

IPv6 Startup

APNIC 20th
Hanoi, Vietnam
September 6th-9th, 2005

Miguel Angel Díaz (miguelangel.diaz@consulintel.es)

César Olvera (cesar.olvera@consulintel.es)

Jordi Palet (jordi.palet@consulintel.es)

Alvaro Vives (alvaro.vives@consulintel.es)

Agenda

1. IPv6 setup in several Platforms
(Windows XP/2003, Linux, BSD)
2. Basic Configuration, Stateless/Stateful
Autoconfiguration, Privacy, Static Routes
3. Transition Mechanisms Configuration
4. Examples of Applications



Part 1

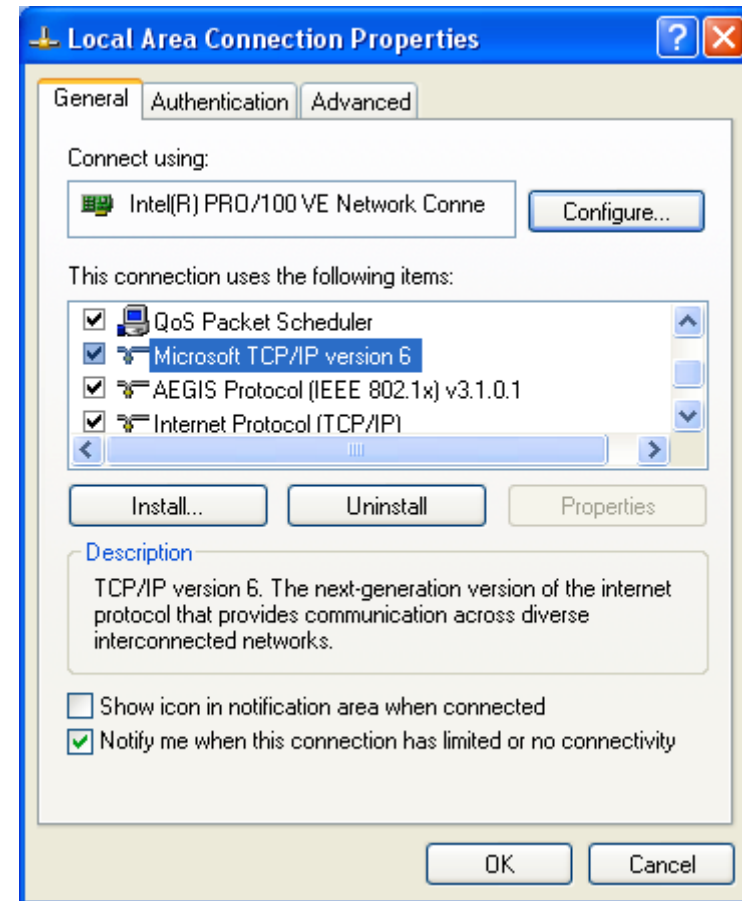
IPv6 Setup in several Platforms (Windows XP/2003, Linux, BSD)

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

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

IPv6 Setup: Linux (1)

- 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



Part 2

Basic Configuration Stateless/Stateful Autoconfiguration, Privacy, Static Routes

Basic Configuration: XP/2003 (1)

- Basic Commands in XP/2003
- 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 SP2)
 - 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** (starting on Windows XP SP2 and Server 2003)
 - Option **store=active|persistent** to save changes
- Equivalences at:
<http://www.microsoft.com/windowsserver2003/technologies/ipv6/ipv62netshtable.mspx>

Basic Configuration: XP/2003 (2)

- **“ipv6” Commands**
 - `ipv6 [-p] [-v] if [ifindex]`
 - `ipv6 [-p] ifcr v6v4 v4src v4dst [nd] [pmlid]`
 - `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: XP/2003 (3)

- **“netsh interface ipv6” Commands**
 - 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
 - install - Installs IPv6
 - isatap - Changes to the ‘netsh interface ipv6 isatap’ context
 - renew - Restarts IPv6 interfaces
 - reset - Resets IPv6 configuration state
 - set - Sets configuration information
 - show - Displays information
 - uninstall - Uninstalls IPv6

Basic Configuration: XP/2003 (4)

- **“netsh interface ipv6 add” Commands**
 - add 6over4tunnel - Creates a 6over4 interface.
 - add address - Adds an IPv6 address on an interface.
 - add dns - Adds a static DNS server address.
 - 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.
- **“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 mobility - Modifies mobility configuration parameters.
 - set prefixpolicy - Modifies prefix policy information.
 - set privacy - Modifies privacy configuration parameters.
 - set route - Modifies route parameters.
 - set state - Sets the state of deprecated functionality.
 - set teredo - Sets Teredo state.
- **“netsh interface ipv6 show” Commands**
 - show address - Shows IPv6 addresses.
 - show bindingcacheentries - Shows binding cache entries.
 - show destinationcache - Shows destination cache entries.
 - show dns - Displays the DNS server addresses.
 - show global - Shows global configuration parameters.
 - show interface - Shows interface parameters.
 - show joins - Shows IPv6 multicast addresses.
 - show mobility - Shows mobility configuration parameters.
 - show neighbors - Shows neighbor cache entries.
 - show prefixpolicy - Shows prefix policy entries.
 - show privacy - Shows privacy configuration parameters.
 - show routes - Shows route table entries.
 - show siteprefixes - Shows site prefix table entries.
 - show state - Shows the state of deprecated functionality.
 - show teredo - Shows Teredo service state.

Basic Configuration: XP/2003 (5)

- 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: XP/2003 (6)

- Ping in XP/2003
- **ping6 [-t] [-a] [-n count] [-l size] [-w timeout] [-s srcaddr] [-r] dest**
 - t Ping the specified host until interrupted
 - a Resolve addresses to hostnames
 - n count Number of echo requests to send
 - l size Send buffer size
 - w timeout Timeout in milliseconds to wait for each reply
 - s srcaddr Source address to use
 - r Use routing header to test reverse route also
- **ping** command default to IPv6 if available

Basic Configuration: XP/2003 (7)

- **Examples of Ping in XP/2003**
- **ping6 www.ipv6tf.org**

Pinging www.ipv6tf.org [2001:800:40:2a03::3]
from 2001:800:40:2a05:9c4d:b1cd:98d5:5a32 with 32 bytes of data:

Reply from 2001:800:40:2a03::3: bytes=32 time<1ms

Reply from 2001:800:40:2a03::3: bytes=32 time<1ms

Reply from 2001:800:40:2a03::3: bytes=32 time<1ms

Reply from 2001:800:40:2a03::3: bytes=32 time<1ms

Ping statistics for 2001:800:40:2a03::3:

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

Approximate round trip times in milli-seconds:

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

Basic Configuration: XP/2003 (8)

- **Examples of Ping in XP/2003**

- **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: XP/2003 (9)

- 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: XP/2003 (10)

- **Examples of Ping in XP/2003**
- **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

Basic Configuration: XP/2003 (11)

- Traceroute in XP/2003
- **tracert6 [-d] [-h maximum_hops] [-w timeout] [-s srcaddr] target_name**
 - d Do not resolve addresses to hostnames
 - h max_hops Maximum number of hops to search for target
 - w timeout Wait timeout milliseconds for each reply
 - s srcaddr Source address to use
 - r Use routing header to test reverse route also
- **tracert** command defaults to IPv6 when available

Basic Configuration: XP/2003 (12)

- **Examples of traceroute in XP/2003:**
- **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 3fe: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: XP/2003 (13)

- 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 **ipv6 if 5**

Basic Configuration: XP/2003 (14)

- 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 **ipv6 if 5**

Basic Configuration: XP/2003 (15)

- 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 **ipv6 if 5**

Basic Configuration: XP/2003 (16)

- 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: XP/2003 (17)

- 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: XP/2003 (18)

- 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: Linux (1)

- **Basic Commands (1)**

- ifconfig
- ping6 <hostcondirIPv6>|<dirIPv6>|[-I <interface>] <link-local-ipv6address>
- traceroute6 <hostcondirIPv6>|<dirIPv6>
- tracepath6 <hostcondirIPv6>|<dirIPv6>
- 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
```

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>
```

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>
```

- **Deleting a Route via an interface:**

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

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

Basic Configuration: Linux (6)

- 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_rlo="2001:618:10:4::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: 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:

```
2001:800:40:2a09:1:2:3:4 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
- [STATELESS] Provides information about:
 - Network Prefix
 - Routing
- Global Addresses are built by two elements
 - Interface Identifier (64 bits based on EUI-64, and usually obtained from IEEE 48 bit MAC Address)
 - Prefix obtained from the 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 XP/2003 hosts, it is enabled by default
- **ipconfig** o **ipv6 if** to check which is the autoconfigured address
- 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 (1)

- Configure a Linux router to send RA packets to the network
- Get a 'radvd' daemon for the used Linux distribution
 - <http://www.rpmfind.net/linux/rpm2html/search.php?query=radvd&submit=Search+...>
- Install it
- Enable routing capabilities
 - `echo 1 > /proc/sys/net/ipv6/conf/all/forwarding`
- Edit `/etc/radvd.conf` file with the following content:

Stateless: Exercise 1 (2)

```
interface eth00
{
    AdvSendAdvert on;

    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 5;

    AdvHomeAgentFlag off;

    prefix 2001:8500:40:2a30::/64
    {
        AdvOnLink off;
        AdvAutonomous on;
        AdvRouterAddr off;
    };
};
```

Stateless: Exercise 1 (3)

- Launch radvd daemon
 - Radvd
- Check that other computers in the network are autoconfigured thanks to our radvd daemon

Stateful Autoconfiguration (1)

- [STATEFUL] 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 still not available in the most common OSs
 - An specific installation of a DHCPv6 application is needed (server and/or client)
 - <http://klub.com.pl/dhcpv6/>
 - <http://sourceforge.net/projects/dhcpv6-linux/>

Stateful: Exercise 1 (1)

- Configuring a DHCPv6 server on Linux
 - Obtain the DHCPv6 implementation for Linux from:
<http://klub.com.pl/dhcpv6/dibbler/dibbler-0.4.0-linux.tar.gz>
 - Untar the file
 - `tar -xvzf dibbler-0.4.0-linux.tar.gz`
 - Make these directories
 - `/var/lib/dibbler`
 - `/etc/dibbler`

Stateful: Exercise 1 (2)

- Edit the content of file server.conf
 - log-level 7
 - log-mode short

 - iface eth0 {
 - T1 1000
 - T2 2000
 - class {
 - pool 2001:3820:40:2a03::10-2001:3820:40:2a03:ffff:ffff:ffff:ffff
 - }

 - option dns-server 2001:800:40:2a03::2, 2001:800:40:2a04::2
 - option domain example.com, test1.example.com

 - }
- The given addresses will be in the prefix 2001:3820:40:2a03::/64 starting from 2001:3820:40:2a03::10
- Copy the file server.conf in the directory /etc/dibbler
- Launch dhcpv6 server
 - dhcpv6-server run

Stateful: Exercise 2 (1)

- Configure DHCPv6 client in Linux
 - Get a DHCPv6 implementation for Linux from: <http://klub.com.pl/dhcpv6/dibbler/dibbler-0.4.0-linux.tar.gz>
 - Untar the file
 - `tar -xvzf dibbler-0.4.0-linux.tar.gz`
 - Create the directories
 - `/var/lib/dibbler`
 - `/etc/dibbler`

Stateful: Exercise 2 (2)

- Edit the content of file server.conf
 - log-mode short
 - iface eth0
 - {
 - IA
 - option dns-server
 - option domain
 - }
- With this configuration you get
 - An IPv6 address
 - DNS servers
 - Domain name
- Copy client.conf file in the directory /etc/dibbler
- Launch dhcpv6 client
 - dhcpv6-client run
- With 'ifconfig eth0' you can 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 (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 (2)

- In Windows XP/2003 hosts, it is enabled by default
- **ipconfig o ipv6 if** to check which is the autoconfigured address
- There are two ways to disable it:
 1. **netsh interface ipv6 set privacy state=disabled store=persistent**
 2. **ipv6 [-p] gpu UseTemporaryAddresses no**
- To check the change: “disable” and “enable” the physical interface on Windows Network Connection, then **ipconfig o ipv6 if**

Privacy (3)

- **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]



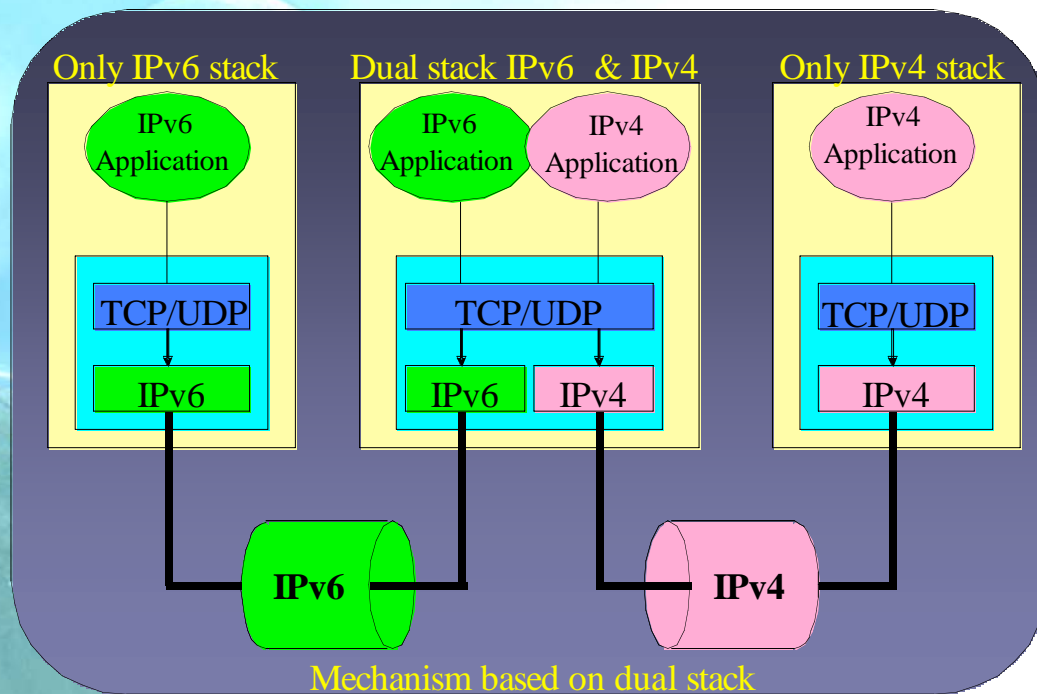
Part 3

Transition Mechanisms Configuration

Transition Mechanisms

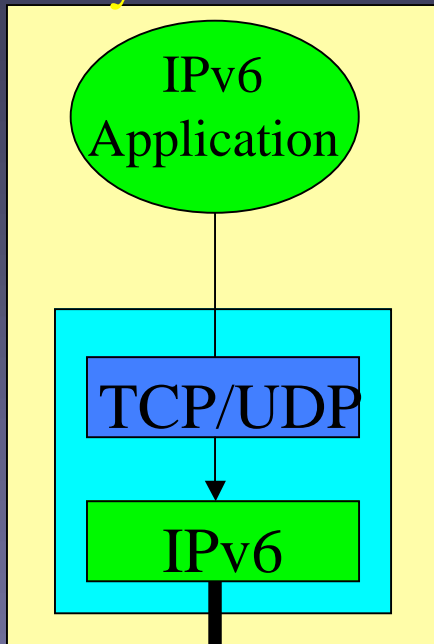
- IPv6 has been designed for easing the transition and coexistence with IPv4
- Several strategies have been designed for coexisting with IPv4 hosts
 - Dual stack: Simultaneous support for both IPv4 and IPv6 stacks
 - Tunnels: IPv6 packets encapsulated in IPv4 ones
 - This is the commonest choice
 - Translation: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

Dual Stack

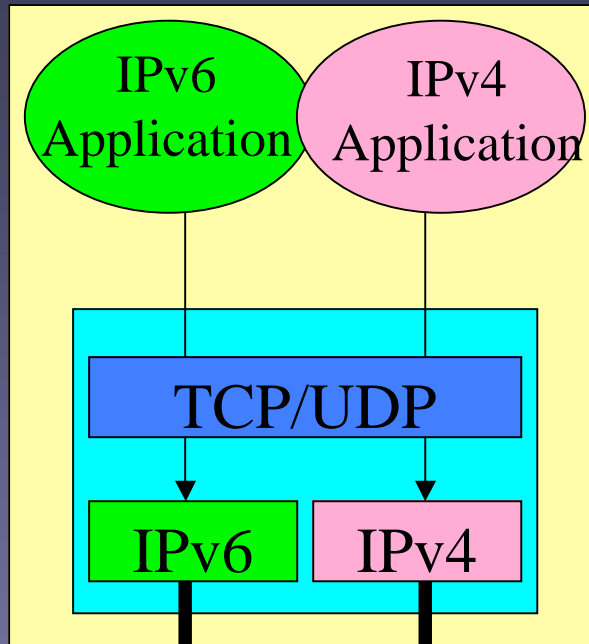


- All the hosts have both stacks IPv4 & IPv6
- IPv6-only communications ==> IPv6 stack, assuming IPv6 network support
- IPv4-only communications ==> IPv4 stack

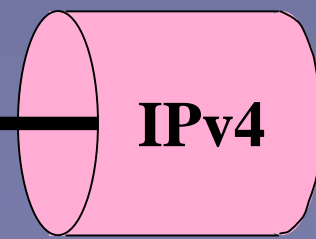
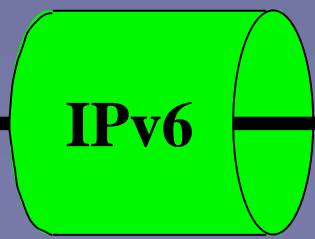
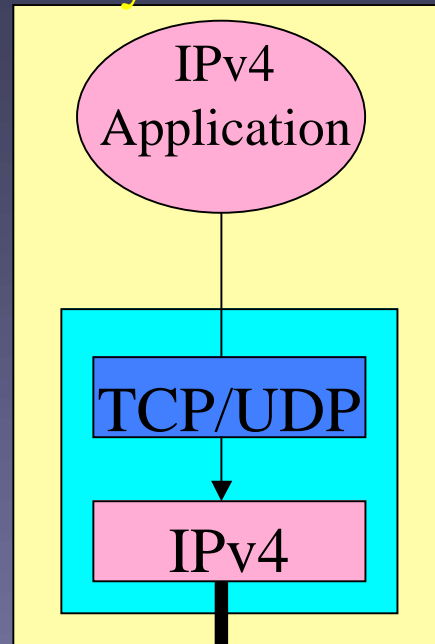
Only IPv6 stack



