configuration: Windows (7)

- Ping up to XP SP1
  - **ping6** [-t] [-a] [-n count] [-l size] [-w timeout] [-s srcaddr] [-r] destination


- Ping
  - **ping** [-t] [-a] [-n count] [-l sixe] [-f] [-i TTL] [-v TOS] [-r count] [-s count] [[-j lista-host] | [-k host-list]] [-w waitng-time] [-R] [-S srcaddr] [-4] [-6] destination
    - **-4      forces the use of IPv4**
    - **-6    forces the use of IPv6**
    - **ping command default to IPv6 if available**

# Basic Configuration: Windows (8)

- **Examples of Ping**

- **ping6 www.ipv6tf.org**

Pinging www.ipv6tf.org [2a01:48:1:0:2e0:81ff:fe05:4658]

from 2a01:48:1:0:9c4d:b1cd:98d5:5a32 with 32 bytes of data:

Reply from 2a01:48:1:0:2e0:81ff:fe05:4658 : bytes=32 time<1ms

Reply from 2a01:48:1:0:2e0:81ff:fe05:4658 : bytes=32 time<1ms

Reply from 2a01:48:1:0:2e0:81ff:fe05:4658 : bytes=32 time<1ms

Reply from 2a01:48:1:0:2e0:81ff:fe05:4658 : bytes=32 time<1ms

Ping statistics for 2a01:48:1:0:2e0:81ff:fe05:4658 :

    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

    Minimum = 0ms, Maximum = 0ms, Average = 0ms

# Basic Configuration: Windows (9)

- **Examples of Ping**

- **ping ::1**

  Pinging ::1 from ::1 with 32 bytes of data:
  Reply from ::1: bytes=32 time<1ms
  Reply from ::1: bytes=32 time<1ms
  Reply from ::1: bytes=32 time<1ms
  Reply from ::1: bytes=32 time<1ms
  Ping statistics for ::1:
      Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
      Minimum = 0ms, Maximum = 0ms, Average = 0ms

- **ping6 fe80::201:4aff:fe18:26c7 (own link-local)**

  Pinging fe80::201:4aff:fe18:26c7 from fe80::201:4aff:fe18:26c7%5 with 32 bytes of data:
  Reply from fe80::201:4aff:fe18:26c7%5: bytes=32 time<1ms
  Reply from fe80::201:4aff:fe18:26c7%5: bytes=32 time<1ms
  Reply from fe80::201:4aff:fe18:26c7%5: bytes=32 time<1ms
  Reply from fe80::201:4aff:fe18:26c7%5: bytes=32 time<1ms
  Ping statistics for fe80::201:4aff:fe18:26c7:
      Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
      Minimum = 0ms, Maximum = 0ms, Average = 0ms

# Basic Configuration: Windows (10)

- Which are my neighbors?
  - **netsh interface ipv6 show neighbors**

    ...

    Interface 5: Local Area Connection

    | Internet Address | Physical Address | Type |
    | --- | --- | --- |
    | fe80::201:4aff:fe18:26c7 | 00-01-4a-18-26-c7 | Permanent |
    | fe80::200:87ff:fe28:a0e0 | 00-00-87-28-a0-e0 | Stale (router) |
    | 2001:db8::201:4aff:fe18:26c7 | 00-01-4a-18-26-c7 | Permanent |
    | 2001:db8::fde7:a76f:62d5:3bb9 | 00-01-4a-18-26-c7 | Permanent |
    | 2001:db8::2a03::3 | 00-e0-81-05-46-57 | Stale |
    | 2001:db8::1 | 00-00-87-28-a0-e0 | Stale |
    | 2001:db8::2 | 00-01-4a-18-26-c7 | Permanent |
    | 2001:db8::4 | 00-01-4a-18-26-c7 | Permanent |

- The reference to specific interface is done with "%"
  - **%5 is about interface 5**

# Basic Configuration: Windows (11)

- **Examples of Ping**

- **ping fe80::200:87ff:fe28:a0e0%5 (link-local neighbor in interface 5)**

  Pinging fe80::200:87ff:fe28:a0e0%5 from fe80::201:4aff:fe18:26c7%5 with 32 bytes of data:

  Reply from fe80::200:87ff:fe28:a0e0%5: bytes=32 time<1ms

  Reply from fe80::200:87ff:fe28:a0e0%5: bytes=32 time<1ms

  Reply from fe80::200:87ff:fe28:a0e0%5: bytes=32 time<1ms

  Reply from fe80::200:87ff:fe28:a0e0%5: bytes=32 time<1ms

  Ping statistics for fe80::200:87ff:fe28:a0e0%5:

  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

  Approximate round trip times in milli-seconds:

  Minimum = 0ms, Maximum = 0ms, Average = 0ms

THE IPv6 PORTAL

Consulintel
Consultores Integrales en Telecomunicaciones

# Basic Configuration: Windows (12)

- Traceroute up to XP SP1
  - **tracert6** [-d] [-h maximum_hops] [-w timeout] [-s srcaddr] target_name


- Traceroute
  - **tracert** [-d] [-h maximum_hops] [-j hosts-list] [-w  waiting time] [-R] [-S srcaddr] [-4] [-6] target_name
    - **-4      Forzar el uso de IPv4**
    - **-6      Forzar el uso de IPv6**
    - **tracert** command defaults to IPv6 when available

# Basic Configuration: Windows (13)

- **Examples of traceroute**
- **tracert www.lacnic.net**

  Tracing route to lacnic.net [2001:12ff:0:2::15] over a maximum of 30 hops:

  1    1 ms    <1 ms    <1 ms  gr2000-00.consulintel.euro6ix.org [2001:800:40:2a05::1]
  2   <1 ms     *        1 ms  2001:800:40:2f02::1
  3    4 ms    1 ms     1 ms  2001:800:40:2f01::2
  4   10 ms     4 ms     4 ms  data-to-tid.tid.euro6ix.org [2001:800:40:2f1a::2]
  5  200 ms   189 ms   189 ms  3ffe:80a::1
  6  388 ms   390 ms   388 ms  v6gw.isc.registro.br [2001:4f8:0:1::10:2]
  7  396 ms   396 ms   387 ms  lacnic.net [2001:12ff:0:2::15]

  Trace complete.

# Basic Configuration: Windows (14)

- Adding an Address:
- **netsh interface ipv6 add address** InterfaceNameOrIndex IPv6Address [[type=] unicast|anycast] [[validlifetime=]Minutes|infinite] [[preferredlifetime=]Minutes|infinite] [[store=] active|persistent]
- Example: netsh interface ipv6 add address 5 2001:db8::2 type=unicast validlifetime=infinite preferredlifetime=10m store=active
- Check the configuration using **netsh interface ipv6 show address 5**

# Basic Configuration: Windows (15)

- Modifying the options in an already configured address:
- **netsh interface ipv6 set address** [interface=] <string> [address=]<IPv6 address> [[type=] unicast|anycast] [[validlifetime=]<integer>| infinite] [[preferredlifetime=]<integer>|infinite] [[store=]active|persistent]
- Example: netsh interface ipv6 set address 5 2001:db8::2 preferredlifetime=infinite
- Check the configuration using **netsh interface ipv6 show address 5**

# Basic Configuration: Windows (16)

- Deleting an Address:
- **netsh interface ipv6 delete address** [interface=]<string> [address=]<IPv6 address> [[store=]active|persistent]
- Example: netsh interface ipv6 delete address 5 2001:db8::2 store=persistent
- To check the configuration using **netsh interface ipv6 show address 5**

# Basic Configuration: Windows (17)

- Adding a Static Route:

- **netsh interface ipv6 add route [prefix=]IPv6Address/Integer** [[interface=]String] [[nexthop=]IPv6Address] [[siteprefixlength=]Integer] [[metric=]Integer] [[publish=]{no | yes | immortal}] [[validlifetime=]{Integer | infinite}] [[preferredlifetime=]{Integer | infinite}] [[store=]{active | persistent}]

- Example: netsh interface ipv6 add route 2002::/16 5 fe80::200:87ff:fe28:a0e0 store=persistent

- Above, fe80::200:87ff:fe28:a0e0 is the default gateway

# Basic Configuration: Windows (18)

- Showing Routes:
- **netsh interface ipv6 show routes**
[[level=]{normal | verbose}]
[[store=]{active | persistent}]

- Example: netsh interface ipv6 show routes

```
Querying active state...
Publish  Type       Met  Prefix                          Idx  Gateway/Interface Name
-------  --------  ----  ------------------------------  ---  ---------------------
no       Manual       0  2002::/16                         5  fe80::200:87ff:fe28:a0e0
no       Autoconf     8  2001:db8::/64                     5  Local Area Connection
no       Autoconf   256  ::/0                              5  fe80::200:87ff:fe28:a0e0
```

# Basic Configuration: Windows (19)

- Deleting a Static Route:

- **netsh interface ipv6 delete route [prefix=]
  <IPv6 address>/<integer> [interface=]
  <string> [[nexthop=]<IPv6 address>] [[store=]
  active|persistent]**

- Example: netsh interface ipv6 delete route
  2002::/16 5 fe80::200:87ff:fe28:a0e0
  store=persistent

- Check using **netsh interface ipv6 show routes**

# Basic Configuration: Windows (20)

- Adding a Static DNS Server:
- **netsh interface ipv6 add dnsserver**
  [[interface=]String]
  [[address=]IPv6Address]
  [[index=]Integer]
- In XP SP1 is used **dns** instead of **dnsserver**

- Example: netsh interface ipv6 add dnsserver
  "Local area network" 2001:db8:1000:1::947c 1
- The "index" represent the position of the DNS server just configured in the DNS servers lists

# Basic Configuration: Windows (21)

- Showing DNS servers:
- **netsh interface ipv6 show dnsservers** [[interface=]string]

- Example: netsh interface ipv6 show dnsservers

```
DNS servers in LAN interface
Index           DNS server
------   ------------------------------------------
1               2001:db8:1000:1::947c
2               2001:db8:1000:1::947c
```

# Basic Configuration: Windows (22)

- Deleting a Static DNS server:
- **netsh interface ipv6 delete dnsserver** [interface=]<string> [[address=]<IPv6 address>| all]

- Example: netsh interface ipv6 delete dnsserver "Local area network" all
- Check using **netsh interface ipv6 show dnsservers**

# Basic Configuration: Linux (1)

## Basic Commands (1)

– **ifconfig**

– **ping6** <hostname>|<ipv6address>|[-I <interface>] <link-local-ipv6address>

– **traceroute6** <hostname>|<ipv6address>

– **tracepath6** <hostname>|<ipv6address>

– **tcpdump**

# Basic Configuration: Linux (2)

```
# ping6 ::1
PING ::1(::1) 56 data bytes
64 bytes from ::1: icmp_seq=1 ttl=64 time=0.047 ms
64 bytes from ::1: icmp_seq=2 ttl=64 time=0.039 ms
64 bytes from ::1: icmp_seq=3 ttl=64 time=0.042 ms
64 bytes from ::1: icmp_seq=4 ttl=64 time=0.020 ms
--- ::1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.020/0.037/0.047/0.010 ms


# ping6 -I eth0 fe80::2e0:81ff:fe05:4657
PING fe80::2e0:81ff:fe05:4657(fe80::2e0:81ff:fe05:4657) from ::1 eth0: 56 data
    bytes
64 bytes from fe80::2e0:81ff:fe05:4657: icmp_seq=1 ttl=64 time=0.056 ms
64 bytes from fe80::2e0:81ff:fe05:4657: icmp_seq=2 ttl=64 time=0.055 ms
64 bytes from fe80::2e0:81ff:fe05:4657: icmp_seq=3 ttl=64 time=0.048 ms
64 bytes from fe80::2e0:81ff:fe05:4657: icmp_seq=4 ttl=64 time=0.128 ms
--- fe80::2e0:81ff:fe05:4657 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.048/0.071/0.128/0.034 ms
```

THE IPv6 PORTAL

ConsulInTel
Consultores Integrales en Telecomunicaciones

# Basic Configuration: Linux (3)

## Basic Commands (2)

- ## Adding an Address

# /sbin/ip -6 addr add <ipv6address>/<prefixlength> dev <interface>

# /sbin/ifconfig <interface> inet6 add <ipv6address>/<prefixlength>

- ## Deleting an Address

# /sbin/ip -6 addr del <ipv6address>/<prefixlength> dev <interface>

# /sbin/ifconfig <interface> inet6 del <ipv6address>/<prefixlength>

Some commands need root permissions

# Basic Configuration: Linux (4)

## Static Routes

- ## Showing Routes

# /sbin/ip -6 route show [dev <device>]

# /sbin/route -A inet6

- ## Adding a Default Route via a Gateway

# /sbin/ip -6 route add <ipv6network>/<prefixlength> via <ipv6address>
  [dev <device>]

#/sbin/route -A inet6 add <ipv6network>/<prefixlength> gw <ipv6address>
  [dev <device>]

# Basic Configuration: Linux (5)

- ## Deleting a Default Route via a Gateway

# /sbin/ip -6 route del <ipv6network>/<prefixlength> via <ipv6address> [dev <device>]

# /sbin/route -A inet6 del <network>/<prefixlength> [dev <device>]

- ## Adding a Route via an interface

# /sbin/ip -6 route add <ipv6network>/<prefixlength> dev <device> metric 1

# /sbin/route -A inet6 add <network>/<prefixlength> dev <device>

# Basic Configuration: Linux (6)

- ## Deleting a Route via an interface

# /sbin/ip -6 route del <ipv6network>/<prefixlength> dev <device>

# /sbin/route -A inet6 del <network>/<prefixlength> dev <device>

- ## Showing Neighbors Table

# ip -6 neigh show [dev <device>]

- ## Adding a Neighbor

# ip -6 neigh add <IPv6 address> lladdr <link-layer address> dev <device>

- ## Deleting a Neighbor

# ip -6 neigh del <IPv6 address> lladdr <link-layer address> dev <device>

# Basic Configuration: BSD (1)

## Basic Commands

- Adding an Address

#>ifconfig <interface> inet6 add <dir. IPv6>

- Deleting an Address

#>ifconfig <interface> inet6 del <dir. IPv6>

# Basic Configuration: BSD (2)

- **Persistent Configuration:**

  Edit file /etc/rc.conf:

  ipv6_enable="YES"

  ipv6_ifconfig_rl0="2001:db8:10:4::4 prefixlen 64"

  In /etc/defaults/rc.conf you can find the different parameters to configure and the defaults values

- To make apply changes in rc.conf you must reboot

# Basic Configuration: BSD (3)

- **Static Routes**

- Adding a Default Route

#>route –n add -inet6 default <dir. IPv6>

- Deleting a Default Route

#>route –n del -inet6 default

# Basic Configuration: Mac OS X (1)

## Basic Commands (1)

- **ifconfig**, **ifconfig en1**
- **ping6** [-dfHnNqtvwW] [-P policy] [-a [aAclsg]] [-b sockbufsiz] [-c count] [-I interface] [-i wait] [-l preload] [-p pattern] [-S sourceaddr] [-s packetsize] [-h hoplimit] [hops...] host
- **traceroute6** [-dlnrv] [-f firsthop] [-g gateway] [-m hoplimit] [-p port] [-q probes] [-s src] [-w waittime] target [datalen]
- **tcpdump**

