Transition Mechanisms

APRICOT 2006

IPv6 Workshop

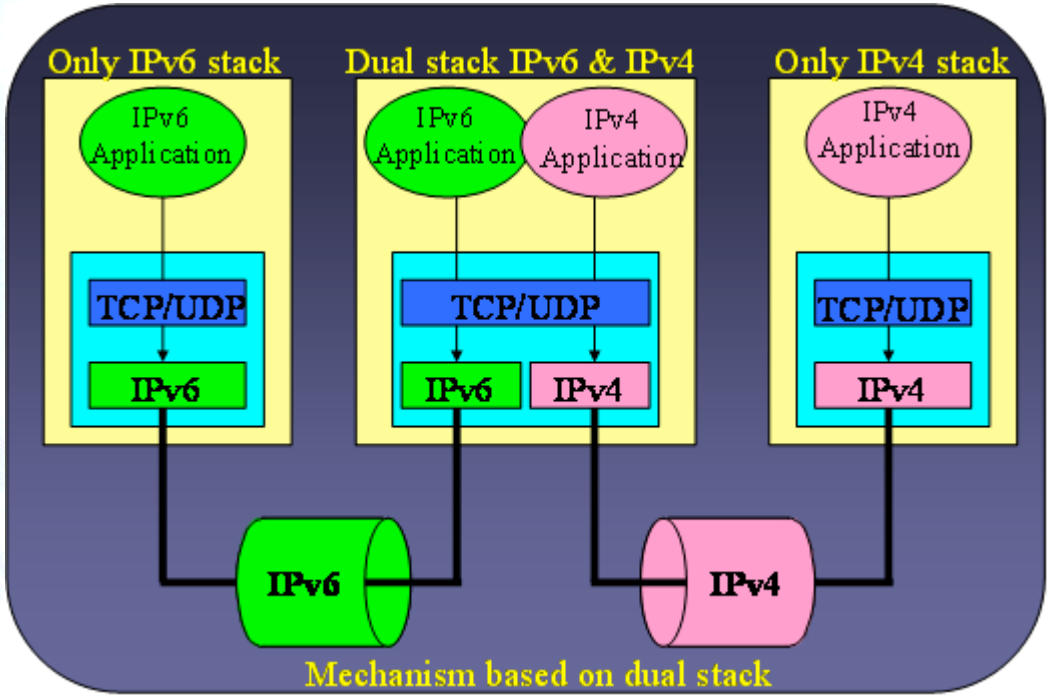
Jordi Palet (jordi.palet@consulintel.es)