Dual stack IPv6 & IPv4

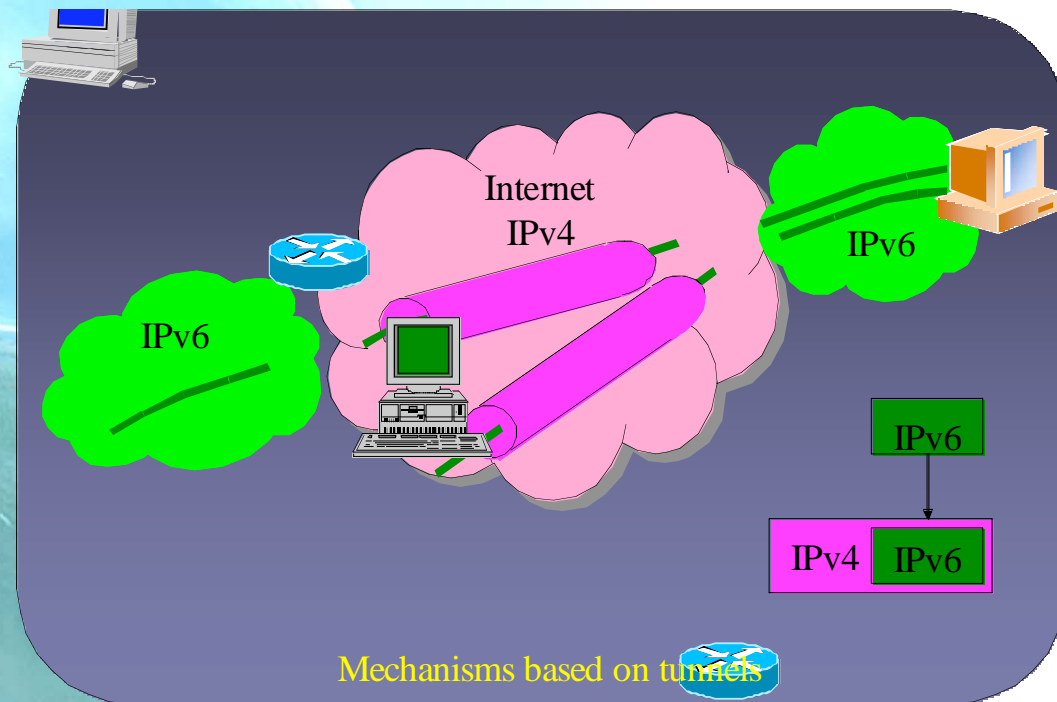


Only IPv4 stack

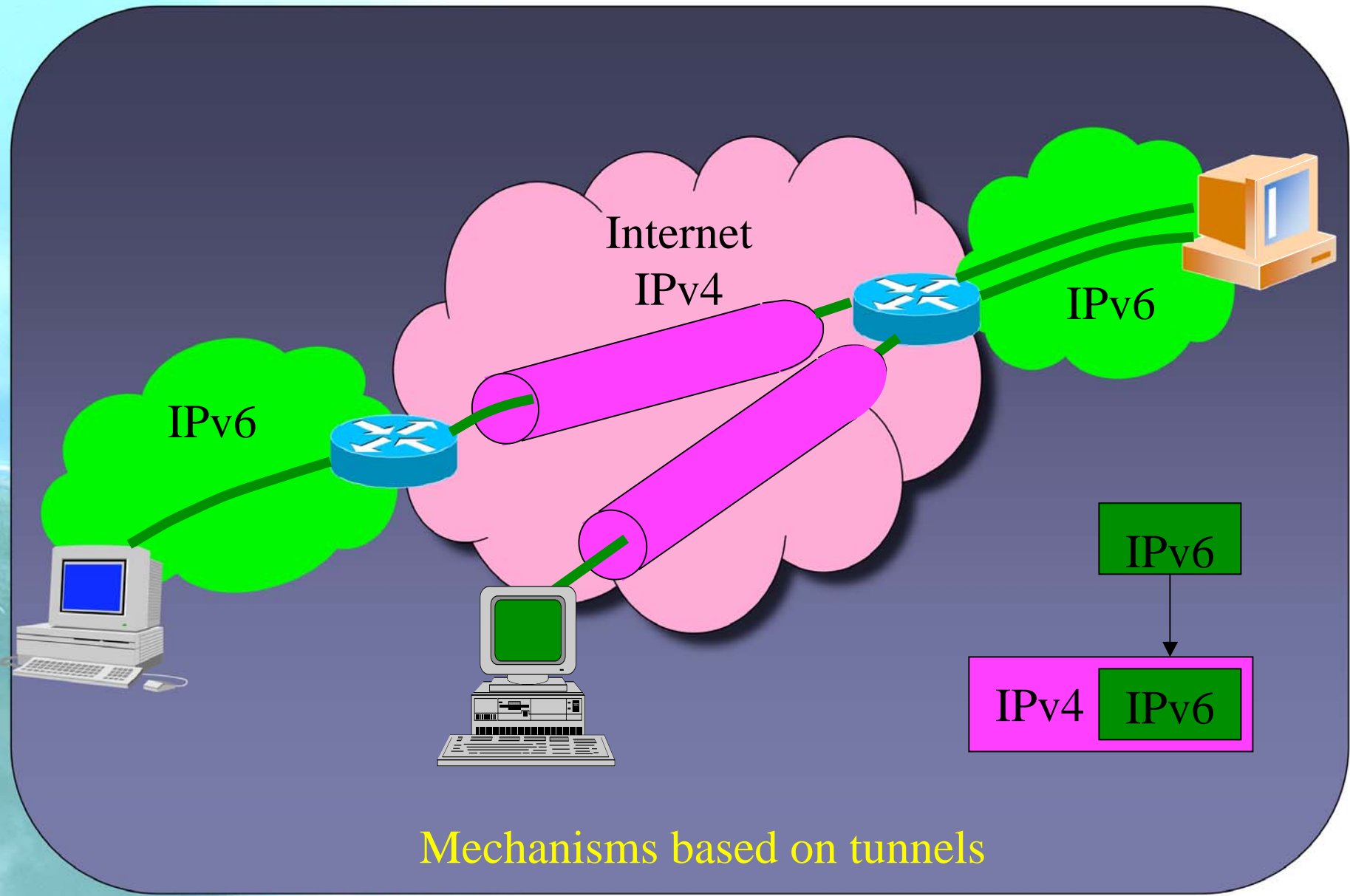


Mechanism based on dual stack

Tunnels IPv6 over IPv4 (1)



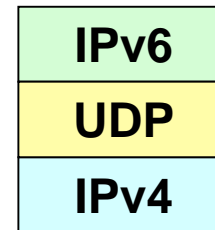
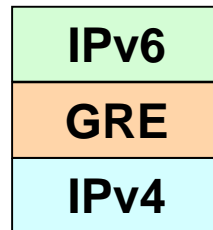
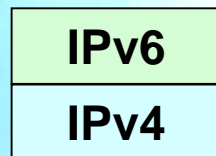
- It is used to provide IPv6 connectivity in IPv4-only networks
- The IPv6 packets are encapsulated into IPv4 packets
- There are different ways to make the encapsulation
 - 6in4, 6to4, 6over4, UDP, etc.
- The resulting packets flow through IPv4 networks towards the tunnel end point (TEP)



Mechanisms based on tunnels

Tunnels IPv6 over IPv4 (2)

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



- Same for IPv4 being used in IPv6-only networks

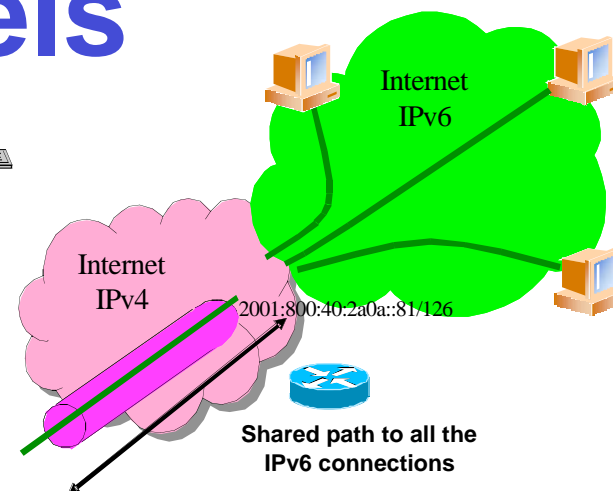
Tunnels IPv6 over IPv4 (3)