# Basic Configuration: Mac OS X (2)

- **$ ping6 ::1**

PING6(56=40+8+8 bytes) ::1 --> ::1

16 bytes from ::1, icmp_seq=0 hlim=64 time=0.101 ms

16 bytes from ::1, icmp_seq=1 hlim=64 time=0.117 ms

16 bytes from ::1, icmp_seq=2 hlim=64 time=0.117 ms

…

- **$ ping6 -I en1 fe80::21b:63ff:fe0a:a963**

PING6(56=40+8+8 bytes) fe80::21b:63ff:fe0a:a963%en1 --> fe80::21b:63ff:fe0a:a963

16 bytes from fe80::21b:63ff:fe0a:a963%en1, icmp_seq=0 hlim=64 time=0.082 ms

16 bytes from fe80::21b:63ff:fe0a:a963%en1, icmp_seq=1 hlim=64 time=0.117 ms

16 bytes from fe80::21b:63ff:fe0a:a963%en1, icmp_seq=2 hlim=64 time=0.148 ms

…

# Basic Configuration: Mac OS X (3)

## Basic Commands(2)

- Adding an Address

$ sudo ifconfig <interface> inet6 2001:db8:1:1::2/64

- Deleting an Address

$ sudo ifconfig <interface> inet6 delete 2001:db8:1:1::2

- Show Configuration

$ ifconfig | grep inet6

$ ifconfig <interface> | grep inet6

# Basic Configuration: Mac OS X (4)

## Static Routes

- ## Adding a Default Route

$ sudo route add -inet6 default [2001:db8:1:1::1, -interface en1]

- ## Deleting a Default Route

$ sudo route delete -inet6 default [2001:db8:1:1::1, -interface en1]

- ## Show IPv6 Routes

$ netstat -r -f inet6

# Basic Configuration: Exercise 1

- **ping6** to link-local Address of a Neighbor

- At the same time, capture packets using tcpdump:

    # tcpdump -t -n -i eth0 -s 512 -vv ip6 or proto ipv6

- Another way to show addresses:

    # /sbin/ip -6 addr show dev eth2
    # ifconfig eth0

- Add and delete the address:

    2001:db8:40:2a09:1:2:3:4 in the eth0 interface

# Basic Configuration: Exercise 2

## Linux

- Add and delete a route through a gateway
- Add and delete a route through an interface
- Show neighbors table
- Add and delete a neighbor

## BSD

- Add and delete a route through a gateway

# Stateless Autoconfiguration (1)

- RFC 2462: IPv6 Stateless Address Autoconfiguration
- Provides information about
  - Network Prefix
  - Routing
- Global Addresses are built by two elements

| Router | Host |
|---|---|
| **Prefix subnet** | **Interface ID** |
| 64 bits | 64 bits |
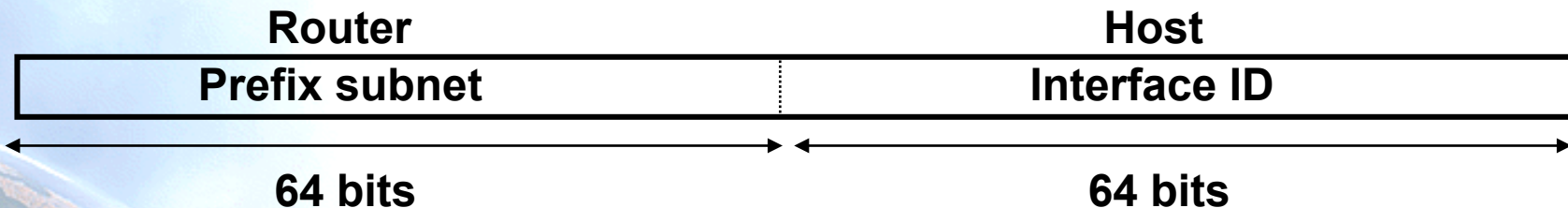
  - Interface Identifier (64 bits based on EUI-64, and usually obtained from IEEE 48 bit MAC Address)
  - Prefix obtained from de Prefix Information Options contain in the Router Advertisements
- Easing the Configuration
  - The user does not need to configure any network parameter in order to obtain native IPv6 connectivity

# Stateless Autoconfiguration (2)

- In Windows hosts, it is enabled by default

- Check which is the autoconfigured address
  - **ipconfig** or **ipv6 if** or **netsh interface ipv6 show addresses**

- Example: **2001:db8:10:10**:201:4aff:fe18:26c7
  - Interface Identifier EUI-64 obtained from this MAC address: 4aff:fe18:26c7
  - Prefix provided by the router: **2001:db8:10:10**

# Stateless: Exercise 1

- Configure a Windows router to send RA packets to the network
  - netsh interface ipv6 set interface <num_interfaz> advertise=enabled
  - netsh interface ipv6 set interface <num_interfaz> forwarding=enabled
  - netsh interface ipv6 set route 2001:db8:a:1::/64 <num_interfaz> publish=yes

- Check that other computers in the network are autoconfigured thanks to Ras
  - **ipconfig**, **ipv6 if** o **netsh interface ipv6 show addresses**

THE IPv6 PORTAL

**Consul** **InTel**
Consultores Integrales en Telecomunicaciones

# Stateless: Exercise 2 (1)

- Configure a Linux router to send RA packets to the network
- Get a 'radvd' daemon for the used Linux distribution
  - Fedora, Mandriva, RH
    - http://www.rpmfind.net/linux/rpm2html/search.php?query=radvd&submit=Search+...
  - Ubuntu
    - sudo apt-get install radvd

# Stateless: Exercise 2 (2)

- Edit /etc/radvd.conf file with the following content

```
interface eth00
{
        AdvSendAdvert on;

        MinRtrAdvInterval 3;
        MaxRtrAdvInterval 5;

        AdvHomeAgentFlag off;

        prefix 2001:db8:1234:5678::/64
        {
          AdvOnLink off;
          AdvAutonomous on;
          AdvRouterAddr off;
        };
};
```

# Stateless: Exercise 2 (3)

- Enable routing capability
  - echo 1 > /proc/sys/net/ipv6/conf/all/forwarding

- Launch radvd daemon

  - sudo radvd [-hv]

- Check that other computers in the network are autoconfigured thanks to our radvd daemon
  - Ifconfig
  - sudo tcpdump -t -n -i eth0 -vv ip6 or proto ipv6

THE IPv6 PORTAL

ConsulinTel
Consultores Integrales en Telecomunicaciones

# Stateful Autoconfiguration (1)

- RFC3315 DHCPv6
- Similar to DHCP in IPv4
- An IPv6 address is provided. This address can be different each time a node gets connected
- Provides information complementary to the stateless one
  - DNS Server (could be IPv6)
  - domain name
  - NTP server (could be IPv6)
  - SIP server  (could be IPv6)
  - SIP domain name
  - Prefix delegation
  - Etc.
- DHCPv6 implementations are available
  - Software for the main Operating Systems for host
  - Dedicated appliances
- Is needed the instalation of DHCPv6 server and/or client
  - http://www.ipv6-to-standard.org

# Stateful: Exercise 1 (1)

- Configuring a DHCPv6 Server on Linux
  - Get the DHCPv6 implementation for Linux (Dibbler)
    - • http://klub.com.pl/dhcpv6/dibbler/
    - • sudo apt-get install dibbler-server [dibbler-client, dibbler-relay]
  - Make sure these directories exist
    - /var/lib/dibbler
    - /etc/dibbler

# Stateful: Exercise 1 (2)

- Edit the file /etc/dibbler/server.conf

  ```
  log-level 7
  log-mode short

  iface "eth0" {
   T1 1000
   T2 2000
   class {
  pool 2001:db8:1234:5678::10- 2001:db8:1234:5678 :ffff:ffff:ffff:ffff
   }

   option dns-server 2001:db8:1::2,2001:db8:1::4
   option domain example.com, test1.example.com

   }
  ```

- The given addresses will be in the prefix 2001:db8:1234:5678 ::/64 starting from 2001:db8:1234:5678 ::10

- Launch dhcpv6 server
  - dhcpv6_server run o sudo dibbler-server run [start, stop, status, help]

- Check
  - /var/log/dibbler

# Stateful: Exercise 2 (1)

- Configure DHCPv6 Client in Linux
  - Get a DHCPv6 implementation for Linux (Dibbler)
    - http://klub.com.pl/dhcpv6/dibbler/
    - sudo apt-get install dibbler-client [dibbler-server, dibbler-relay]
  - Make sure these directories exist
    - /var/lib/dibbler
    - /etc/dibbler

# Stateful: Exercise 2 (2)

- Edit the file /etc/dibbler/client.conf

  log-mode short

  iface eth0
  {

  IA
  option dns-server
  option domain
  }

- With this configuration you get
  – An IPv6 address
  – DNS servers
  – Domain name


- Launch dhcpv6 client

  – dhcpv6-client run o o sudo dibbler-client run [start, stop, status, help]

- Check

  – /var/log/dibbler , also 'ifconfig eth0' to check if you have got an IPv6 address
- In /etc/resolv file you can check the DNS servers obtained
- Note that you don't get routing information, so you can't make ping6
  – The routing information is obtained by means of stateless autoconfiguration (RA)

# Privacy Extensions (1)

- RFC 3041 Privacy Extensions for Stateless Address Autoconfiguration in IPv6

- Extension of Stateless Autoconfiguration

- It generates a global address that changes over time

- It makes more difficult to identify when different addresses used in different transactions actually correspond to the same node

# Privacy: Windows (1)

- From Windows XP/2003, it is enabled by default
- **netsh interface ipv6 show privacy** to see the status
- **ipconfig, ipv6 if** or **netsh interface ipv6 show addresses** to see the autoconfigured address
- To disable it:
  - **netsh interface ipv6 set privacy state=disabled store=persistent**
  - **ipv6 [-p] gpu UseTemporaryAddresses no**
- To check the change **ipconfig, ipv6 if** or **netsh**
- **interface ipv6 show addresses**
- In XP/2003 is posible to need "disable" and "enable" the physical interface on Network Connection in order to really see the change

# Privacy: Windows (2)

- **Additional options with netsh command:**

- netsh interface ipv6 set privacy [[state=] enabled|disabled] [[maxdadattempts=] <integer>] [[maxvalidlifetime=]<integer>] [[maxpreferredlifetime=]<integer>] [[regeneratetime=]<integer>] [[maxrandomtime=]<integer>] [[randomtime=] <integer>] [[store=]active|persistent]

# Privacidad: Linux

- Disable by default
  - Enable - temporal
    - sysctl -w net.ipv6.conf.all.use_tempaddr=2
    - sysctl -w net.ipv6.conf.default.use_tempaddr=2
    - sysctl -w net.ipv6.conf.eth0.use_tempaddr=2
  - Enable - permanent
    - Editar en /etc/sysctl.conf
    - net.ipv6.conf.all.use_tempaddr = 2
    - net.ipv6.conf.default.use_tempaddr = 2
    - net.ipv6.conf.eth0.use_tempaddr = 2
    - Restart
  - Disable
    - sysctl -w net.ipv6.conf.eth0.use_tempaddr=0
    - sysctl -w net.ipv6.conf.all.use_tempaddr=0

THE IPv6 PORTAL

ConsulinTel
Consultores Integrales en Telecomunicaciones

# Privacidad: MAC OS X

- Disable by default
  - Enable  - temporary
    - sysctl net.inet6.ip6.use_tempaddr=1

  - Disable
    - sysctl net.inet6.ip6.use_tempaddr=0

# Part 3

# Transition Mechanisms Configuration

# Transition Mechanisms

- IPv6 has been designed for easing the transition and coexistence with IPv4
- Several strategies have been designed for coexisting with IPv4 hosts
  - Dual stack: Simultaneous support for both IPv4 and IPv6 stacks
  - Tunnels: IPv6 packets encapsulated in IPv4 ones
    - This is the commonest choice
  - Translation:This should be the last choice because it isn't perfect

# Dual Stack (1)



IPv6 only stack

IPv6 and IPv4
dual stack

IPv4 only stack

IPv6
Application

IPv6
Application

IPv4
Application

IPv4
Application

TCP/UDP

TCP/UDP

TCP/UDP

IPv6

IPv6

IPv4

IPv4

IPv6

IPv4

Dual Stack Mechanism
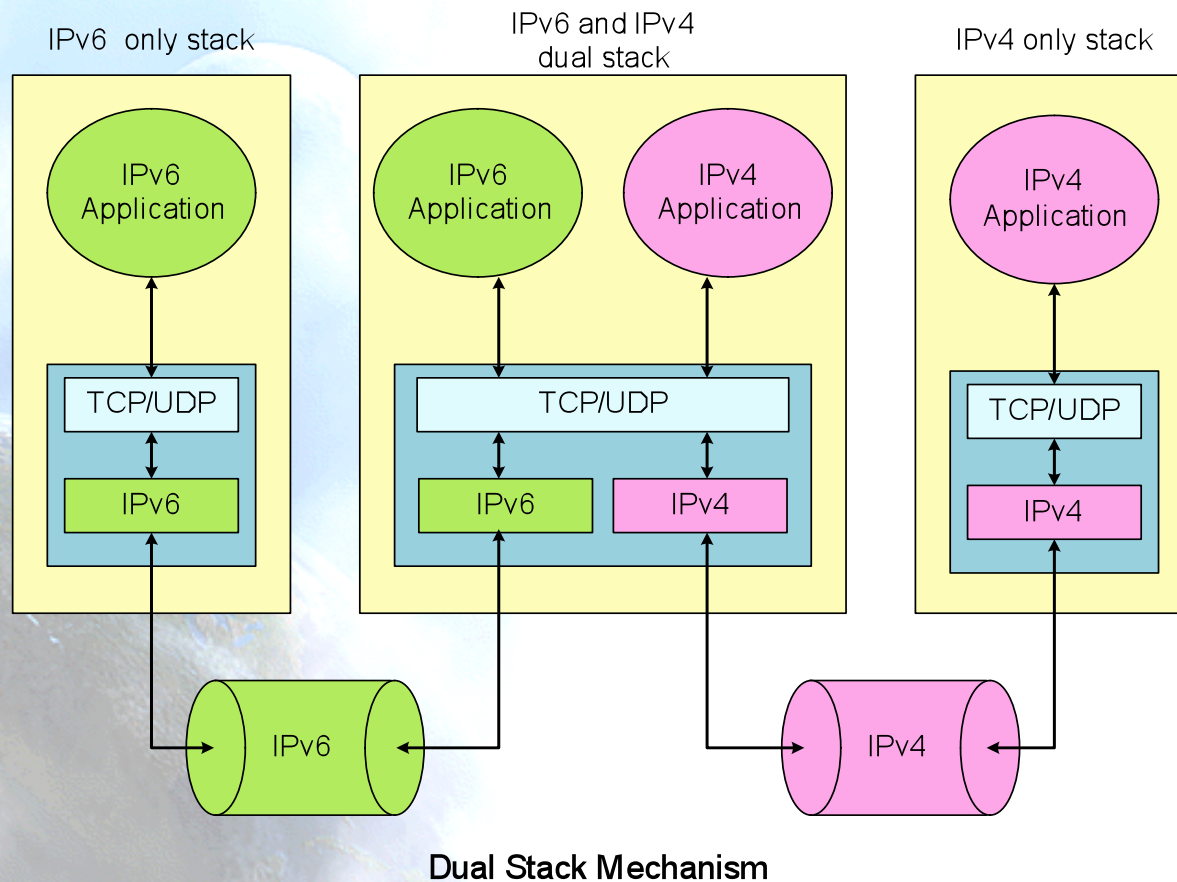
- The hosts have IPv6, IPv6 or IPv6-IPv4 stacks
- IPv6-only communications ==> IPv6 stack, assuming IPv6 network support
- IPv4-only communications ==> IPv4 stack