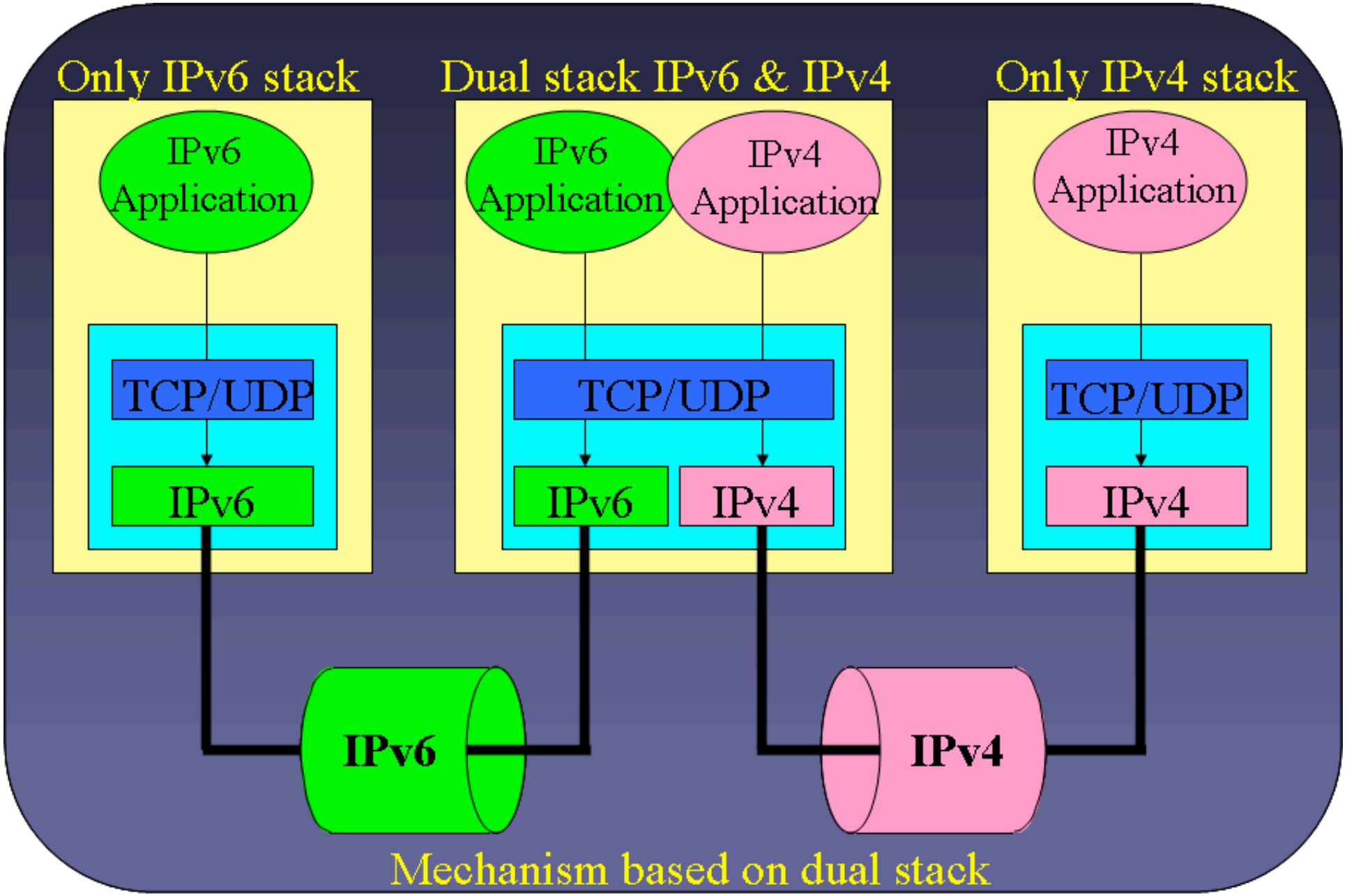
Transition Mechanisms

- 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

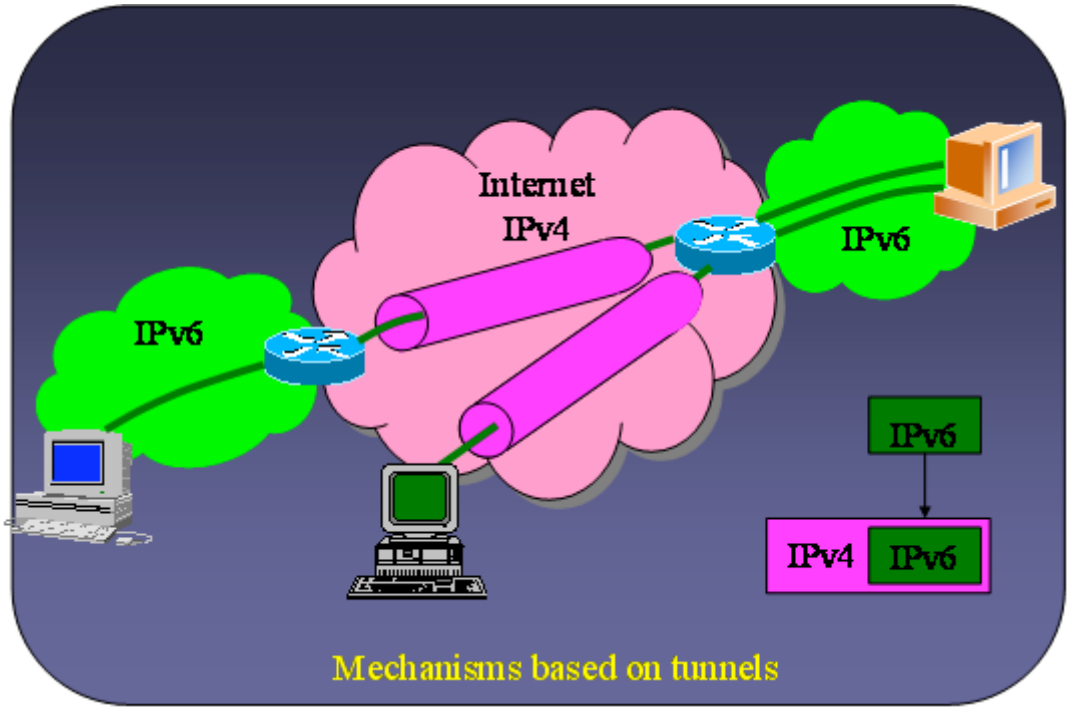
- All the hosts have both stacks IPv4 & IPv6
- IPv6-only communications ==> IPv6 stack, assuming IPv6 network support
- IPv4-only communications ==> IPv4 stack