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

6in4 Tunnels

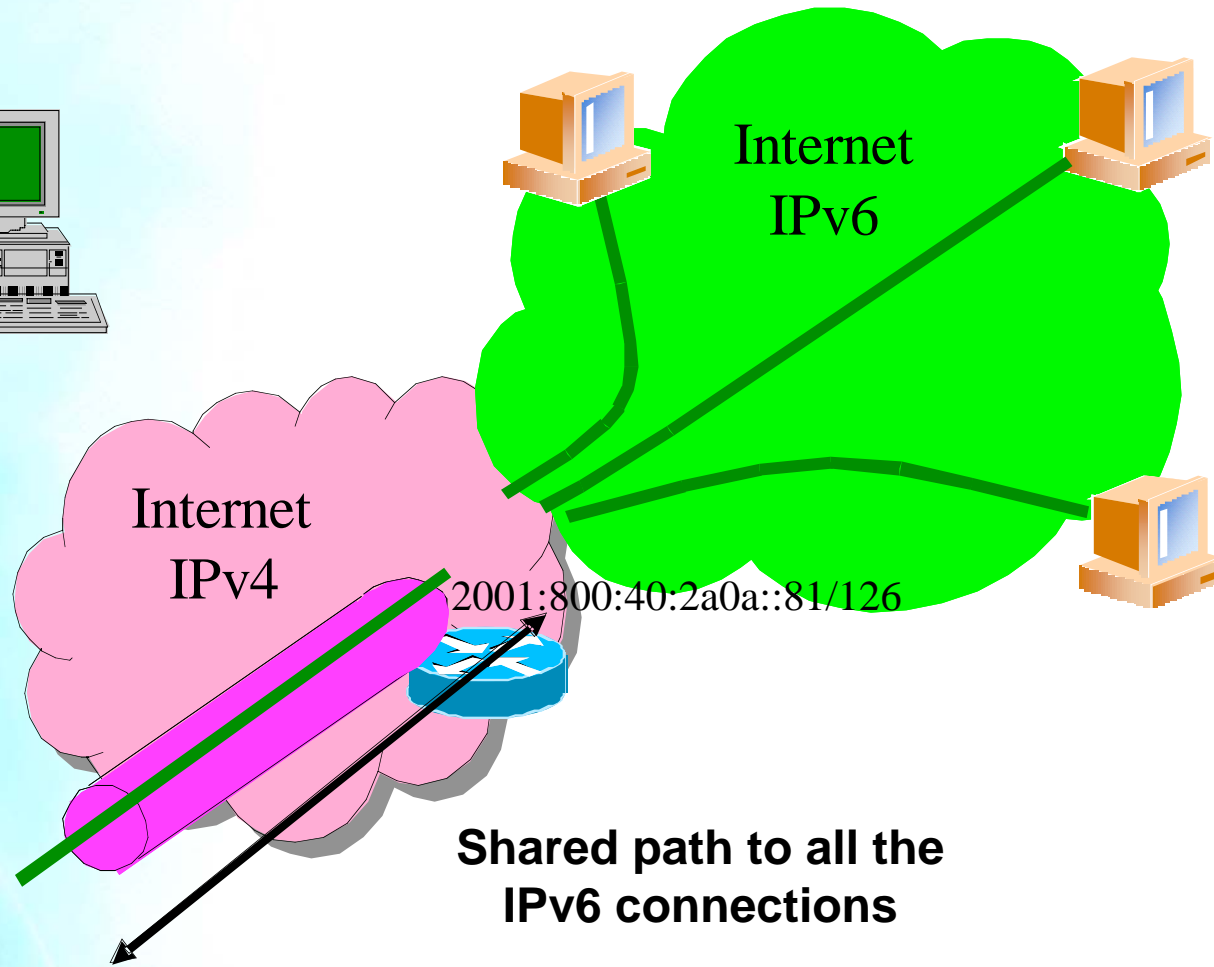
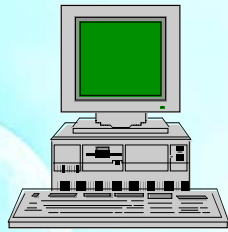


- It encapsulates directly the IPv6 packet into the IPv4 packet
- It is usually used between:
 - end host ==> router
 - router ==> router



- 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

2001:800:40:2a0a::82/126

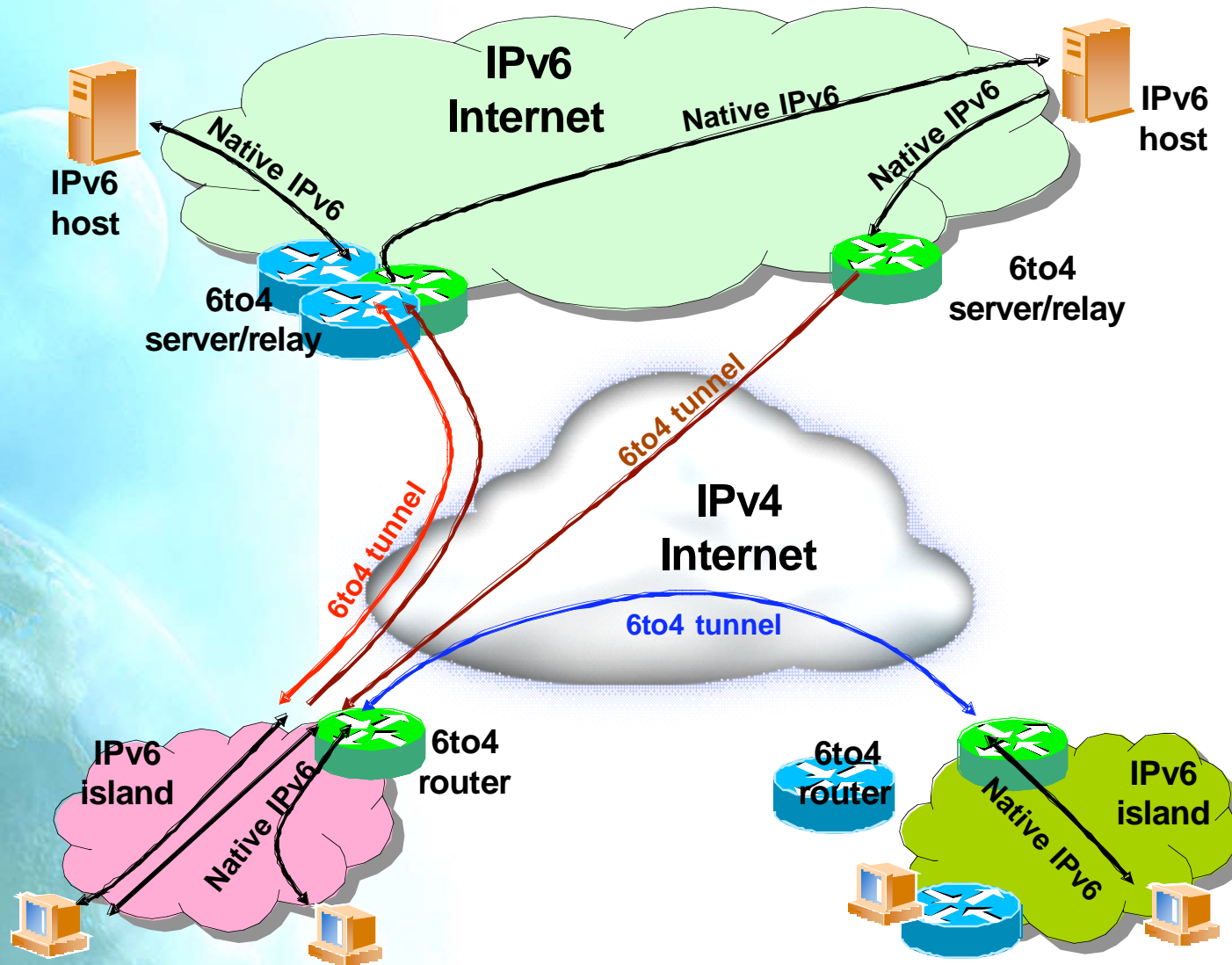


`2001:800:40:2a0a::82/126`

Tunnel Broker

- The 6in4 tunnels require the manual configuration of the devices involved in the tunnel creation
- To ease the address assignment and the IPv6 tunnel creation, the Tunnel Broker (TB) concept has been developed
 - It is an 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 build the IPv6 tunnel. However, the concept is similar to the one previously enounced