# Dual Stack (2)
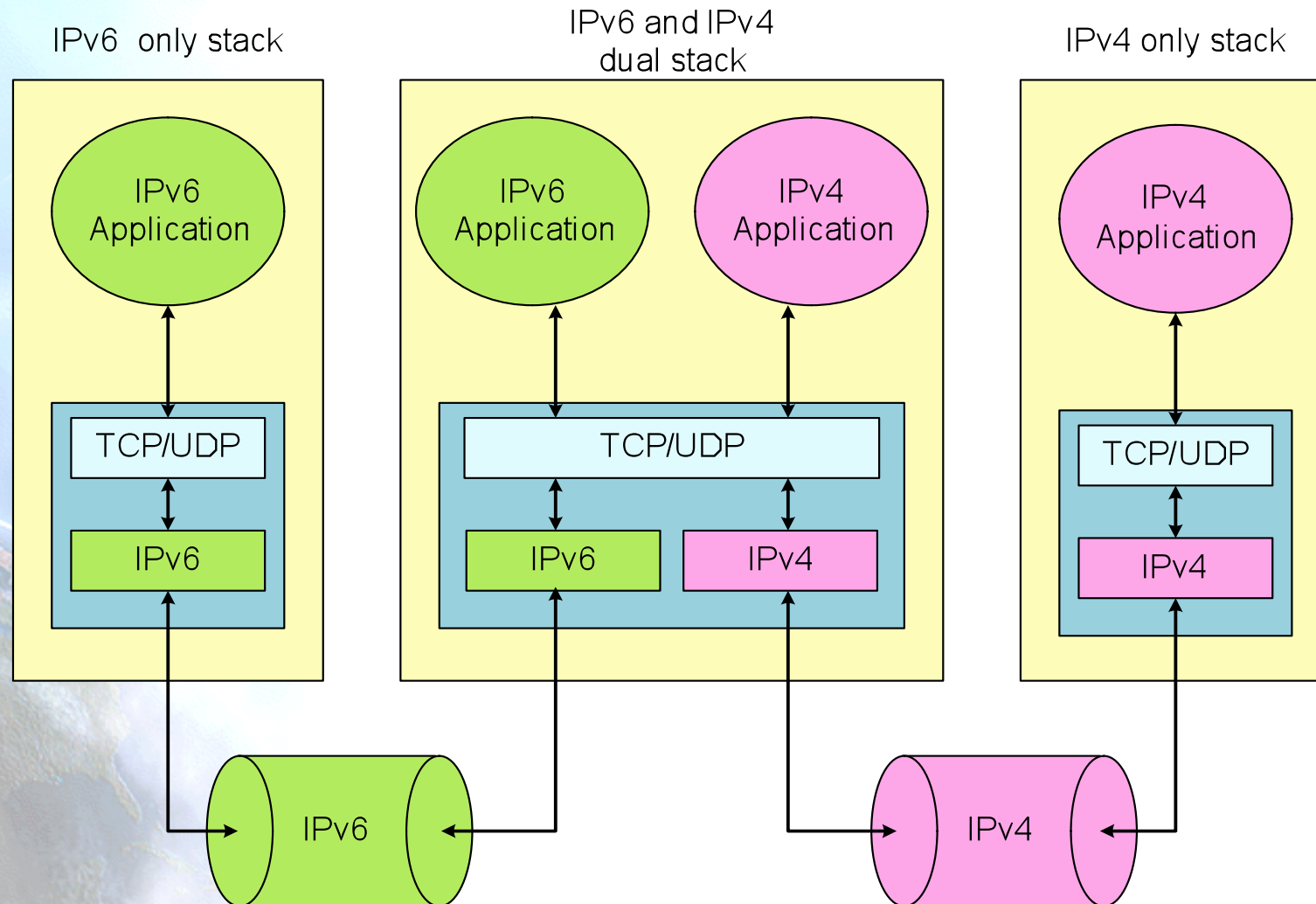


Dual Stack Mechanism

# Tunnels IPv6 in IPv4 (1)

Internet IPv4

IPv6

IPv6

Tunnel Mechanism

IPv6

IPv4 | IPv6
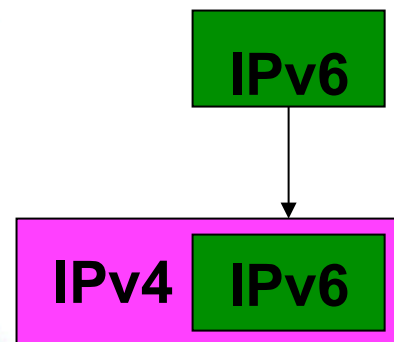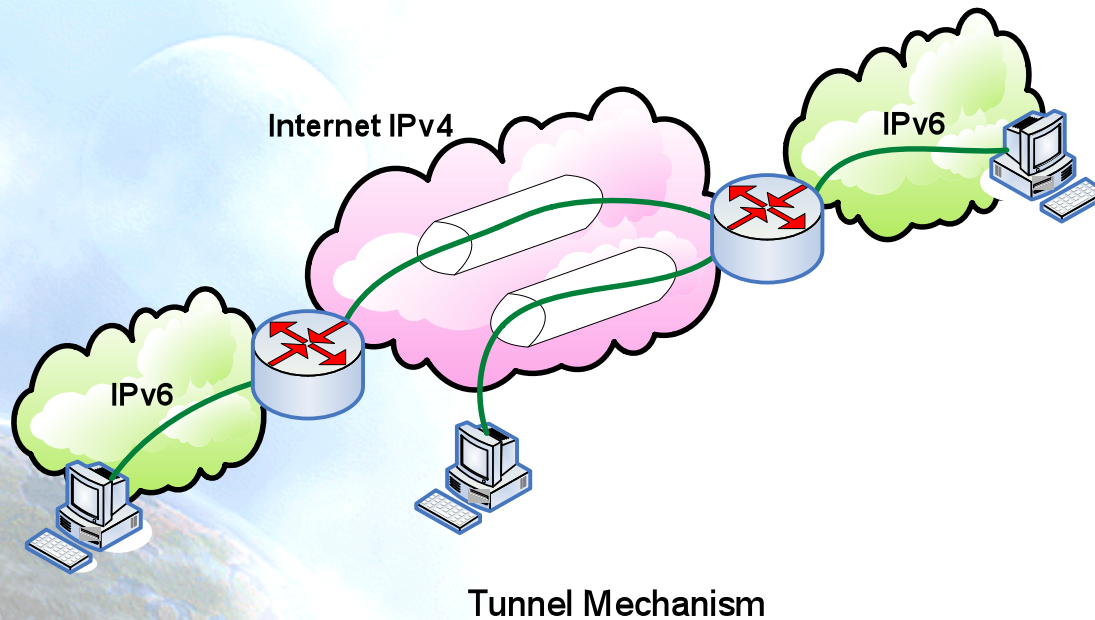
- Used to provide IPv6 connectivity in IPv4-only networks
- IPv6 packets are encapsulated into IPv4 packets
- The resulting packets flow through IPv4 networks towards the tunnel end point (TEP)

# Tunnels IPv6 in IPv4 (2)



Internet IPv4

IPv6

IPv6

Tunnel Mechanism

IPv6

IPv4  IPv6
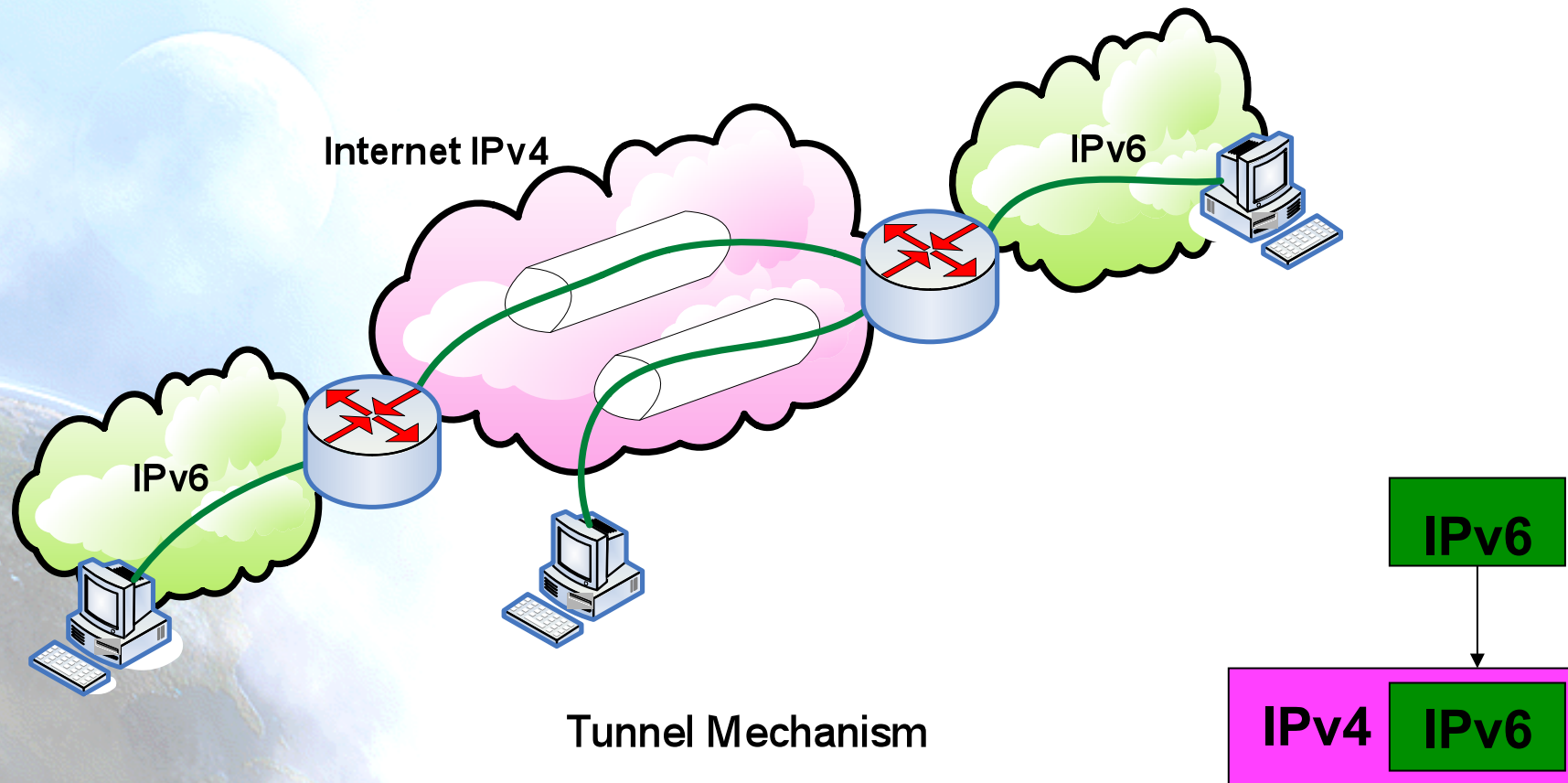
# Type of Tunnels IPv6 in IPv4 (1)

- There are different ways for encapsulating the IPv6 packets into IPv4 ones

| IPv6 |
|------|
| IPv4 |

| IPv6 |
|------|
| GRE  |
| IPv4 |

| IPv6 |
|------|
| UDP  |
| IPv4 |
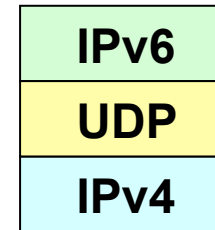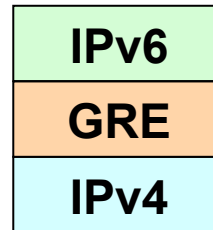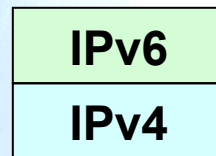
- Same for IPv4 being used in IPv6-only networks

# Types of Tunnels IPv6 in IPv4 (2)

- Some transition mechanism based on tunnels:
  - 6in4 (*) [6in4]
  - TB (*) [TB]
  - TSP [TSP]
  - 6to4 (*) [6to4]
  - TEREDO (*) [TEREDO], [TEREDOC]
  - Automatic tunnels[TunAut]
  - ISATAP [ISATAP]
  - 6over4 [6over4]
  - AYIYA [AYIYA]
  - Silkroad [SILKROAD]
  - DSTM [DSTM]

- (*) Commonest mechanisms and explained in depth in the following slides

# 6in4 Tunnels

- It encapsulates directly the IPv6 packet into the IPv4 packet
- It is usually used between:
  - end host ==> router
  - router ==> router

- However, it is also possible for
  - end host ==> end host
- From the point of view of IPv6 the tunnel is considered as a point-to-point link
  - Only an IPv6 network-hop although several IPv4-hops exist in the path
- The IPv6 addresses of both tunnel-ends belong to the same prefix
- All the IPv6 connections of the end-host flow always through the router located at the tunnel-end-point
- The 6in4 tunnels can be built from end-hosts located behind a NAT box
  - It is essential that the NAT implementation supports "proto-41 forwarding" [PROTO41] to let the IPv6-encasulated packets traverse the NAT box

Internet IPv6

Commun path for all IPv6 communications

2001:db8:40:2a0a::81/126

Internet IPv4

2001:db8:40:2a0a::82/126
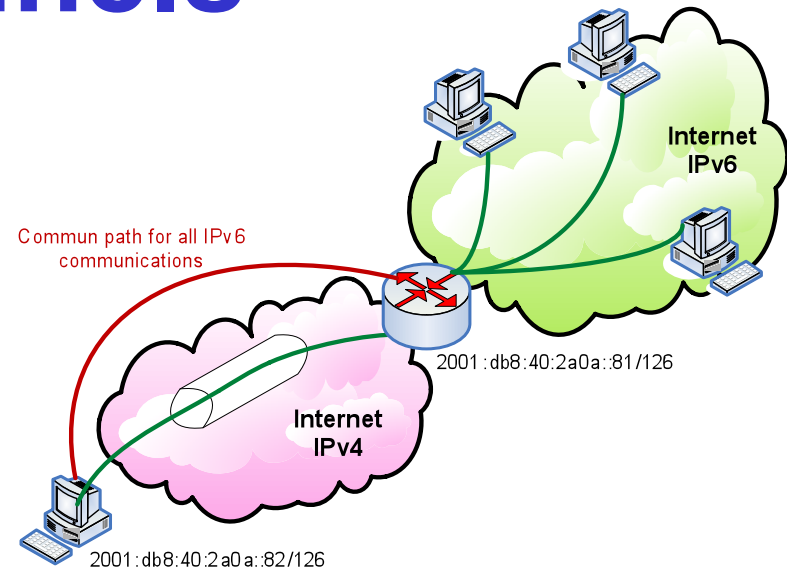
# Tunnel Broker (1)

- RFC3053 IPv6 Tunnel Broker
- The 6in4 tunnels require the manual configuration of the devices involved in the tunnel creation
- To easy the address assignment and the IPv6 tunnel creation, the Tunnel Broker (TB) concept has been developed
  - It is a intermediate host which the end user is connected, usually by using a web browser
- The user asks to the TB the creation of an IPv6 tunnel. The TB assigns to the user an IPv6 address and gives to the user instructions for building the tunnel in the user's side
- The TB also configures the router, which is the TEP for the end user
- In http://www.ipv6tf.org/using/connectivity/test.php exists a list of available TBs
- TSP [TSP] is a special case of TB because it is based on an application installed in the user's host which contacts to the TSP server to built the IPv6 tunnel. However, the concept is similar to the one previously enounced