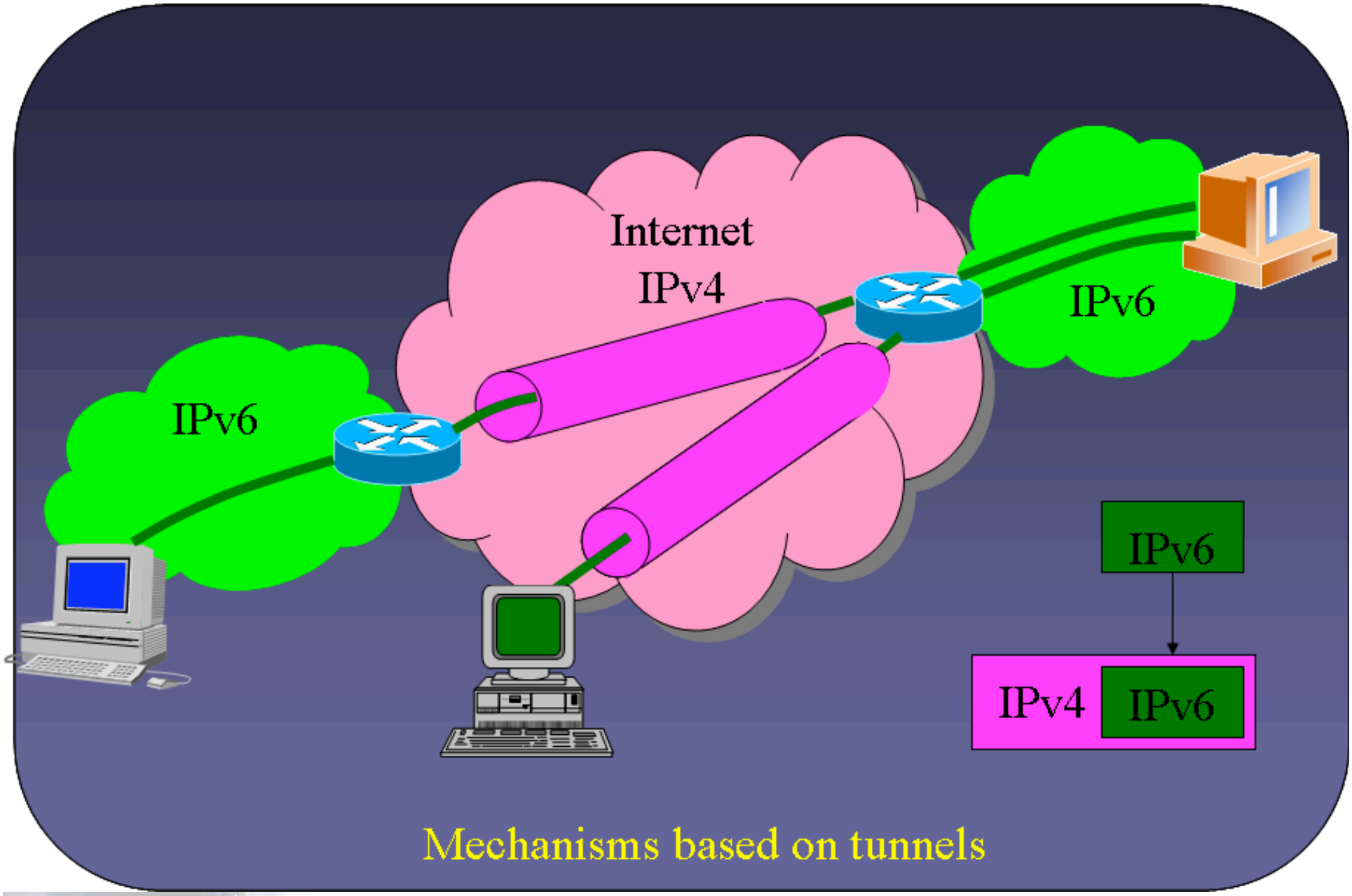



Only IPv6 stack
Dual stack IPv6 & IPv4
Only IPv4 stack
IPv6 Application
IPv6 Application
IPv4 Application
IPv4 Application
TCP/UDP
TCP/UDP
TCP/UDP
IPv6
IPv6
IPv4
IPv4
IPv6
IPv4
Mechanism based on dual stack

Tunnels: IPv6 in IPv4 (1)

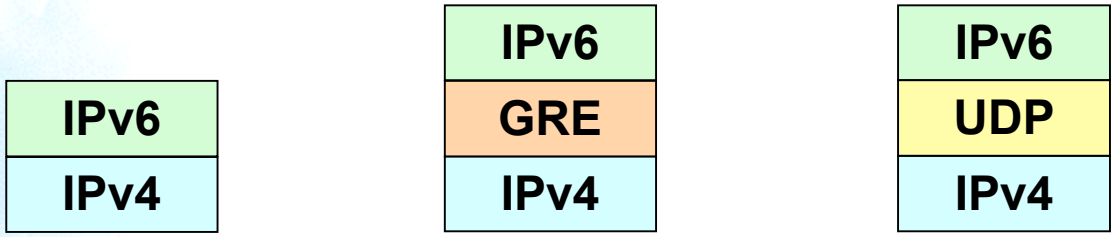
- It is used to provide IPv6 connectivity in IPv4-only networks
- The IPv6 packets are encapsulated into IPv4 packets