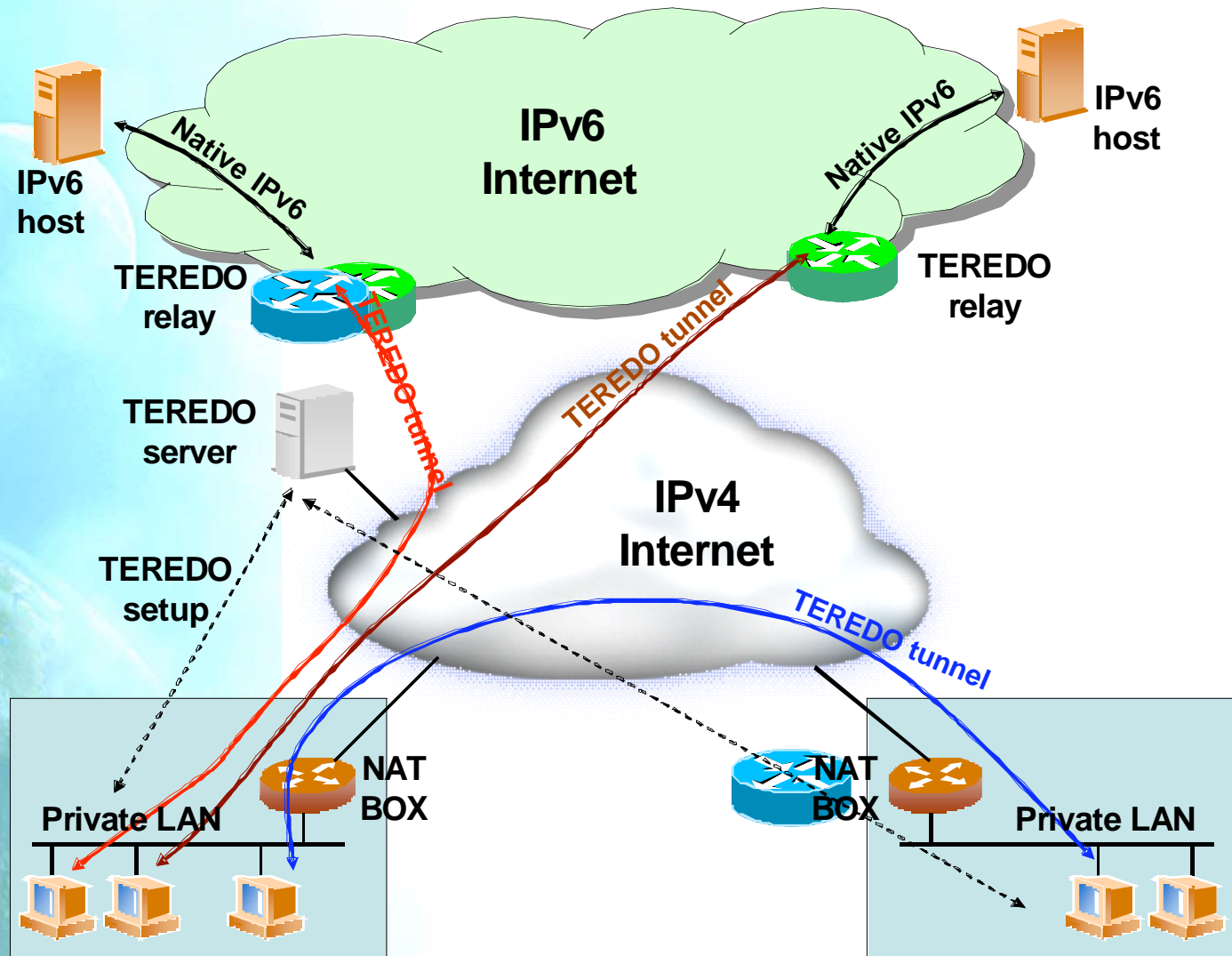
6to4 Tunnels (1)



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"

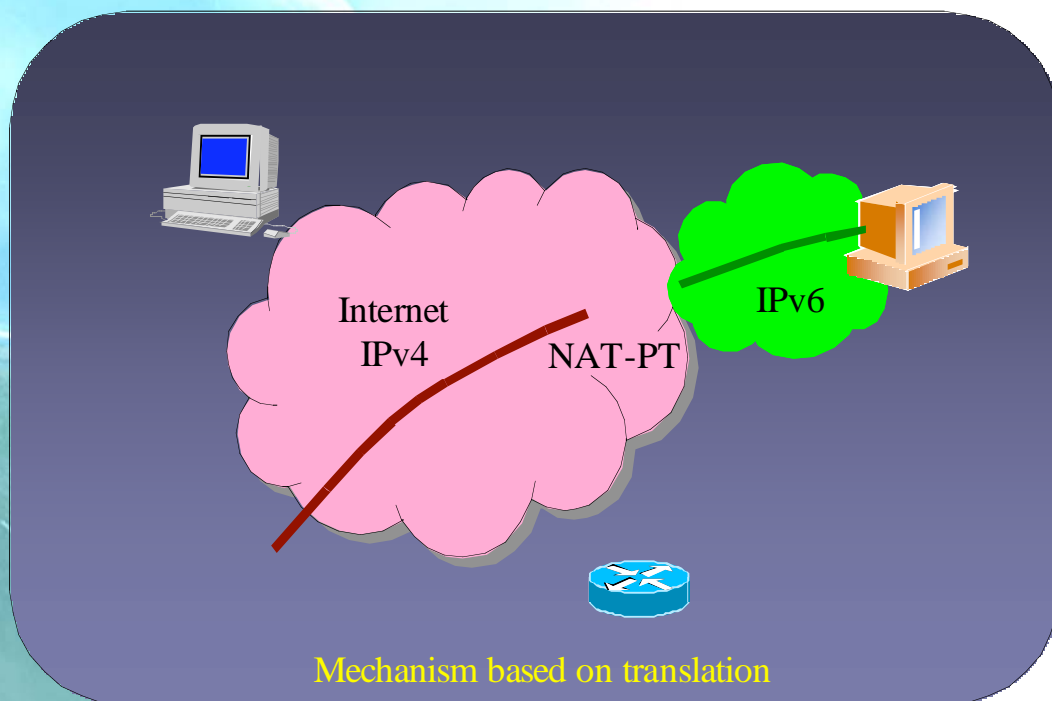
Teredo (1)



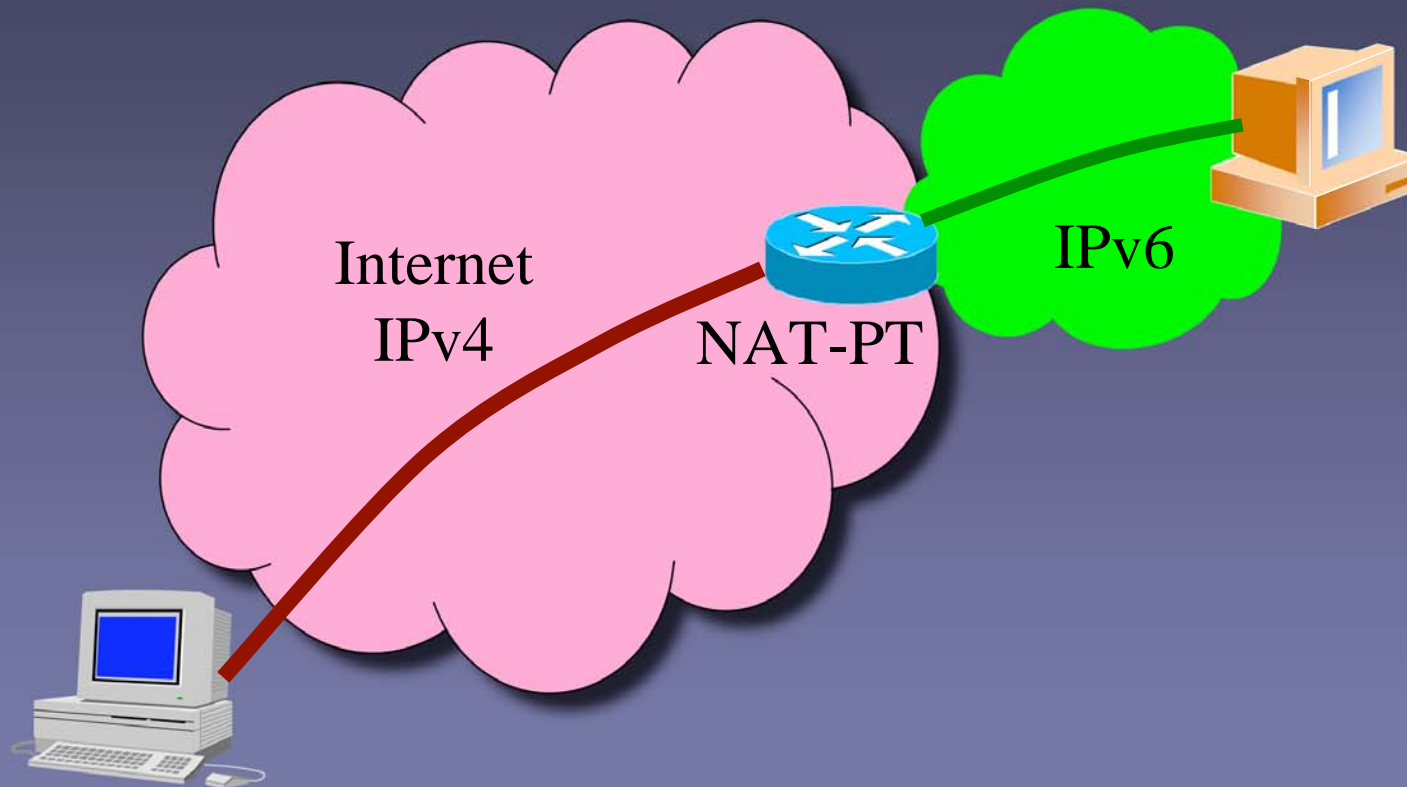
Teredo (2)

- 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 [STUN]:
 - Full Cone
 - Restricted Cone
- It does not work in the following NAT type:
 - Symmetric
- 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 3FFE:831F::/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

Translation



- 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.