# Tunnel Broker (2)



Web Interface

Internet IPv6

Tunel 6in4

Internet IPv4

Cliente

Router – Tunnel Server
2001:db8:40:2a0a::81/126

2001:db8:40:2a0a::82/126
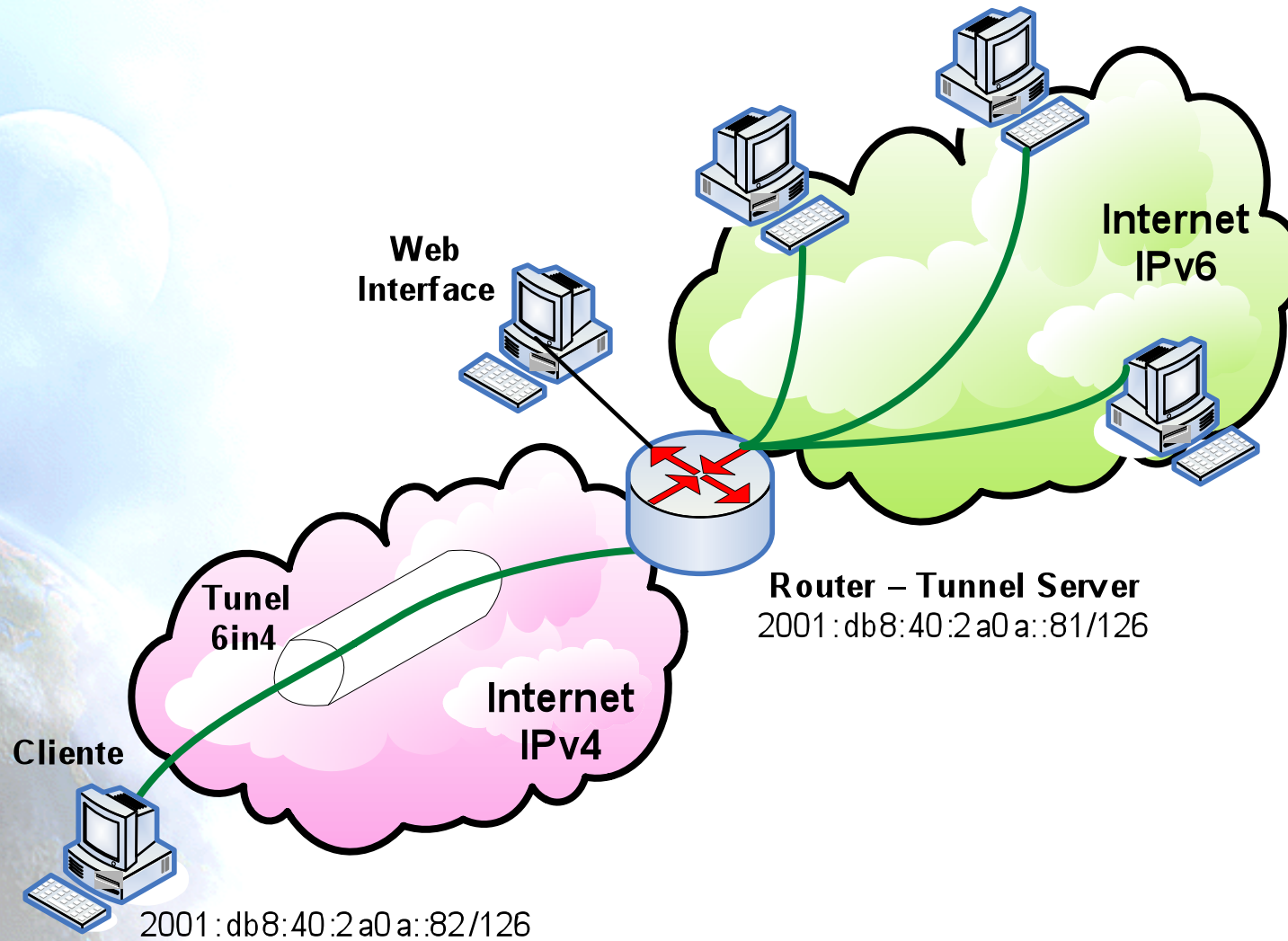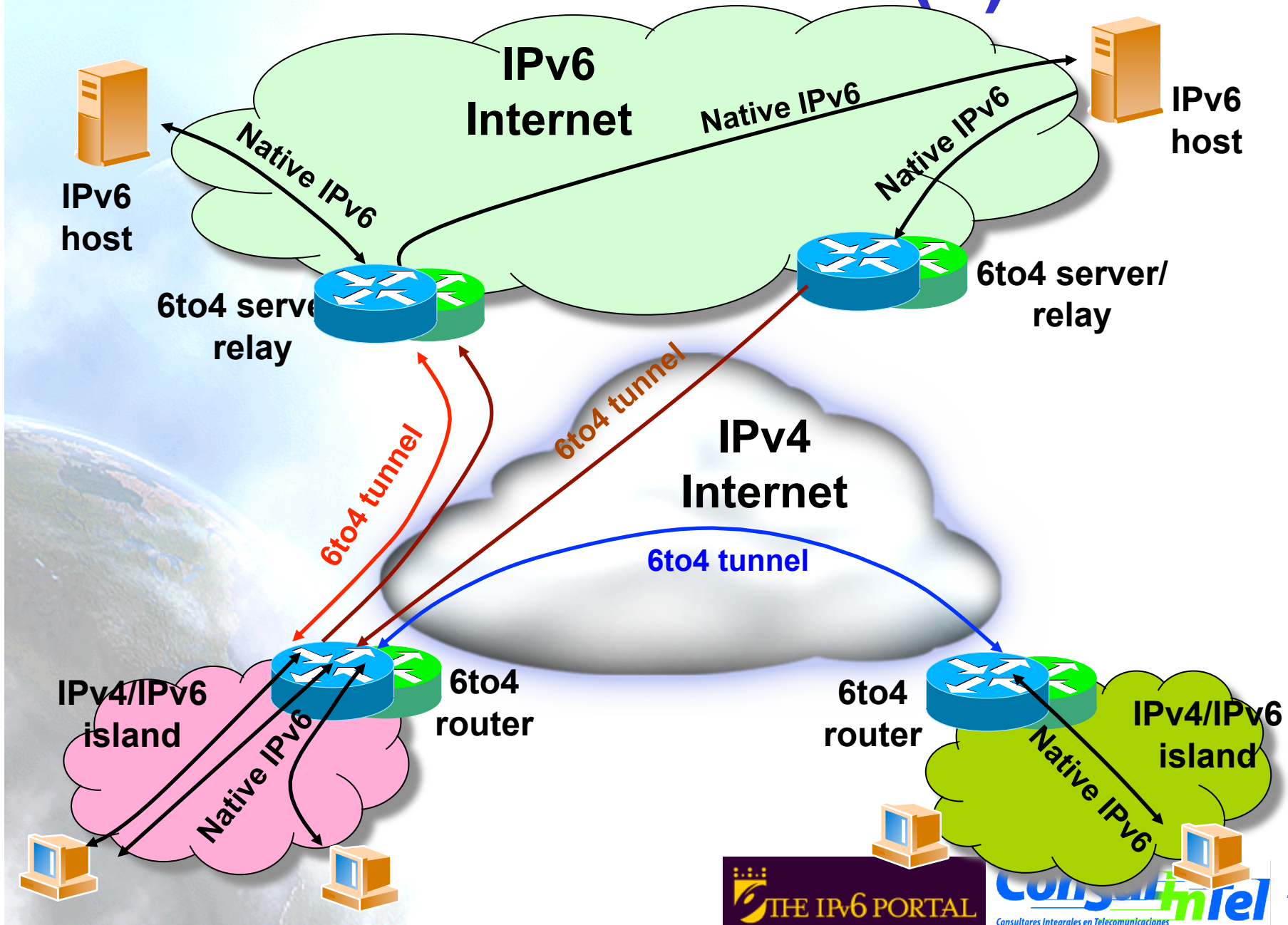
# 6to4 Tunnels (1)

# 6to4 Tunnels (2)

- IPv6 packets are encapsulated into IPv4 ones, in a similar way than the 6in4 tunnels

- Differences:
  - The user's IPv6 address does not depend on the router used to get IPv6 connected but on the public IPv4 used by the user
    - Prefix 2002::/16
  - All the user's outgoing IPv6 packets are always sent to the same "6to4 relay". However the user's incoming IPv6 packets could come from different "6to4 relays"

- IPv4 anycast prefix:
  - 192.88.99.1

# TEREDO (1)
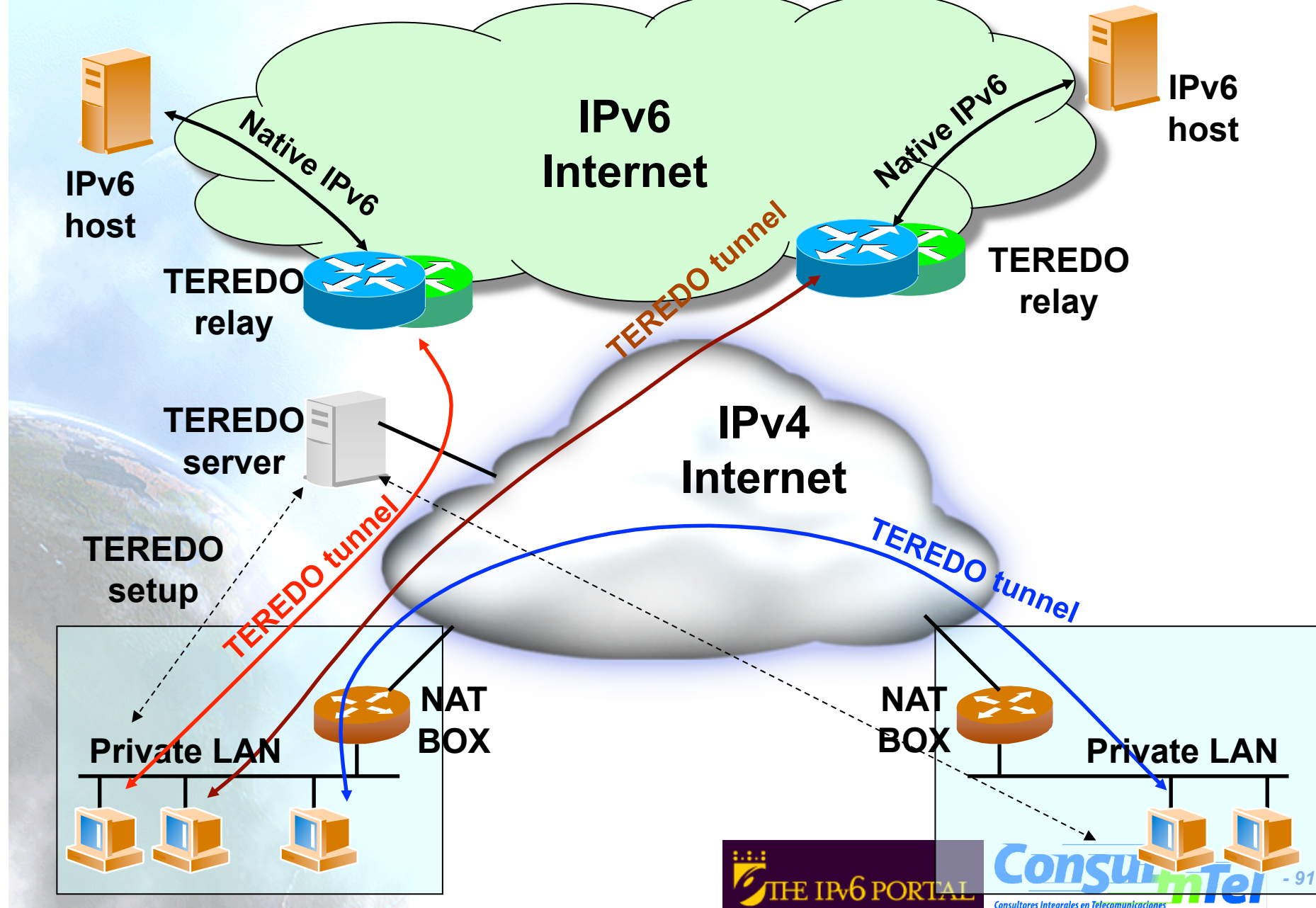
# TEREDO (2)

- TEREDO [TEREDO] [TEREDOC] is thought for providing IPv6 to hosts that are located behind a NAT box that is not "proto-41 forwarding"
  - It encapsulates the IPv6 packets into UDP/IPv4 packets
- It only works in the following NAT types
  - Full Cone
  - Restricted Cone
  - Symmetric (supported from Windows Vista)
- TEREDO uses different agents to work:
  - TEREDO Server
  - TEREDO Relay
  - TEREDO Client
- The user configures in its host a TEREDO Server which provides an IPv6 address from the 2001:0000::/32 prefix and such an address is based on the user's public IPv4 address and used UDP port
  - If the TEREDO Server is also a TEREDO Relay, the user has also IPv6 connectivity with any IPv6 hosts
  - Otherwise, the user only has IPv6 connectivity with other TEREDO users
- Microsoft currently provides public TEREDO Servers for free, but not TEREDO Relays

# Softwires: RFC4925

- Protocol being discussed within IETF's Softwire WG. Characteristics:
  - "Universal" transition mechanism based on tunnels
    - IPv6-in-IPv4, IPv6-in-IPv6, IPv4-in-IPv6, IPv4-in-IPv4
    - NAT traversal on access networks
    - Provides IPv6 prefix delegation (/48, /64, etc.)
    - User authentication for tunnel creation using AAA infrastructure
    - Possibility of secure tunnels
    - Low overhead of IPv6 packets over the tunnels
    - Supports portable devices with scarce hardware resources
  - Will enable provision of IPv6 connectivity to devices like ADSL routers, mobile phones, PDAs, etc. when no native IPv6 connectivity exists
  - Could provide IPv4 connectivity to devices with IPv6 only connectivity
- Softwires is not a new protocol but the definition of how to use existing protocols in order to provide IPv6 connectivity on IPv4 only networks and vice versa
- It is based on L2TPv2 (RFC2661) and L2TPv3 (RFC3991)