- There are different ways to make the encapsulation
  - 6in4, 6to4, 6over4, UDP, etc.
- The resulting packets flow through IPv4 networks towards the tunnel end point (TEP)




Internet
IPv4
IPv6
IPv6
IPv6
IPv4
IPv6
Mechanisms based on tunnels

Tunnels IPv6 in IPv4 (2)

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



- Same for IPv4 being used in IPv6-only networks

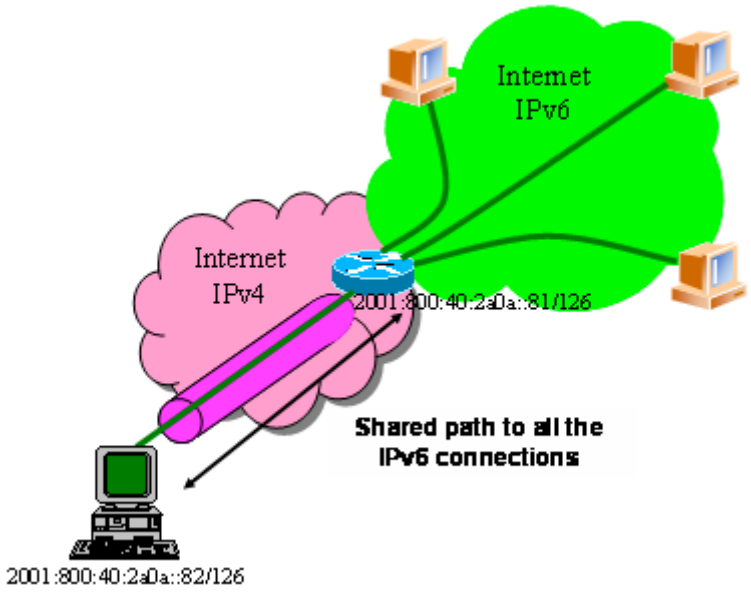
Tunnels IPv6 in IPv4 (3)