Mechanism based on translation

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 2003)
- E6: Setup a Teredo Client (Windows XP/2003)
- E7: 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 ==> 2001:10:20:30::12/126
 - IPv6 gateway address ==> 2001:10:20:30::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 ==> 2001:10:20:30::11/126
 - IPv6 gateway address ==> 2001:10:20:30::12/126
 4. Check IPv6 connectivity between both alumni
 - Alumni A ==> ping6 IPv6_Address_Alumna_B
 - Alumni B ==> ping6 IPv6_Address_Alumna_A
- (*) 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
 - 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

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: Setting-Up a 6to4 Relay (Windows 2003)

- The 6to4 Relay configuration is very ease in case of Windows 2003
 - 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
 - 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 ping6 and traceroute6 (ping and tracert on Windows XP/2003) to check IPv6 connectivity

E6: Setting-Up a Teredo Client (Windows XP/2003)

- There are other Teredo implementations for other Operating Systems such as:
 - Linux: <http://www.simpnalempin.com/dev/miredo/>
 - FreeBSD: <http://www-rp.lip6.fr/teredo/>
- Windows XP/2003 presents 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.ipv6.vol.cz
 - teredo.ipv6.wind.com
 - teredo.via.ecp.fr
- 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: Use of IPv4/IPv6 Proxies (1)

- An IPv4/IPv6 proxy is not the same that a transition mechanism based on translation (NAT-PT)
- The 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
- So, 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

E7: 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

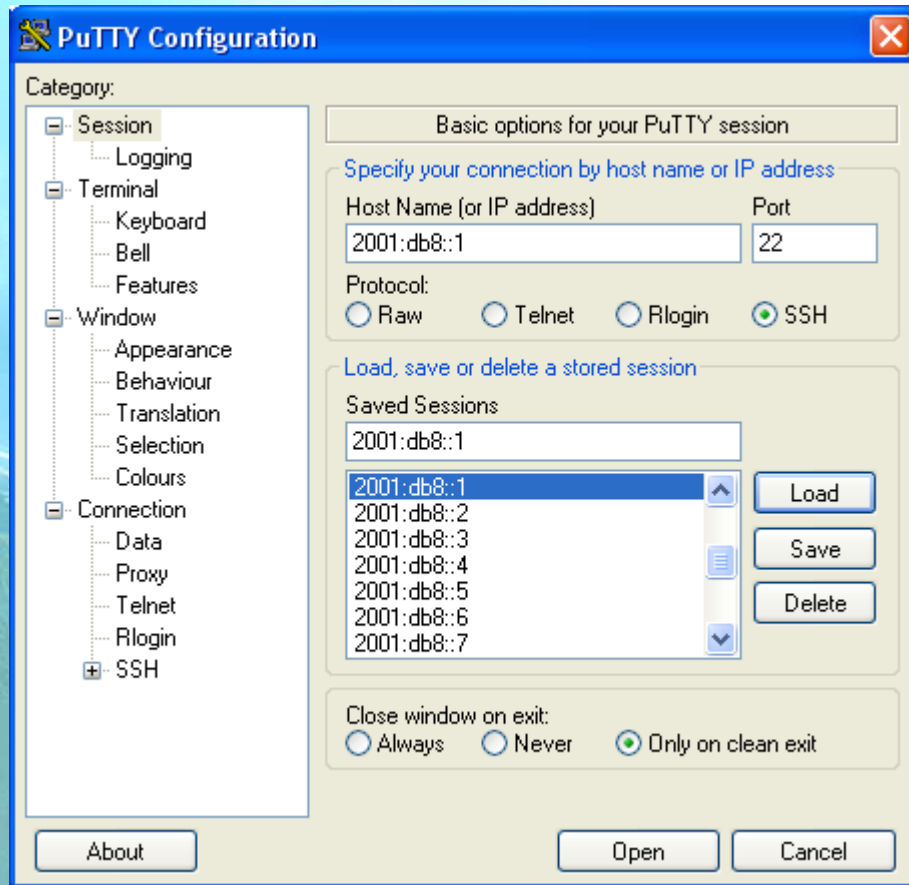
IPv6 Applications (1)

- 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
- Captures y Decodes IPv4/IPv6 Traffic
- Very useful for connectivity validation and troubleshooting
- Available at <http://www.ethereal.com/download.html>

The screenshot displays the Ethereal interface with a packet capture on the Intel(R) PRO/100 VE Network Connection. The main packet list shows various traffic, with packet 106 highlighted. The details pane for packet 106 shows an Ethernet II frame containing an Internet Protocol version 6 (IPv6) packet. The IPv6 packet is an ICMPv6 Neighbor Solicitation (Type 135) with a source address of 2001:800:40:2a05::1 and a destination address of 2001:800:40:2a05:7975:8ec8:5897:4c94. The payload length is 32 bytes. The bottom of the window shows the raw packet data in hexadecimal and ASCII.

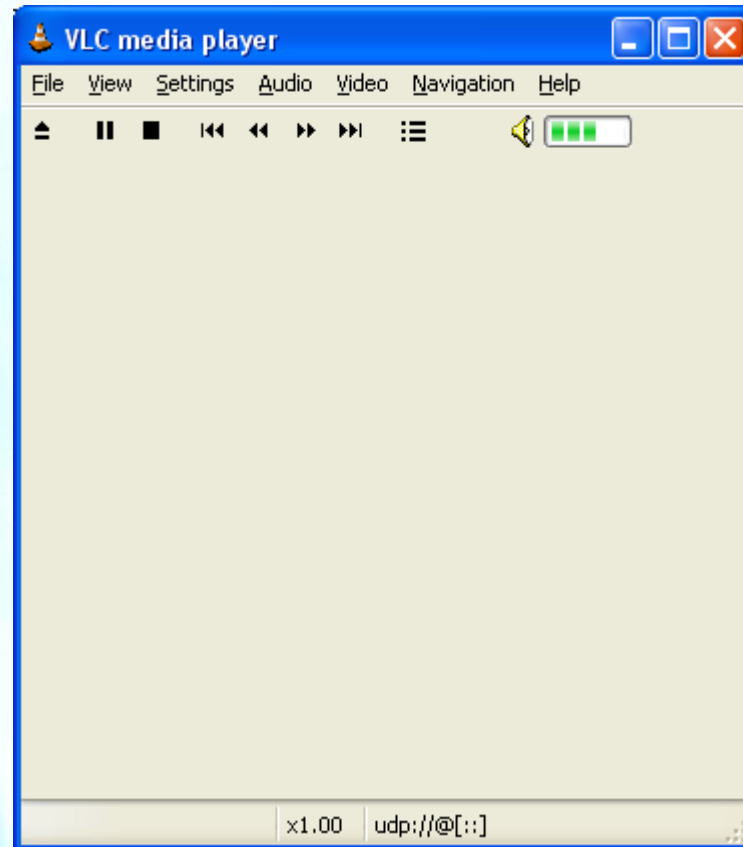
No.	Time	Source	Destination	Protocol	Info
99	18.041823	2001:200:0:8002:203:47ff:fea5:3085	2001:800:40:2a05:7975:8ec8:5897:4c94	HTTP	HTTP/1.1 304 Not Modified
100	18.043191	2001:800:40:2a05:7975:8ec8:5897:4c94	2001:200:0:8002:203:47ff:fea5:3085	HTTP	GET /logo/1/images/kame3.png HTTP/1.1
101	18.069029	2001:200:0:8002:203:47ff:fea5:3085	2001:800:40:2a05:7975:8ec8:5897:4c94	HTTP	HTTP/1.1 304 Not Modified
102	18.185625	2001:800:40:2a05:7975:8ec8:5897:4c94	2001:200:0:8002:203:47ff:fea5:3085	TCP	1234 > http [ACK] Seq=1087 Ack=557 Win=16724 Len=0
103	18.356567	2001:200:0:8002:203:47ff:fea5:3085	2001:800:40:2a05:7975:8ec8:5897:4c94	HTTP	HTTP/1.1 304 Not Modified
104	18.487387	2001:800:40:2a05:7975:8ec8:5897:4c94	2001:200:0:8002:203:47ff:fea5:3085	TCP	1235 > http [ACK] Seq=663 Ack=390 Win=16891 Len=0
105	19.112371	fe80::201:4aff:fe18:26c7	ff02::1:ff97:4c94	ICMPv6	Multicast listener report
106	22.829347	2001:800:40:2a05::1	2001:800:40:2a05:7975:8ec8:5897:4c94	ICMPv6	Neighbor solicitation
107	22.829384	2001:800:40:2a05:7975:8ec8:5897:4c94	2001:800:40:2a05::1	ICMPv6	Neighbor advertisement
108	29.559766	fe80::200:87ff:fe28:a0e0	ff02::d	PIMv2	Hello
109	31.291516	2001:800:40:2a05:7975:8ec8:5897:4c94	2001:630:d0:131:230:48ff:fe51:564d	TCP	1236 > http [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1440
110	31.341843	2001:630:d0:131:230:48ff:fe51:564d	2001:800:40:2a05:7975:8ec8:5897:4c94	TCP	http > 1236 [SYN, ACK] Seq=0 Ack=1 Win=5760 Len=0 MSS=14
111	31.341878	2001:800:40:2a05:7975:8ec8:5897:4c94	2001:630:d0:131:230:48ff:fe51:564d	TCP	1236 > http [ACK] Seq=1 Ack=1 Win=17280 Len=0
112	31.342115	2001:800:40:2a05:7975:8ec8:5897:4c94	2001:630:d0:131:230:48ff:fe51:564d	HTTP	GET / HTTP/1.1
113	31.391834	2001:630:d0:131:230:48ff:fe51:564d	2001:800:40:2a05:7975:8ec8:5897:4c94	TCP	http > 1236 [ACK] Seq=1 Ack=418 Win=6432 Len=0
114	31.635986	2001:630:d0:131:230:48ff:fe51:564d	2001:800:40:2a05:7975:8ec8:5897:4c94	HTTP	HTTP/1.1 200 OK[Unreassembled Packet]
115	31.637772	2001:630:d0:131:230:48ff:fe51:564d	2001:800:40:2a05:7975:8ec8:5897:4c94	HTTP	Continuation or non-HTTP traffic
116	31.637807	2001:800:40:2a05:7975:8ec8:5897:4c94	2001:630:d0:131:230:48ff:fe51:564d	TCP	1236 > http [ACK] Seq=418 Ack=2881 Win=17280 Len=0
117	31.690628	2001:630:d0:131:230:48ff:fe51:564d	2001:800:40:2a05:7975:8ec8:5897:4c94	HTTP	Continuation or non-HTTP traffic