THE IPv6 PORTAL

Consulintel
Consultores Integrales en Telecomunicaciones

# Softwires Encapsulating based on L2TPv2

- Described on RFC5571
- There are two entities:
  - Softwires Initiator (SI): agent who solicits the tunnel
  - Softwires Concentrator (SC): agent who creates the tunnel (tunnel end-point)
- PPP is used to transport IPvx (x=4 or 6) in IPvx (x=4 or 6) packets
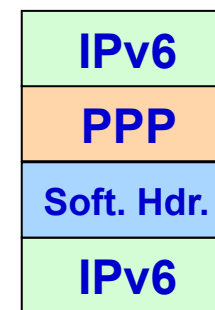  - Optionally PPP packets can be encapsulated on UDP for NAT traversal
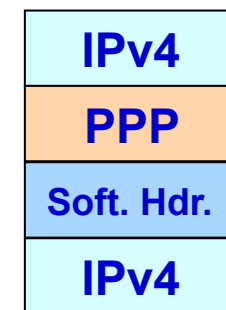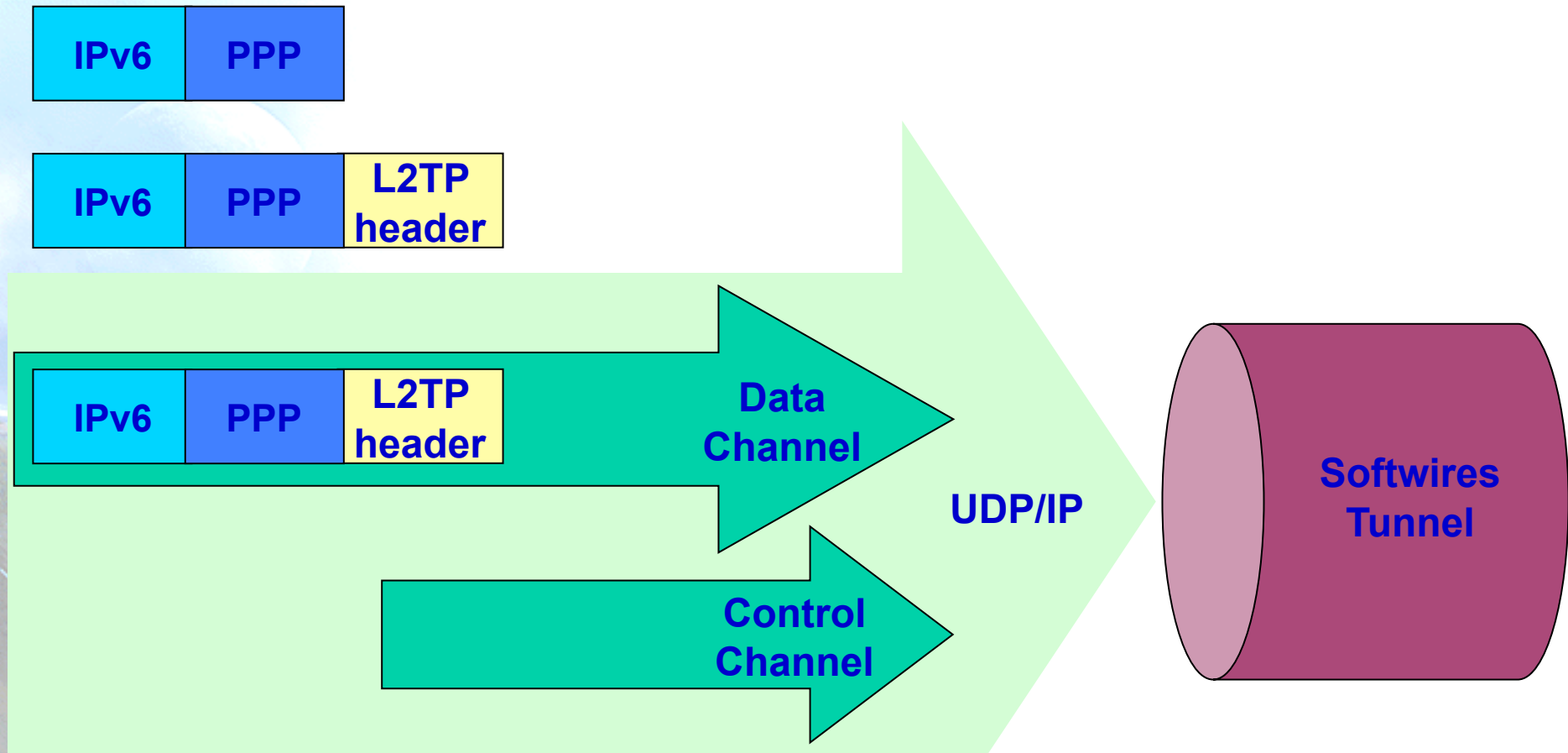
**IPv6-in-IPv4 Tunnel**

| IPv6 |
|------|
| PPP |
| Soft. Hdr. |
| IPv4 |

| IPv6 |
|------|
| PPP |
| Soft. Hdr. |
| UDP* |
| IPv4 |
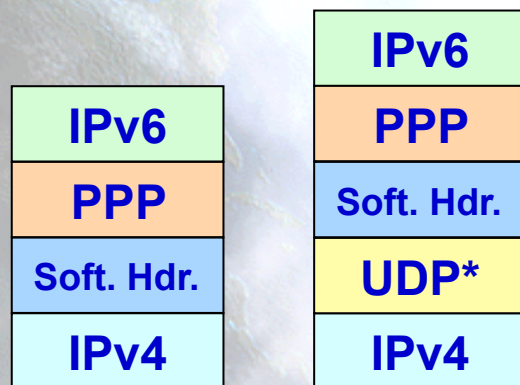
**IPv4-in-IPv6 Tunnel**

| IPv4 |
|------|
| PPP |
| Soft. Hdr. |
| IPv6 |

**IPv6-in-IPv6 Tunnel**

| IPv6 |
|------|
| PPP |
| Soft. Hdr. |
| IPv6 |
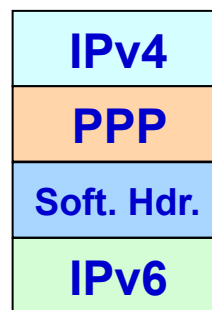
**IPv4-in-IPv4 Tunnel**

| IPv4 |
|------|
| PPP |
| Soft. Hdr. |
| IPv4 |

**\* Optional**

# Softwires Based on L2TPv2

| IPv6 | PPP |
|------|-----|

| IPv6 | PPP | L2TP header |
|------|-----|-------------|

| IPv6 | PPP | L2TP header | **Data Channel** |
|------|-----|-------------|------------------|

**Control Channel**

**UDP/IP**

**Softwires Tunnel**

- There are a Control and a Data Plane
- PPP is used as an encapsulating protocol

# Example of Use

- An expected use of Softwires is for providing IPv6 connectivity to domestic users through an IPv6-only access network
  - The SC is on ISP's network (DSLAM, Aggregation Router, or other device)
  - The SI is on user's network (the CPE or other device)
  - The SC provides IPv6 connectivity to the SI and the SI act as IPv6 router for user networks
  - Prefix delegation (DHCP-PD) is used between the SC and the SI to provide an IPv6 prefix (typically a /48)
- Other uses are possible:
  - VPNs over IPv4 or IPv6
  - IPv4 connectivity over an IPv6-only access network
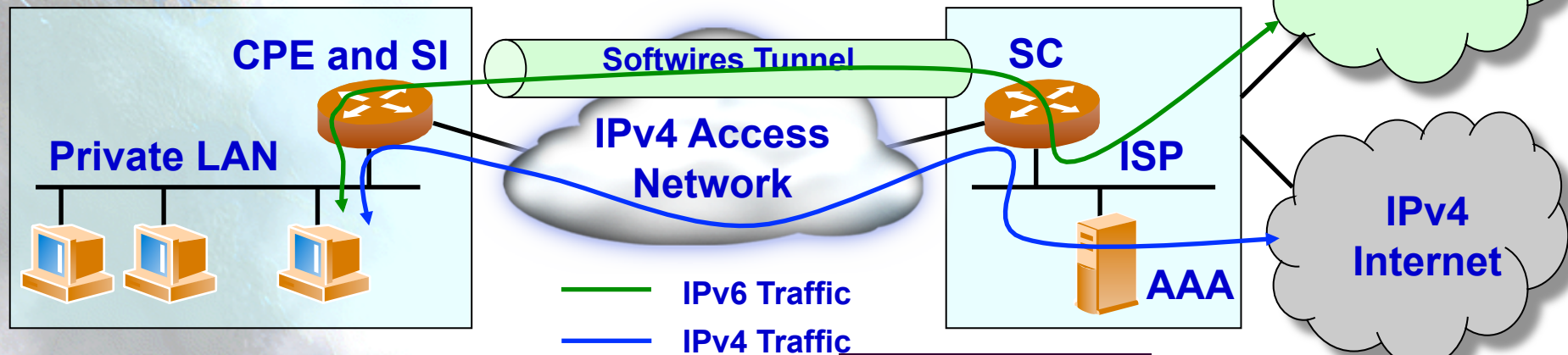
# Softwires Encapsulating Based on L2TPv3

- Same philosophy as with L2TPv2 but with L2TPv3 particularities:
  - Transport over IP/UDP of other layer two protocols different than PPP: HDLC, FR, ATM, Ethernet or MPLS
  - Enhanced header format for better performance in the SC (speeds equal to T1/E1, T3/E3, OC48)
  - Minimum overhead on encapsulated packets (only 4 to 12 extra bytes)
  - Adds EAP as authentication mechanism to CHAP and PAP used in L2TPv2

**IPv6-in-IPv4 Tunnel**

| IPv6 |
| --- |
| Layer 2 |
| Soft. Hdr. |
| IPv4 |

| IPv6 |
| --- |
| Layer 2 |
| Soft. Hdr |
| UDP* |
| IPv4 |

- **HDLC**
- **PPP**
- **FR**
- **ATM**
- **Ethernet**
- **MPLS**

**\* Optional**

THE IPv6 PORTAL

Consulintel
Consultores Integrales en Telecomunicaciones

# Translation IPv4/IPv6

IPv6

Internet
IPv4

NAT-PT

**Translation Mechanism**
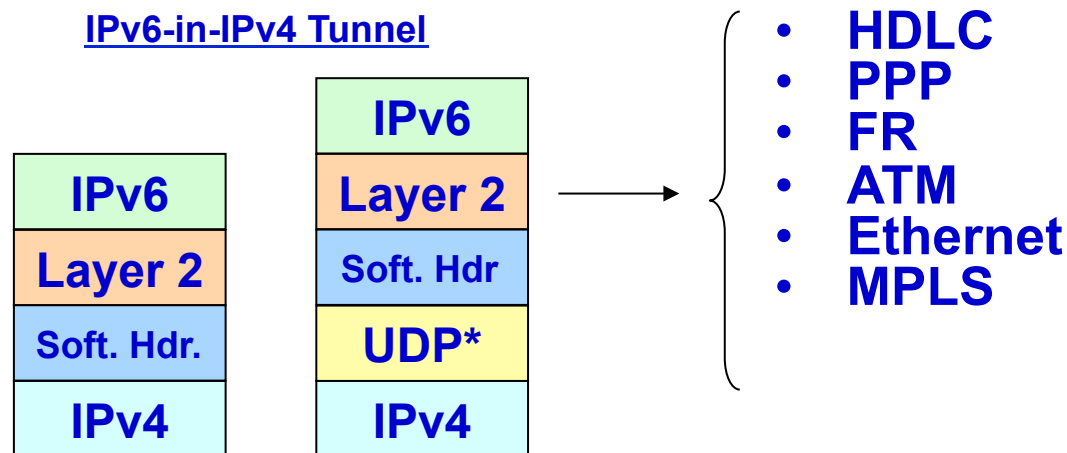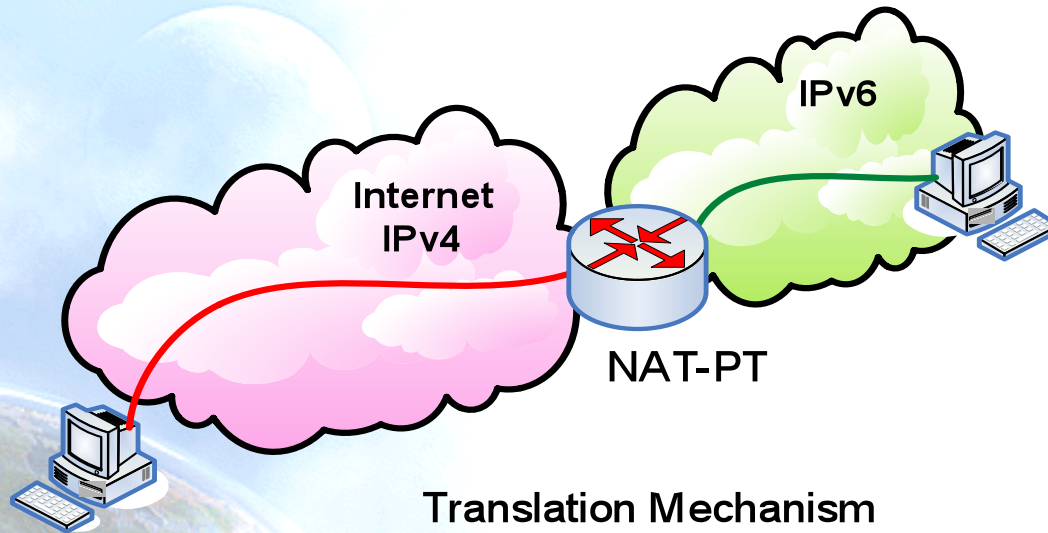
- There are several solutions, but all of them try to translate IPv4 packets into IPv6 and vice-versa
  - [SIT], [BIS], [TRT], [SOCKSv64]
- The commonest is NAT-PT [NATPT], [NATPTIMPL]
  - An intermediate node (router) modifies the IPv4 headers to convert them into IPv6 headers
  - The treatment of the packets is complex
- It is the worst solution because the translation is not perfect and it requires ALGs support, in the same way that IPv4-NATs
  - DNS, FTP, VoIP, etc.

# Configuration of Transition Mechanisms: Exercises

- E1: Setup a 6in4 tunnel between two alumni's hosts
- E2: Delete the 6in4 tunnel
- E3: Get IPv6 connectivity by means of a 6in4 tunnel by using a TB
    - See the path to different IPv6 web sites
    - See the path to the provided IPv6 address from a looking glass
- E4: Get IPv6 connectivity by means of a 6to4 tunnel
    - See the path to different IPv6 web sites
    - See the path to the provided IPv6 address from a looking glass
- E5: Setup a 6to4 relay (Windows)
- E6: Setup a TEREDO Client (Windows XP/2003)
- E7: Setup a TEREDO Client (Linux)
- E8: Setup a TEREDO Server (Linux)
- E9: Usage of IPv4/IPv6 proxies
    - 46Bouncer
    - Windows XP/2003

# E1: 6in4 Tunnel Setup (1)

1. Exercise to be made with partners (*)
   - Alumni A ==> ADD_IPv4_A
   - Alumni B ==> ADD_IPv4_B
2. Alumni A sets up the tunnel in his side by using the following data:
   - Local IPv6 address ==> ADD_IPv4_A
   - Remote IPv4 address ==> ADD_IPv4_B
   - IPv6 address ==> 2001:db8:20:30::12/126
   - IPv6 gateway address ==> 2001:db8:20:30::11/126
3. Alumni B sets up the tunnel in his side by using the following data:
   - Local IPv4 address ==> ADD_IPv4_B
   - Remote IPv4 address ==> ADD_IPv4_A
   - IPv6 address ==> 2001:db8:20:30::11/126
   - IPv6 gateway address ==> 2001:db8:20:30::12/126
4. Check IPv6 connectivity between both alumni
   - Alumni A ==> ping6 IPv6_Address_Alumna_B
   - Alumni B ==> ping6 IPv6_Address_Alumna_A
5. Enable forwarding
   - Alumni A ==> enable forwarding in both tunnel and LAN interfaces
   - Alumni B ==> enable forwarding in both tunnel and LAN interfaces

- (*) This exercise does not provide global IPv6 connectivity, just IPv6 connectivity between alumni A and alumni B

# E1: 6in4 Tunnel Setup (2)

- Scripts for setting up 6in4 tunnels
  - Windows XP/2003 (from the command line window)
    - netsh interface ipv6 add v6v4tunnel "Tunnel01" Address_IPv4_local Address_IPv4_remote
    - netsh interface ipv6 add address "Tunnel01" Address_IPv6
    - netsh interface ipv6 add route ::/0 "Tunnel01" Address_gateway_IPv6 publish=yes
    - netsh interface ipv6 set interface "Tunnel01" forwarding=enable
    - netsh interface ipv6 set interface "LAN" forwarding=enable
  - Linux/UNIX (from the shell)
    - modprobe ipv6
    - ip tunnel add Tunnel01 mode sit remote Address_IPv4_remote local Address_IPv4_local ttl 255
    - ip link set Tunnel01 up
    - ip addr add Address_IPv6/126 dev Tunnel01
    - ip route add 2000::/3 dev Tunnel01
  - FreeBSD
    - gifconfig gif0 Address_IPv4_local Address_IPv4_remote
    - ifconfig gif0 inet6 Address_IPv6 Address_gateway_IPv6 prefixlen 128
    - route -n add -inet6 default Address_gateway_IPv6