- 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]

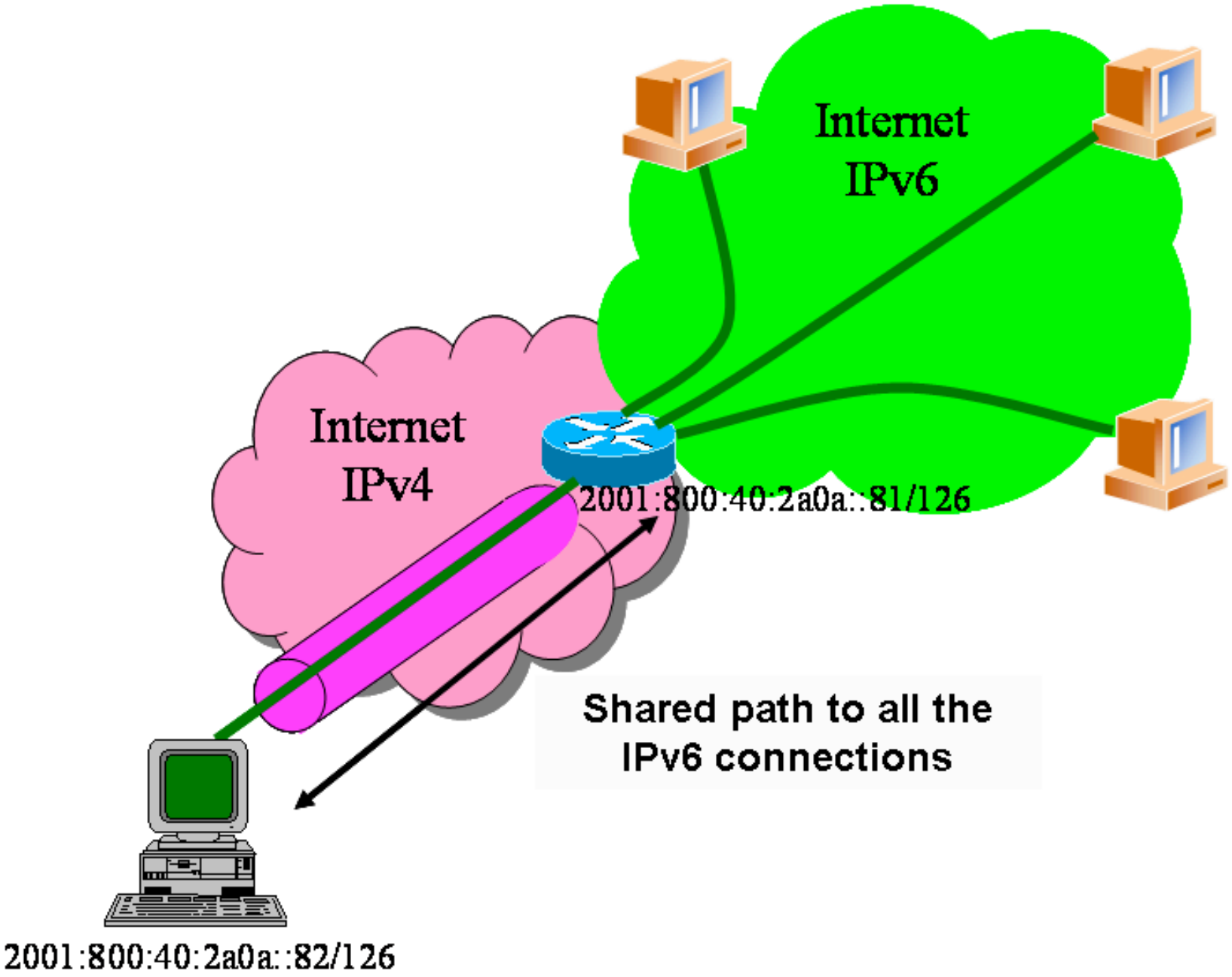
- (*) Commoner mechanisms and explained in depth in the following slides