Frame 106 (86 bytes on wire, 86 bytes captured)
Ethernet II, Src: 00:00:87:28:a0:e0, Dst: 00:01:4a:18:26:c7
Internet Protocol version 6
Version: 6
Traffic class: 0x00
Flowlabel: 0x00000
Payload length: 32
Next header: ICMPv6 (0x3a)
Hop limit: 255
Source address: 2001:800:40:2a05::1
Destination address: 2001:800:40:2a05:7975:8ec8:5897:4c94
Internet Control Message Protocol v6
Type: 135 (Neighbor solicitation)
Code: 0

```
0000 00 01 4a 18 26 c7 00 00 87 28 a0 e0 86 dd 60 00  ..J.&... .(....
0010 00 00 00 20 3a ff 20 01 08 00 00 40 2a 05 00 00  @*..yu
0020 00 00 00 00 00 01 20 01 08 00 00 40 2a 05 79 75  ..X.L...
0030 8e c8 58 97 4c 94 87 00 fd f3 00 00 00 00 20 01  ...@*.yu ..X.L...
0040 08 00 00 40 2a 05 79 75 8e c8 58 97 4c 94 01 01  ..X.L...
0050 00 00 87 28 a0 e0
```

IPv6 Applications (5)

- **VLC**
- Multimedia Client and Server
- Unicast y Multicast Support
- Available at <http://www.videolan.org/vlc/>



IPv6 Applications (6)

- **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 (7)

- Web
- The most common Clients: Firefox, IE, Konqueror, Opera, Safari
- Servers: Apache 2, IIS

The screenshot shows the 'The IPv6 Portal' website in a Mozilla Firefox browser window. The browser's address bar displays 'http://www.ipv6tf.org/news/newsroom.php'. The website's header features a large IPv6 address '1:2:3:4:5:6' and navigation links for 'newsroom', 'agenda', 'pressroom', 'RSS feeds', and 'newsletter'. The main content area is titled 'Main Headlines' and includes several news items with brief descriptions and dates. On the right side, there is a 'SEARCH' section with a dropdown menu for 'Select a Section' and a list of user roles including 'POLICY MAKER', 'JOURNALIST', 'ISP', 'MANAGER', 'ENGINEER', and 'END USER'. At the bottom right, there is a 'LOGIN' section with fields for 'cesar.olvera' and a password, along with a 'login' button and a 'Register' link. The browser's status bar at the bottom shows 'Done' and the IP address '2001:800:40:2a03:3'.

IPv6 Applications (8)

- **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

IPv6 Applications: Exercise 1 (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 =  
2001:800:40:2a03::3
```

IPv6 Applications: Exercise 1 (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 2001:800:40:2a03::3
```

IPv6 Applications: Exercise 1 (3)

- **Linux:**

```
#dig aaaa www.kame.net @2001:800:40:2a03::3
;; 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
```

IPv6 Applications: Exercise 2

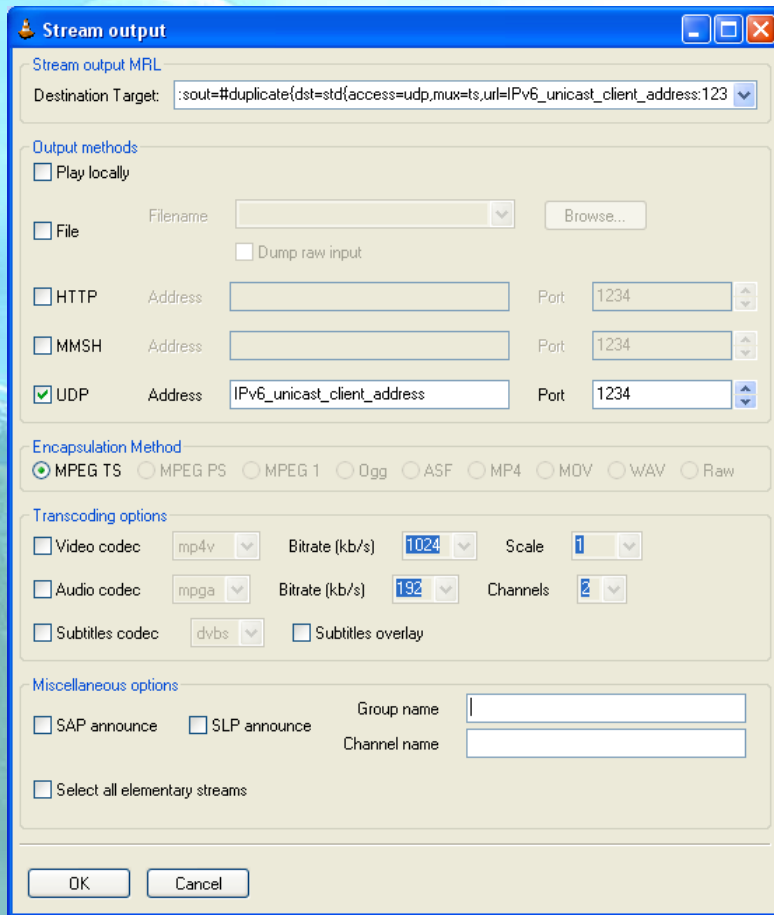
- To install (in case those are not already installed):
 - SSH Client with IPv6 support (Putty)
 - FTP Client (Command line on BSD, Linux, Windows)
 - Web Browser (Firefox, IE)
 - Ethereal
 - VLC
 - VNC

IPv6 Applications: Exercise 3

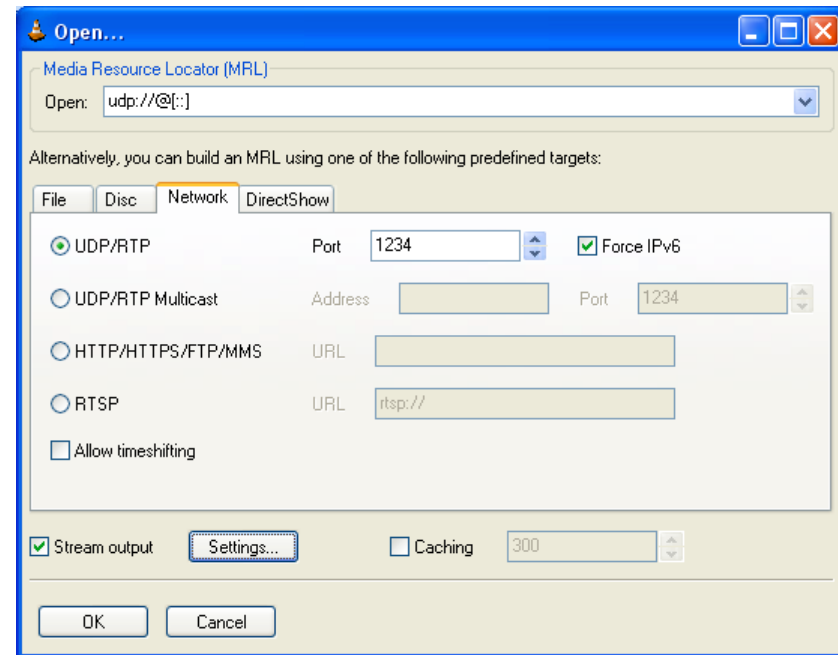
- To use the different services while Ethereal (or tcpdump) is used, in order to capture packets
- To use the SSH client to access by v4 or v6 choosing by means of DNS resolution
- To use the SSH client to access by v4 or v6 choosing by means of an application parameter (linux: #ssh -6|-4)(XP: ping -6|-4)