THE IPv6 PORTAL

Consul**InTel**
Consultores Integrales en Telecomunicaciones

# E1: 6in4 Tunnel Setup (3)

- Scripts for setting up 6in4 tunnels
  - FreeBSD >= 4.4
    - ifconfig gif0 create
    - ifconfig gif0 tunnel Address_IPv4_local Address_IPv4_remote
    - ifconfig gif0 inet6 Address_IPv6 Address_gateway_IPv6 prefixlen 128
    - route add -inet6 default Address_gateway_IPv6
  - NetBSD
    - ifconfig gif0 Address_IPv4_local Address_IPv4_remote
    - ifconfig gif0 inet6 Address_IPv6 Address_gateway_IPv6 prefixlen 128
    - route -n add -inet6 default Address_gateway_IPv6
  - OpenBSD
    - ifconfig gif0 giftunnel Address_IPv4_local Address_IPv4_remote
    - ifconfig gif0 inet6 Address_IPv6 Address_gateway_IPv6 prefixlen 128
    - route -n add -inet6 default Address_gateway_IPv6

# E2: Deleting 6in4 tunnels (1)

- Exercise to be done by each alumni (individually)
- The alumni deletes the tunnel configured previously according to the configuration script of its Operating System
- The alumni has to check that the tunnel has been deleted by using:
  - ipconfig on Windows XP/2003
  - ifconfig on Unix/Linux/*BSD

# E2: Deleting 6in4 Tunnels (2)

- Scripts for deleting 6in4 tunnels
  - Windows XP/2003 (from the command line window)
    - netsh interface ipv6 del route ::/0 "Tunnel01" Address_gateway_IPv6
    - netsh interface ipv6 del address "Tunnel01" Address_IPv6
    - netsh interface ipv6 del int "Tunnel01"
  - Linux/UNIX (from the shell)
    - ip route del 2000::/3 dev Tunnel01
    - ip addr del Address_IPv6/126 dev Tunnel01
    - ip link set Tunnel01 down
    - ip tunnel del Tunnel01 mode sit remote Address_IPv4_remote local Address_IPv4_local ttl 255
  - FreeBSD
    - route delete -inet6 default
    - ifconfig gif0 inet6 delete Address_IPv6
    - ifconfig gif0 down

# E2: Deleting 6in4 Tunnels (3)

- Scripts for deleting 6in4 tunnels
  - FreeBSD >= 4.4
    - route delete -inet6 default Address_gateway_IPv6
    - ifconfig gif0 inet6 Address_IPv6 prefixlen 128 delete
    - ifconfig gif0 delete
  - NetBSD
    - route delete -inet6 default
    - ifconfig gif0 inet6 delete Address_IPv6
    - ifconfig gif0 down
  - OpenBSD
    - ifconfig gif0 inet6 delete Address_IPv6
    - ifconfig gif0 deletetunnel
    - ifconfig gif0 down
    - route delete -inet6 default

# E3: IPv6 Connectivity via a TB

1. Choose a TB from http://www.ipv6tf.org/using/connectivity/test.php
2. Follow the steps provided by the TB
3. Check that the IPv6 connectivity is available
   - ping6, traceroute6 (ping & tracert on windows)
     - www.kame.net, www.6power.org, www.ipv6.org
   - Browsing to the same web sites
4. Check the path to the assigned IPv6 address from an external looking glass
   - http://www.ipv6tf.org/using/connectivity/looking_glass.php
   - http://www.ipv6.udg.mx/lg.php
   - http://www.v6.dren.net/lg/

# E4: IPv6 Connectivity with 6to4 (1)

1. Choose a 6to4 relay from http://www.ipv6tf.org/ using/connectivity/6to4.php
2. Follow the configuration script according to the proper Operating System
3. Check that the IPv6 connectivity is available
   - ping6, traceroute6 (ping & tracert en windows)
     - www.kame.net, www.6power.org, www.ipv6.org
   - Browsing to the same web sites
4. Check the path to the assigned IPv6 address from an external looking glass
   - http://www.ipv6tf.org/using/connectivity/ looking_glass.php
   - http://www.ipv6.udg.mx/lg.php
   - http://www.v6.dren.net/lg/

# E4: IPv6 Connectivity with 6to4 (2)

- Scripts for deleting the 6to4 tunnels
  - Windows XP/2003 (from the command line window)
    - netsh int ipv6 6to4 set relay *<Address_6TO4_RELAY>* enabled 1440
  - Linux/UNIX (from the shell)
    - ip tunnel add tun6to4 mode sit ttl 80 remote any local *<Address_public_IPv4_local>*
    - ip link set dev tun6to4 up
    - ip -6 addr add **2002:XXYY:ZZUU::1/16** dev tun6to4
    - ip -6 route add 2000::/3 via ::192.88.99.1 dev tun6to4 metric 1

    - **Note that XXYY:ZZUU** is the hexadecimal notation for Address_public_IPv4_local (the public IPv4 address) according to the following:

    - Address_public_IPv4_local = 60.172.21.22 ->   60 -> 3C
    -                                                172 -> AC
    -                                                 21 -> 15
    -                                                222 -> DE

    - 60.172.21.22 -> XXYY:ZZUU = 3CAC:15DE

# E4: IPv6 Connectivity with 6to4 (3)

- Scripts for deleting 6to4 tunnels
  - \*BSD
    - Be sure that there is at least one stf(4) interface configured in the kernel
      - In http://www.netbsd.org/Documentation/kernel/ information about that can be found

    - ifconfig stf0 inet6 **2002:XXYY:ZZUU::**1 prefixlen 16 alias
    - route add -inet6 default 2002:c058:6301::1

    - **Note that XXYY:ZZUU** is the hexadecimal notation for Address_public_IPv4_local (the public IPv4 address) according to the following:

    - Address_public_IPv4_local = 60.172.21.22 ->   60 -> 3C
    - 172 -> AC
    - 21 -> 15
    - 222 -> DE

    - 60.172.21.22 -> XXYY:ZZUU = 3CAC:15DE

# E5: 6to4 Relay (1) Windows 2003

- The 6to4 Relay configuration is done using **netsh interface ipv6 set**
  - netsh interface ipv6 set interface interface="Local area connection" forwarding=enabled
  - netsh interface ipv6 set state state=enabled undoonstop=disabled
  - netsh interface ipv6 set relay name=192.88.99.1 state=enabled interval=1440
    - Instead the anycast address 192.88.99.1, is possible to use any of http://www.ipv6tf.org/using/connectivity/6to4.php
  - netsh interface ipv6 set routing routing=enabled sitelocals=enabled

- Every 6to4 packet received by the "Local area connection" interface will be forwarded to the proper IPv6 destination
- In order to check the 6to4 relay configuration
  - A 6to4 tunnel can be configured in other host (following the instructions of previous slides) and the 6to4 server in such a new host will be the 6to4 relay just configured
  - Doing ping and traceroute to check IPv6 connectivity

THE IPv6 PORTAL

ConsulinTel
Consultores Integrales en Telecomunicaciones

# E5: 6to4 Relay (2) Windows Vista and 7

- The 6to4 Relay configuration is done using **netsh interface ipv6 6to4 set**
  - netsh interface ipv6 6to4 set interface " Local area connection " routing=enabled
  - netsh interface ipv6 6to4 set state state=enabled undoonstop=disabled
  - netsh interface ipv6 6to4 set relay name=192.88.99.1 state=enabled interval=1440
    - Instead the anycast address 192.88.99.1, is possible to use any of http://www.ipv6tf.org/using/connectivity/6to4.php
  - netsh interface ipv6 6to4 set routing routing=enabled sitelocals=enabled

- Every 6to4 packet received by the "Local area connection" interface will be forwarded to the proper IPv6 destination
- In order to check the 6to4 relay configuration
  - A 6to4 tunnel can be configured in other host (following the instructions of previous slides) and the 6to4 server in such a new host will be the 6to4 relay just configured
  - Doing ping and traceroute to check IPv6 connectivity

# E6: TEREDO Client Windows

- Windows XP/2003 have an implementation of TEREDO Client
- From a DOS window type the following:
  - set teredo client teredo.ipv6.microsoft.com. 60 34567
  - a public TEREDO Server by Microsoft is used
    - teredo.ipv6.microsoft.com
- There exist other experimental TEREDO Server/Relays (without guaranteed service)
  - teredo.remlab.net (France)
  - teredo.autotrans.consulintel.com (Spain)
  - teredo.ipv6.microsoft.com (USA, Redmond) (default for Windows XP/2003/2008/Vista/7)
  - teredo.ngix.ne.kr (South Korea)
  - debian-miredo.progsoc.org (Australia)
- Check the provided IPv6 address
  - ipconfig
- Check the data of the TEREDO interface
  - netsh int ipv6 show teredo
  - netsh int ipv6 show int teredo
- Global IPv6 connectivity is not provided because Microsoft does not provide any TEREDO Relay
- IPv6 connectivity with other TEREDO clients is available
  - Check by pinging to the IPv6 address of other alumni's TEREDO Client

# E7: Cliente TEREDO Linux

- There are TEREDO implementations for other OSs such as
  - Linux: http://www.remlab.net/miredo/
  - FreeBSD: http://www-rp.lip6.fr/teredo/
- For Linux, get the TEREDO Client (Miredo)
  - Linux
    - http://www.remlab.net/miredo/
  - Ubuntu
    - sudo apt-get install miredo
- Check
  - Ifconfig
  teredo    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet6 addr: fe80::ffff:ffff:ffff/64 Scope:Link
          inet6 addr: 2001:0:53aa:64c:855:2fda:a826:a6d2/32 Scope:Global
          UP POINTOPOINT RUNNING NOARP  MTU:1280  Metric:1
          RX packets:443 errors:0 dropped:0 overruns:0 frame:0
          TX packets:487 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:528574 (528.5 KB)  TX bytes:40575 (40.5 KB)
- Which  is my TEREDO Server/Relay? #cat /etc/miredo/miredo.conf

# E8: TEREDO Server Linux

- Get the TEREDO Server (Miredo)
  - Linux
    - http://www.remlab.net/miredo/
  - Ubuntu
    - sudo apt-get install miredo-server
- Edit the file /etc/default/miredo-server
  - Descomentar START_MIREDO_SERVER=true
- Create/edit the file /etc/miredo/miredo-server.conf
  - ServerBindAddress 192.0.2.221
  - Prefix 2001:0::
- Start the server
  - sudo miredo-server run
- Check by connecting a TEREDO client with this server
  - In the file of the client /etc/miredo/miredo.conf you shuld edit
    - ServerAddress     [ipv4 address of the TEREDO server]

# E9: Use of IPv4/IPv6 Proxies (1)

- IPv4/IPv6 proxy is not the same that a transition mechanism based on translation (NAT-PT)
- Proxy is an intermediate host working on the application level
  - It receives TCP connections over a protocol (IPv4 or IPv6) and it extracts all the data from the application level
  - Then it establishes TCP connection (IPv6 or IPv4) with the destination host and it put in the new connection the application data extracted in the previous step
- It allows connections between
  - Client IPv4 ==> Proxy IPv4/IPv6 ==> Server IPv6
  - Client IPv6 ==> Proxy IPv6/IPv4 ==> Server IPv4
- There are two well-known proxies
  - 46Bouncer (Windows y Linux)
  - Windows XP/2003

# E9: Use of IPv4/IPv6 Proxies (2)

- Implement a IPv4/IPv6 Proxy on Windows XP/2003
  - Forward the TCP/ IPv4 8220 port to the TCP/IPv6 80 port of www.kame.net (2001:200:0:8002:203:47ff:fea5:3085)
  - netsh int port set v4tov6 Port_v4_TCP_local Address_IPv6_remote Port_v6_TCP_remote Address_IPv4_local
  - netsh int port set v4tov6 8220 2001:200:0:8002:203:47ff:fea5:3085 80 Address_IPv4_local
  - Check with http://address_IPv4_local

- Implement a IPv6/IPv4 Proxy on Windows XP/2003
  - Forward the TCP/IPv6 8330 port to the TCP/IPv4 80 port of www.kame.net (203.178.141.194)
  - netsh int port set v6tov4 8330 203.178.141.194 80 Address_IPv6_local

# Part 4

# Examples of Applications

# IPv6 Applications (1)

- Client-Server model implies that it is possible to have Client/Server applications working:
  - IPv4 Only
  - IPv6 Only
  - IPv4 + IPv6
- Thus provides a set of combinations that is needed to consider jointly with the availability or unavailability of IPv4/IPv6 connectivity

# IPv6 Applications (2)

- **DNS lookups** are used to make or differentiate an available service through IPv4 and/or IPv6

- If a clients wants to connect to service.example.com, when resolving the domain name he/she can get an IPv4, IPv6 or both addresses

- In the case of getting both (v4 and v6) it is up to the client which protocol (v4/v6) to choose. The common practice is to choose v6 as the first option by default

# IPv6 Applications (3)
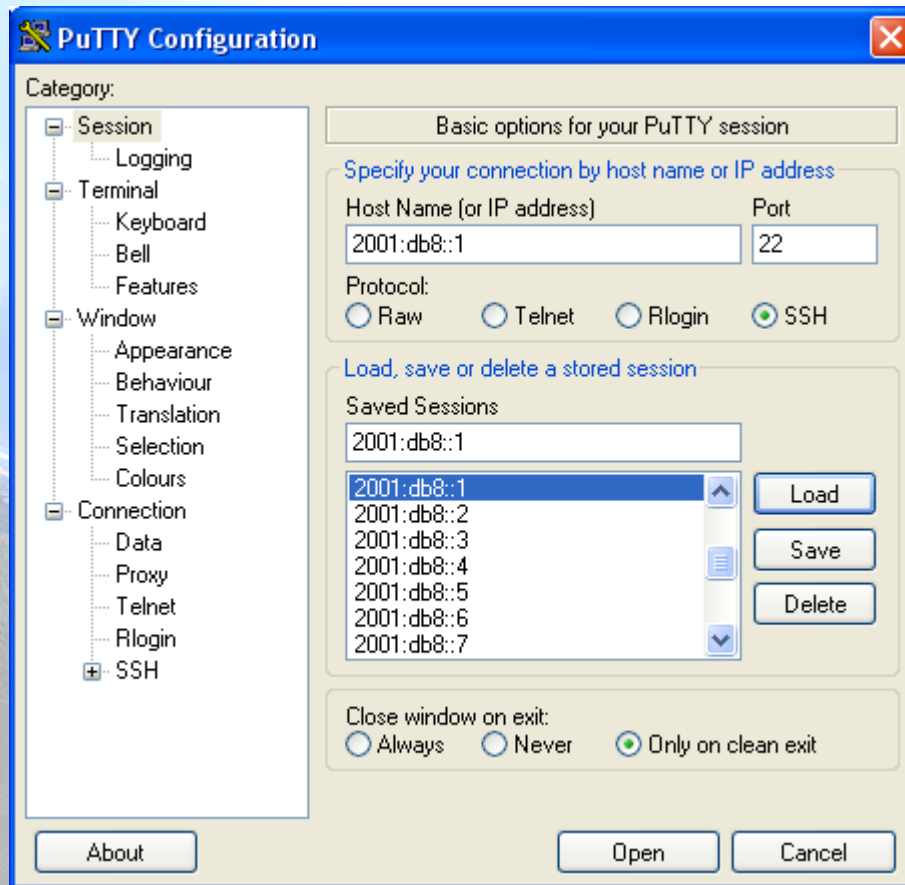


- **Putty**
- IPv4/IPv6 Client for Telnet and SSH
- Very useful for Administration and Management of devices
- Available at http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