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



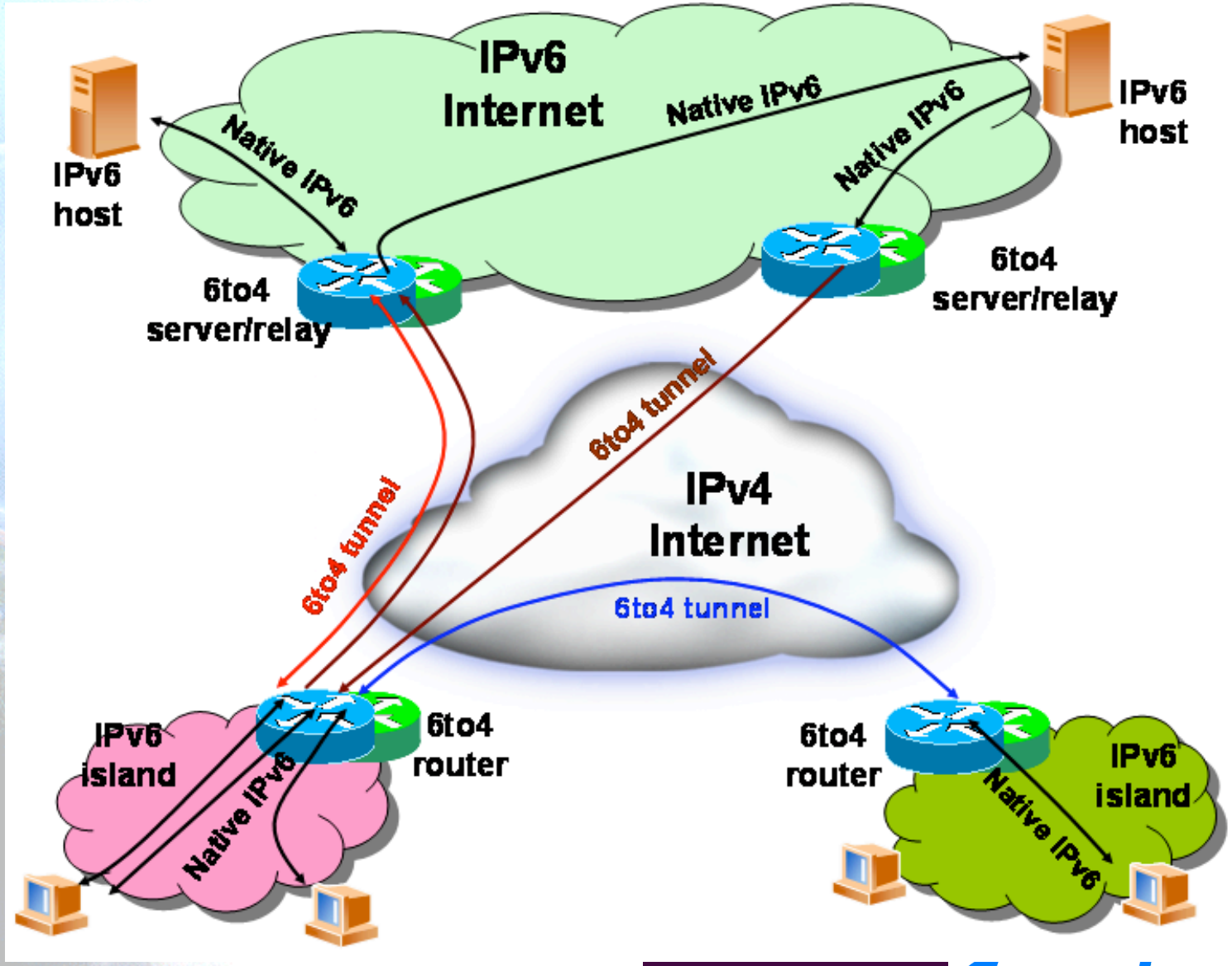
- From the point of view of IPv6 the tunnel is considered as a point-to-point link
  - Only an IPv6 network-hop although several IPv4-hops exist in the path
- The IPv6 addresses of both tunnel-ends belong to the same prefix
- All the IPv6 connections of the end-host flow always through the router located 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