IPv6 Applications: Exercise 4 (1)

- VLC with Unicast



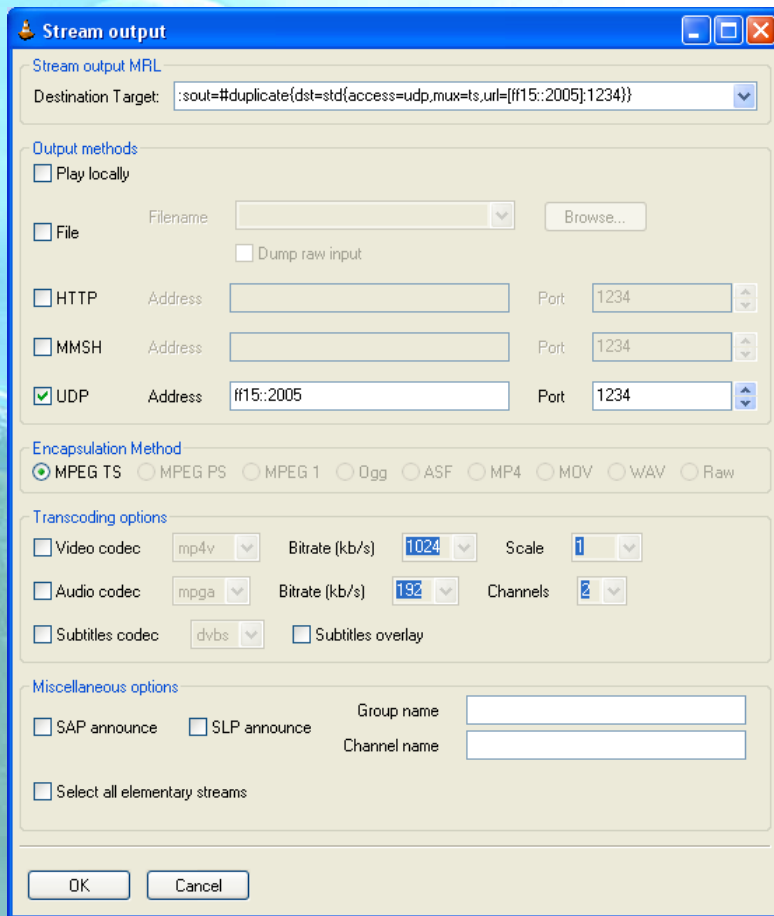
Server



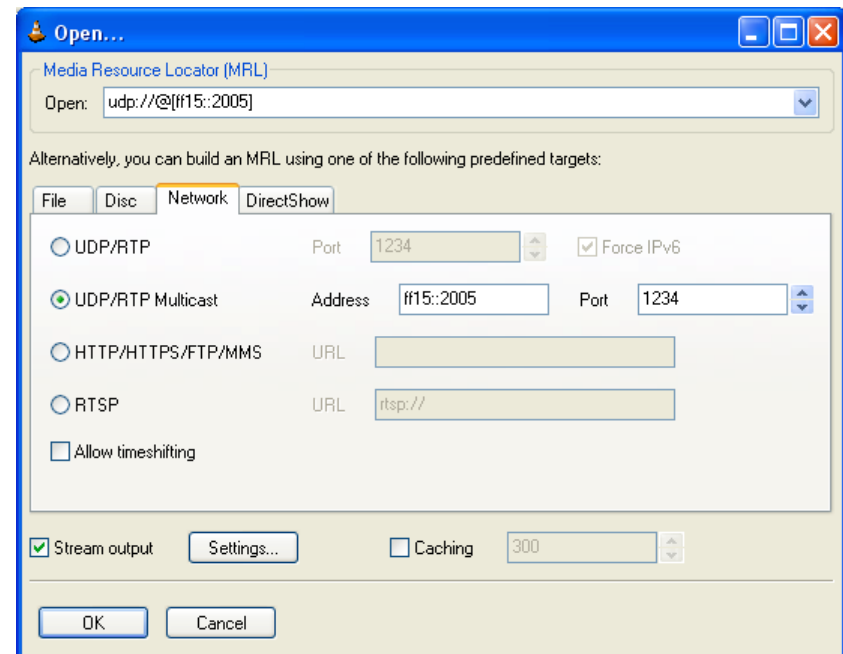
Client

IPv6 Applications: Exercise 4 (2)

- VLC with Multicast

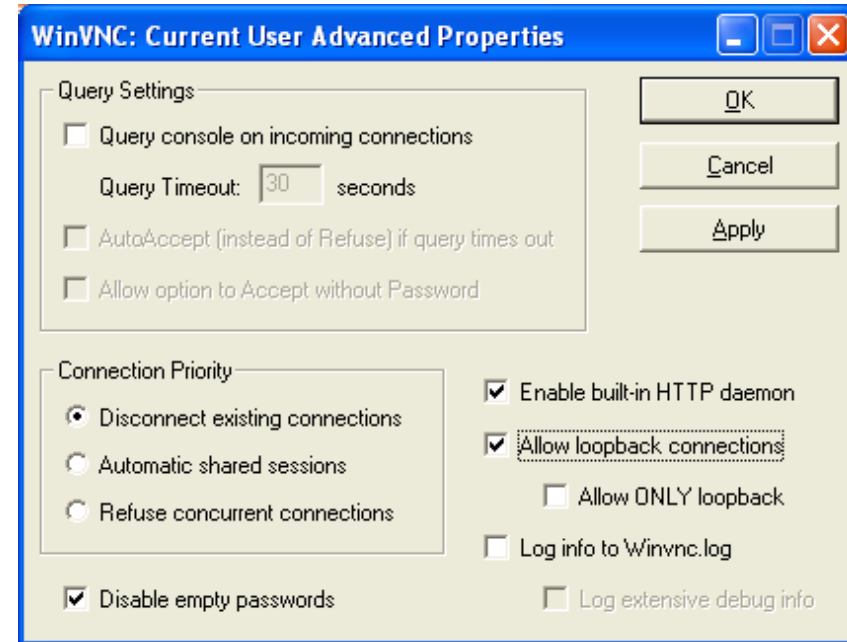
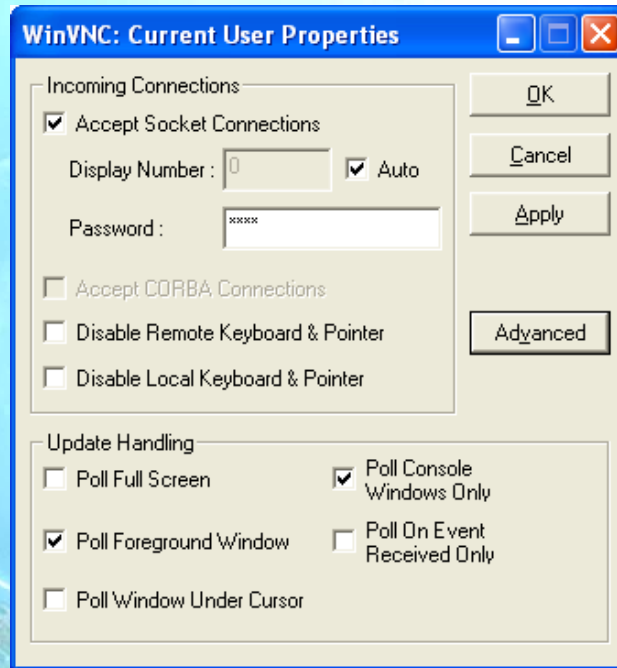


Server



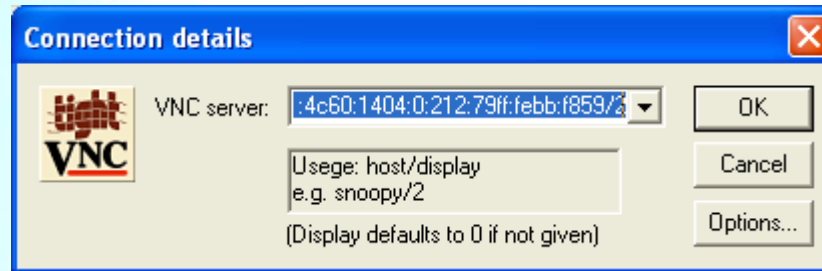
Client

IPv6 Applications: Exercise 5 (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”

IPv6 Applications: Exercise 5 (2)



- **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] draft-huitema-v6ops-teredo-05
- [TEREDOC]
<http://www.microsoft.com/technet/prodtechnol/winxpro/maintain/teredo.msp>
- [ISATAP] draft-ietf-ngtrans-isatap-24
- [AYIYA] draft-massar-v6ops-ayiya-02
- [SILKROAD] draft-liumin-v6ops-silkroad-02
- [DSTM] draft-ietf-ngtrans-dstm-10
- [SIIT] RFC2765
- [NATPT] RFC2767
- [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] RFC2462
- [STATEFULL] RFC3315
- [PRIVACY] RFC3041
- Windows IPv6
 - http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sag_ip_v6_add_utils.msp
 - <http://www.microsoft.com/technet/community/columns/cableguy/cg0902.msp>.