# IPv6 Applications (4)

- **Ethereal and Wireshark**
- Captures y Decodes IPv4/IPv6 Traffic
- Very useful for connectivity validation and troubleshooting
- http://www.ethereal.com y http://www.wireshark.org
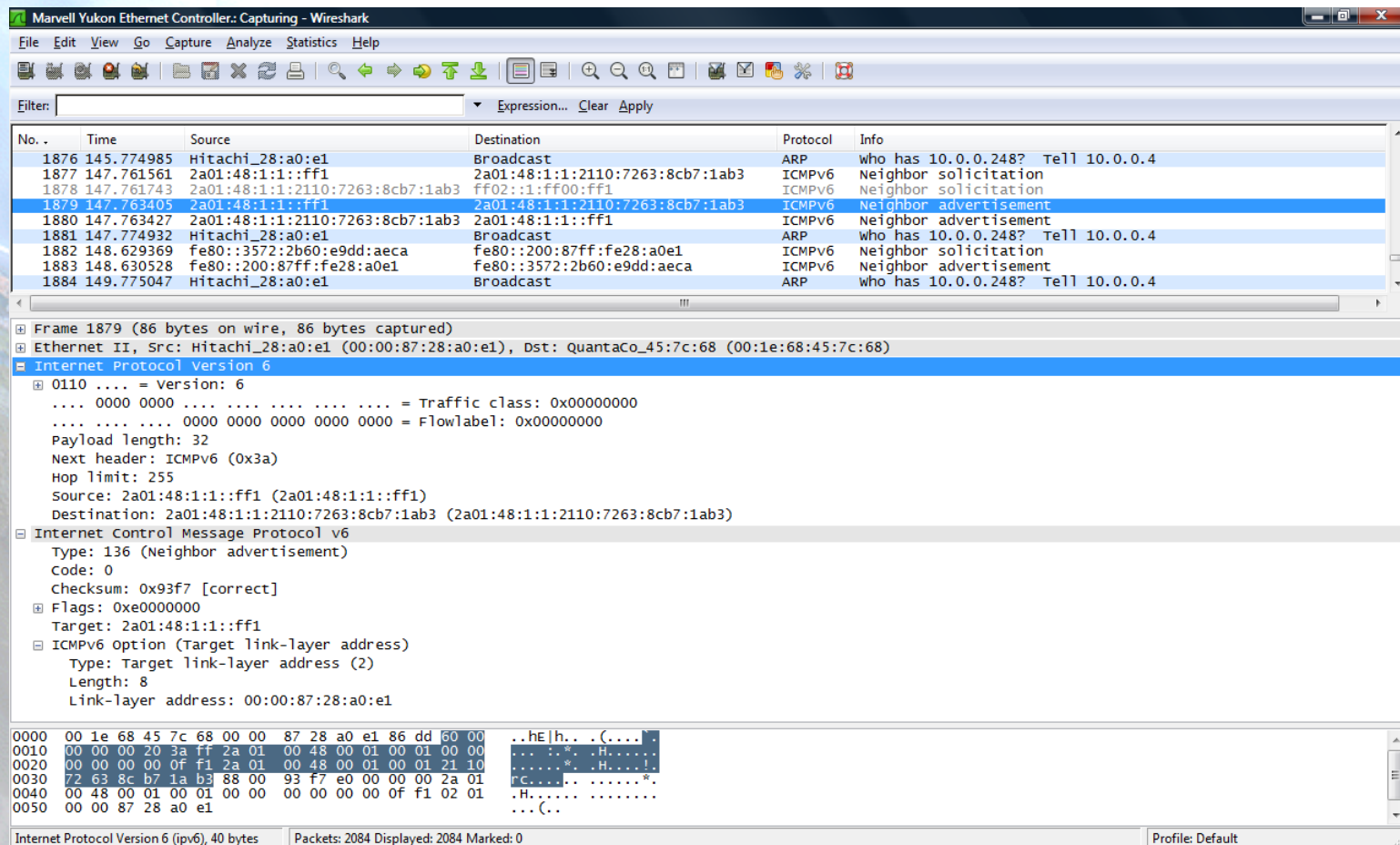
# IPv6 Applications (5)

- **VLC**
- Multimedia Client and Server
- Unicast y Multicast Support
- Available at http://www.videolan.org/vlc/
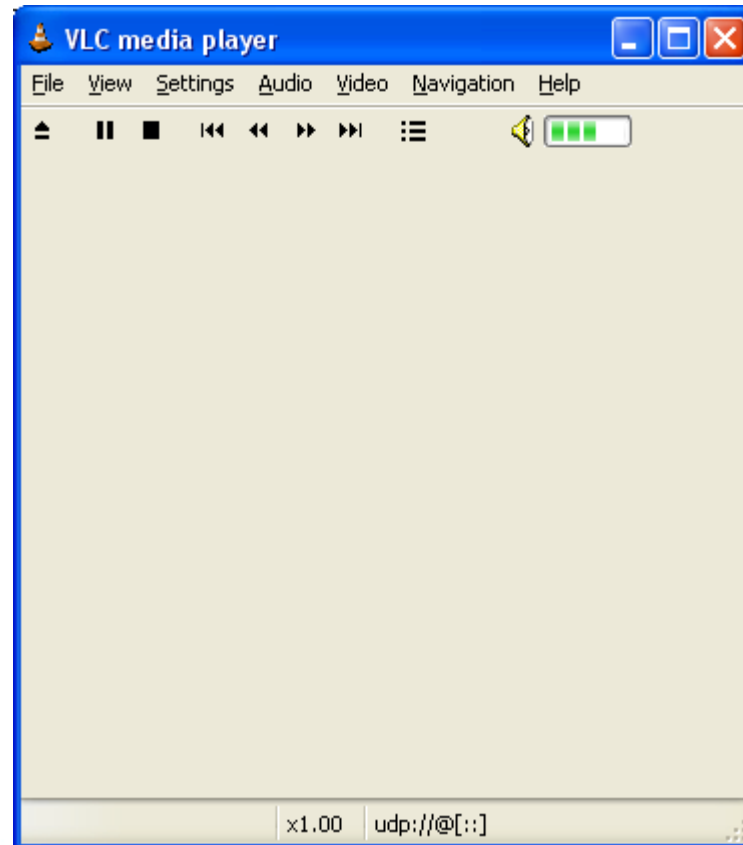
# IPv6 Applications (6)

- **Microsoft Windows Media Player and Server**

- Multimedia Client and Server

- It supports both IPv4/IPv6 Unicast/Multicast

- http://www.microsoft.com/windows/windowsmedia/default.aspx

WMP Client

# IPv6 Applications (7)

- **VNC**
  - Remote Access to a PC using IPv6
  - Graphic Environment

- Client/server Model
  - Server installed in remote PC which is the target
  - Client installed in local PC for remote access

- Supported for
  - Windows XP
  - Linux

- Available at
  - http://jungla.dit.upm.es/~acosta/paginas/vncIPv6.html

# IPv6 Applications (8)
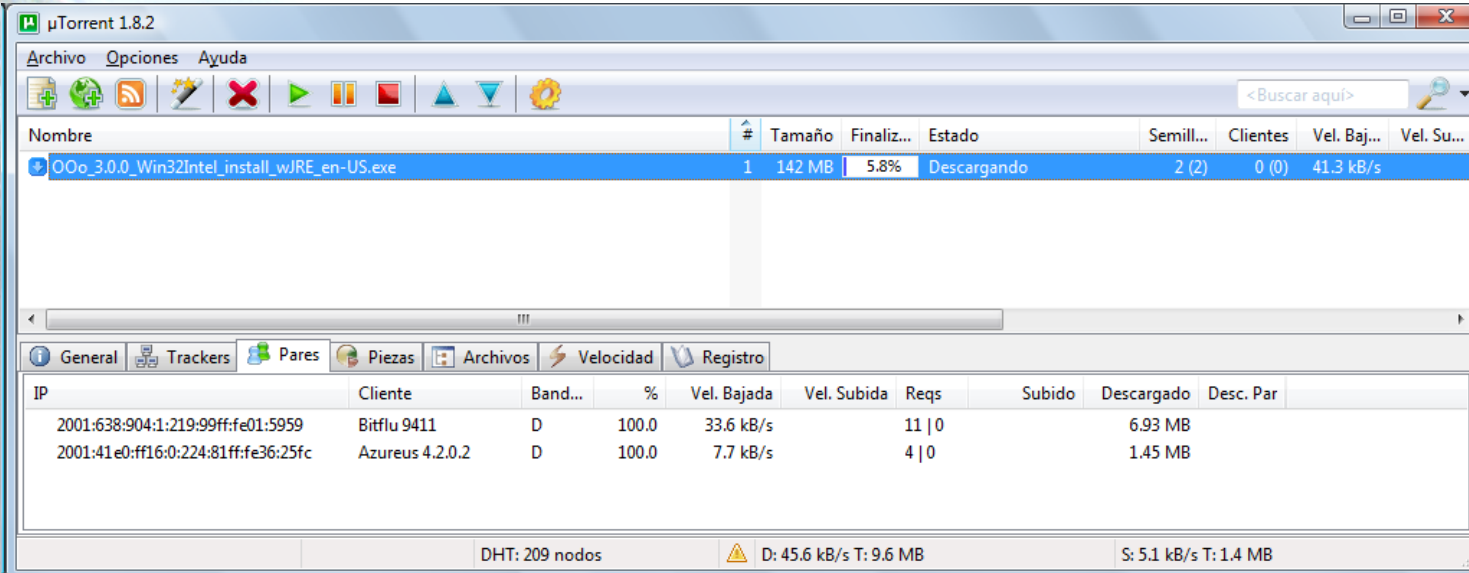
- **Web**
- The most common Clients: Firefox, IE, Konqueror, Opera, Safari
- Servers: Apache, IIS
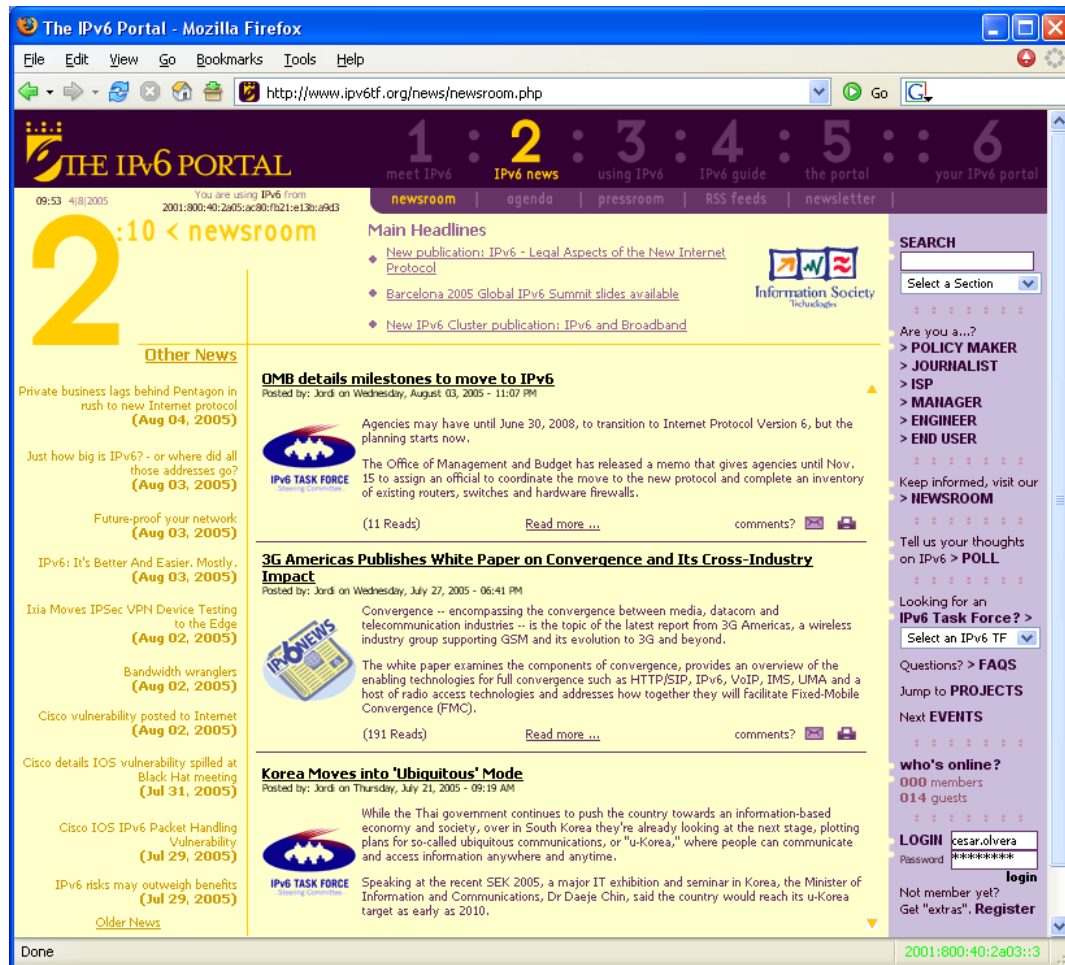
# IPv6 Applications (9)

- **BitTorrent**

- IPv6 Clients: Azureus, BitTornado, uTorrent, etc.

- Trackers IPv6
  - NIIF/HUNGARNET http://ipv6.niif.hu/index.php?mn=3&sm=6&lg=en
  - SixXS  http://www.sixxs.net/tools/tracker/
  - The Pirate Bay  http://thepiratebay.org/

# IPv6 Applications (10)

- **IPv6 to Standard**
  - List of devices, applications and services supporting IPv6
  - http://www.ipv6-to-standard.org

# IPv6 Applications (11)

- **FreeBSD**
- You can use FreeBSD ports:

  #>cd /usr/ports

  #>make search key="ipv6"

- A list of available IPv6 applications with IPv6 support will appear. Among the information of each application you can find the *path*, which is the folder where we will go and from where we can install the application:

  #>cd path

  #>make install

- This starts a search over different source code servers, from where the application will be downloaded, compiled and installed
- You can also download just the source code, that will be in /usr/ports/distfiles, using instead of make install, make fetch

# E1: IPv6 Applications (1)

- **Windows**

  **C:\>nslookup**

  >set type=a

  >www.ipv6tf.org

  Name:   www.ipv6tf.org

  Address:  213.172.48.141

  >set type=aaaa

  >www.ipv6tf.org

  www.ipv6tf.org  AAAA IPv6 address =
     2a01:48:1:0:2e0:81ff:fe05:4658

# E1: IPv6 Applications (2)

- **Linux:**

  **# dig a www.ipv6tf.org**

  ;; QUESTION SECTION:

  ;www.ipv6tf.org.          IN      A

  ;; ANSWER SECTION:

  www.ipv6tf.org.     172800  IN  A   213.172.48.141


- **# dig aaaa www.ipv6tf.org**

  ;; QUESTION SECTION:

  ;www.ipv6tf.org.           IN      AAAA

  ;; ANSWER SECTION:

  www.ipv6tf.org.   172800 IN  AAAA  2001:7f9:1000:1::103

# E1: IPv6 Applications (3)

- **Linux:**

  **#dig aaaa www.kame.net @2001:7f9:1000:1::103**

  ;; QUESTION SECTION:

  ;www.kame.net.          IN      AAAA

  ;; ANSWER SECTION:

  www.kame.net. 86400 IN  AAAA
      2001:200:0:8002:203:47ff:fea5:3085

  ;; Query time: 400 msec

  ;; SERVER: 2001:800:40:2a03::3#53
    (2001:800:40:2a03::3)

  ;; WHEN: Fri Jun 24 13:49:41 2005

  ;; MSG SIZE  rcvd: 107

# E2: IPv6 Applications

- To install (in case those are not already installed):

    - SSH Client with IPv6 support (Putty)

    - FTP Client (Command line on BSD, Linux, Windows)

    - Web Browser (Firefox, IE)

    - Ethereal

    - VLC

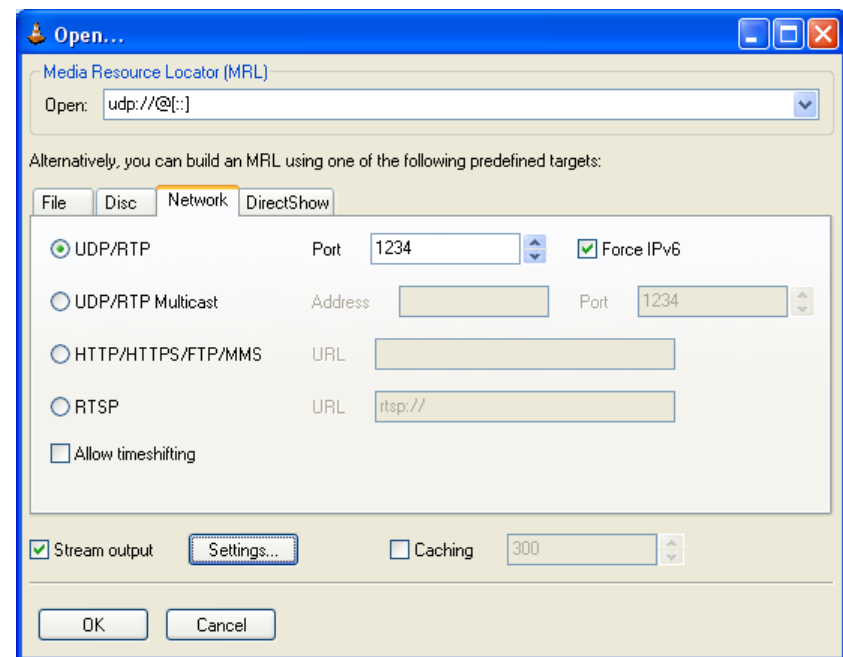    - VNC

# E3: IPv6 Applications

- To use the different services while Wireshark (or tcpdump) is used, in order to capture packets

- To use the SSH client to access by v4 or v6 choosing by means of DNS resolution

- To use the SSH client to access by v4 or v6 choosing by means of an application parameter (linux: #ssh -6|-4)(XP: ping -6|-4)

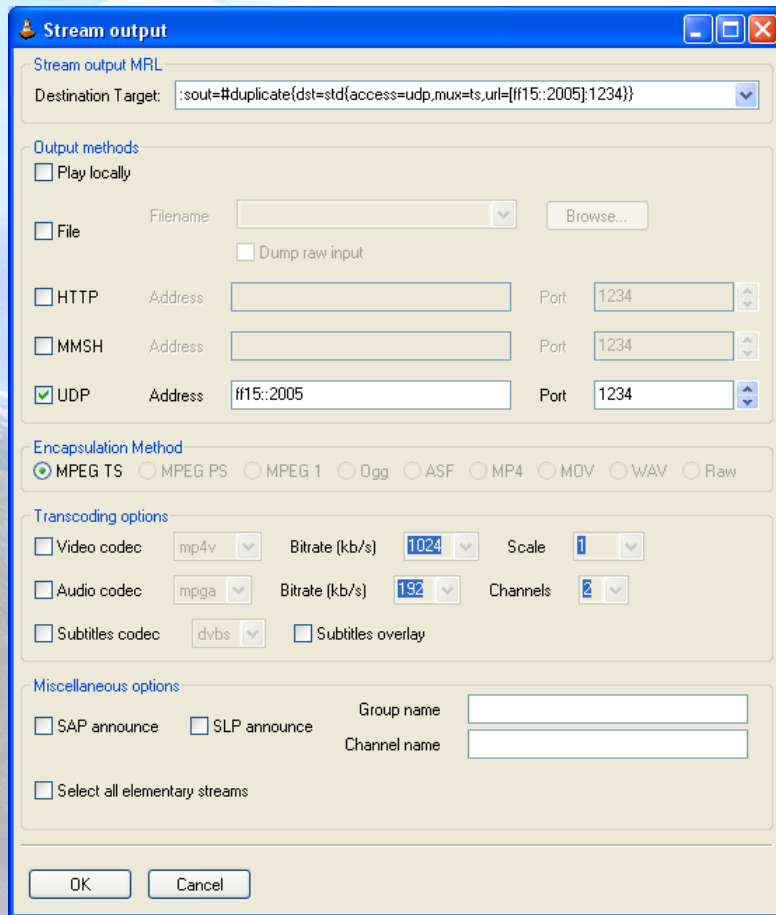# E4: IPv6 Applications (1)
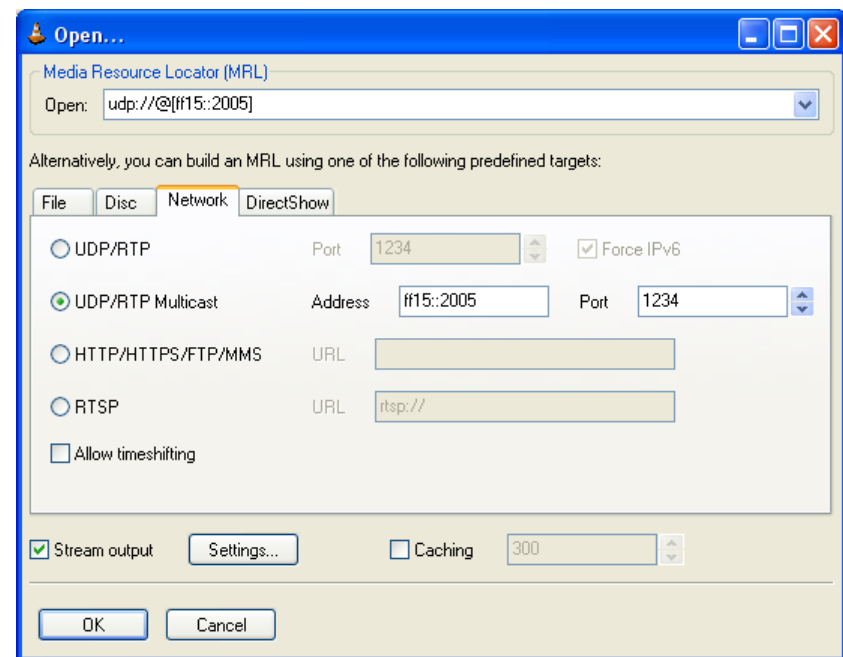
- VLC with Unicast
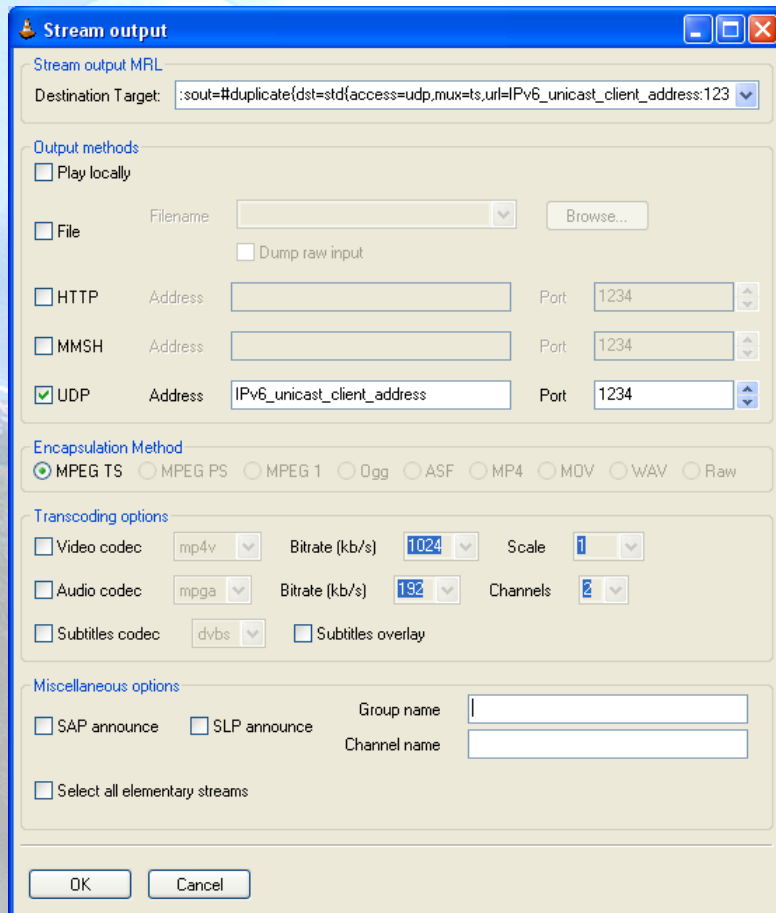


Server



Client

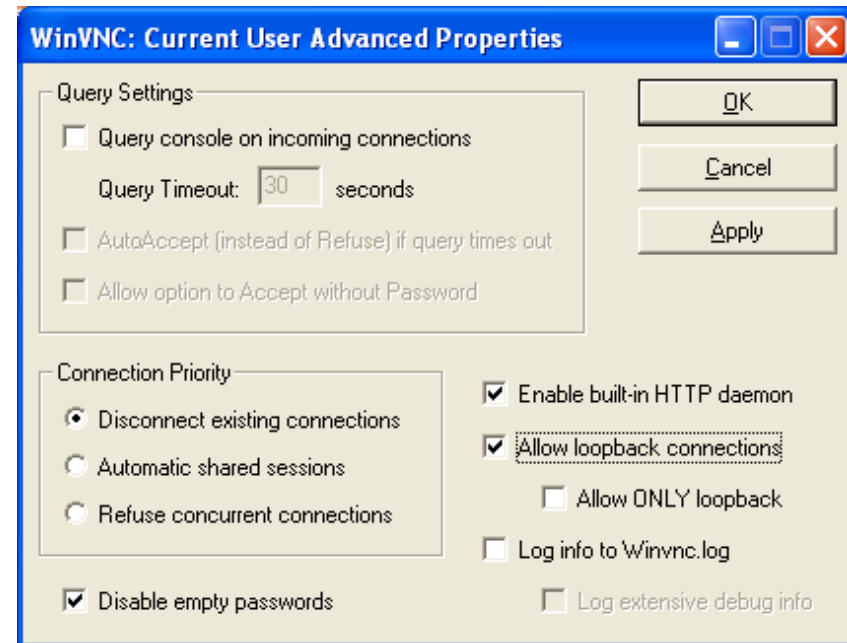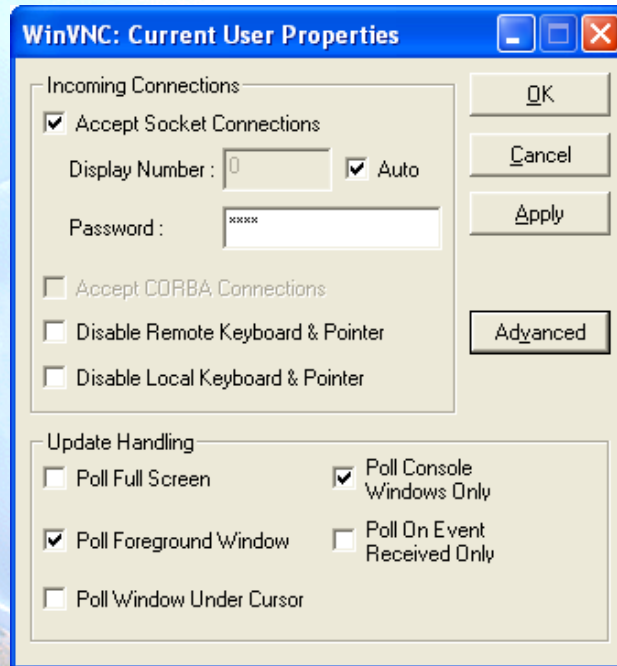# E4: IPv6 Applications (2)

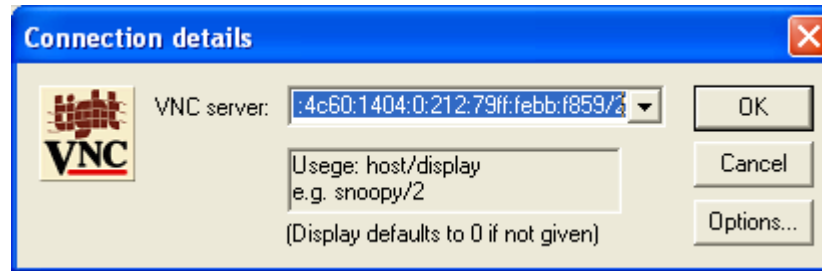- VLC with Multicast



Server



Client

# E5: IPv6 Applications (1)



- **VNC Server Properties**
  - It is needed to configure the "Display Number" so as to receive the connections
    - Default value is 0
  - It is needed to define a password
- VNC Server Properties = => Advanced
  - Also enable "allow loopback connections"

# E5: IPv6 Applications (2)



Connection details

VNC server: :4c60:1404:0:212:79ff:febb:f859/3    OK    Cancel    Options...

Usege: host/display
e.g. snoopy/2

(Display defaults to 0 if not given)

- **VNC client**
  - VNC server is specified trough
    - An IPv6 address
    - Or a DNS name
  - Then, the "Display" is added after the VNC server
    - It is specified by a number separate from VNC server with a '/'

# References (1)

- [6in4] RFC1933
- [TunAut] RFC1933
- [6to4] RFC3056
- [6over4] RFC2529
- [TB] RFC3053
- [TSP] draft-vg-ngtrans-tsp-01, http://www.hexago.com/index.php?pgID=step1
- [TEREDO] RFC4380
- [TEREDOC] http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/teredo.mspx
- [ISATAP] RFC4214
- [AYIYA ] draft-massar-v6ops-ayiya-02
- [SILKROAD ] draft-liumin-v6ops-silkroad-02
- [DSTM] draft-ietf-ngtrans-dstm-10
- [SIIT] RFC2765
- [NATPT] RFC2766, RFC4966
- [BIS] RFC2767
- [TRT] RFC3142
- [SOCKSv64 ] RFC3089

# References (2)

- [PROTO41] draft-palet-v6ops-proto41-nat-04
- [STUN] RFC3489
- [NATPTIMPL]
  - http://www.ipv6.or.kr/english/download.htm ==> Linux 2.4.0
  - http://www.ispras.ru/~ipv6/index_en.html ==> Linux y FreeBSD
  - http://research.microsoft.com/msripv6/napt.htm Microsoft
  - ftp://ftp.kame.net/pub/kame/snap/kame-20020722-freebsd46-snap.tgz ==> KAME snapshot (22.7.2002)
  - http://ultima.ipv6.bt.com/
- [STATELESS] RFC4862
- [STATEFULL] RFC3315
- [PRIVACY] RFC3041
- Windows IPv6
  - http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sag_ip_v6_add_utils.mspx
  - http://www.microsoft.com/technet/community/columns/cableguy/cg0902.mspx.

# Thanks !

**Contact:**

– **Jordi Palet Martínez (Consulintel): jordi.palet@consulintel.es**

**The IPv6 Portal:**

• **http://www.ipv6tf.org**