Internet
IPv6
Internet
IPv4
2001:800:40:2a0a::81/126
Shared path to all the
IPv6 connections
2001:800:40:2a0a::82/126

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

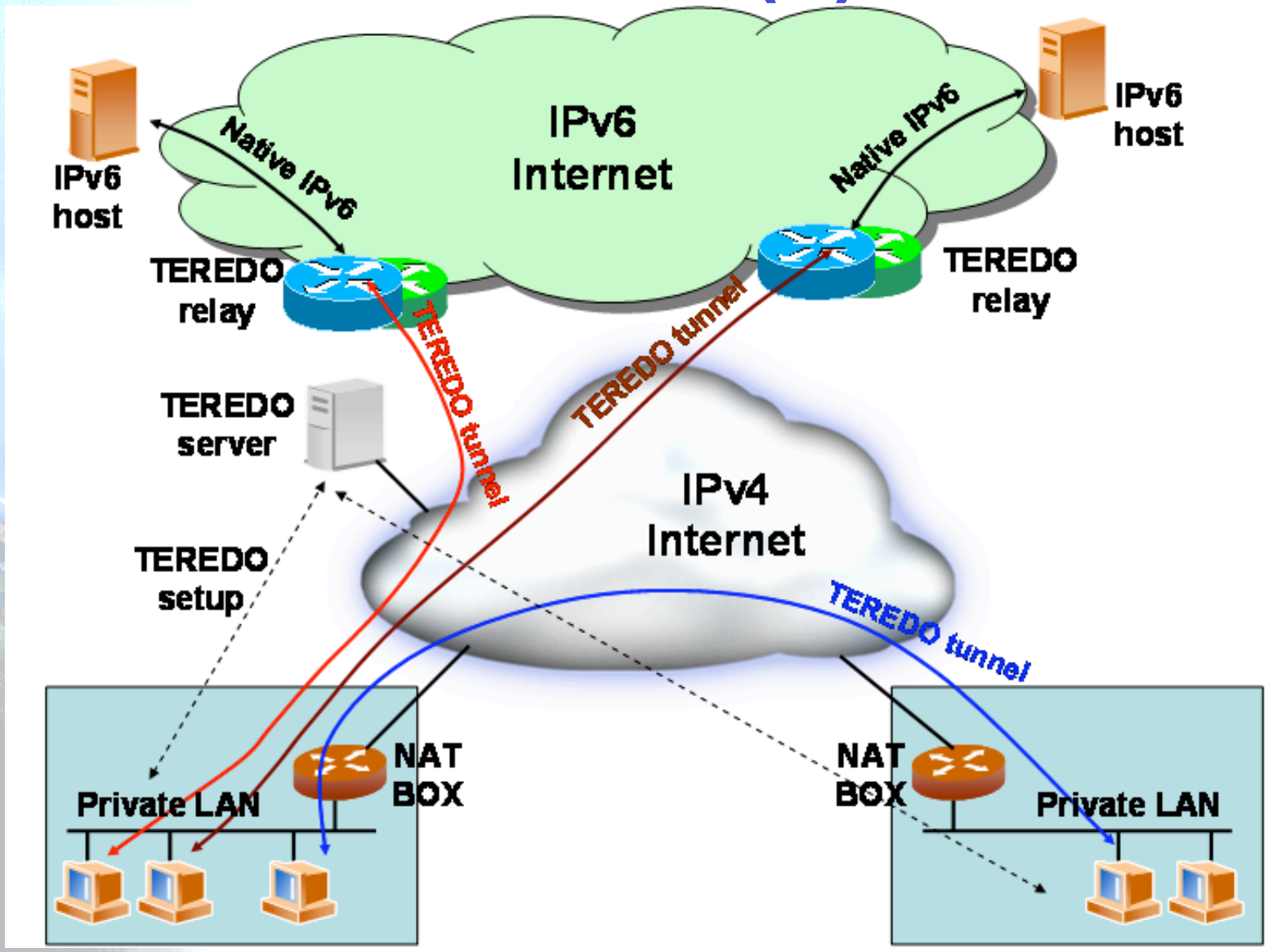
6to4 Tunnels (1)



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)

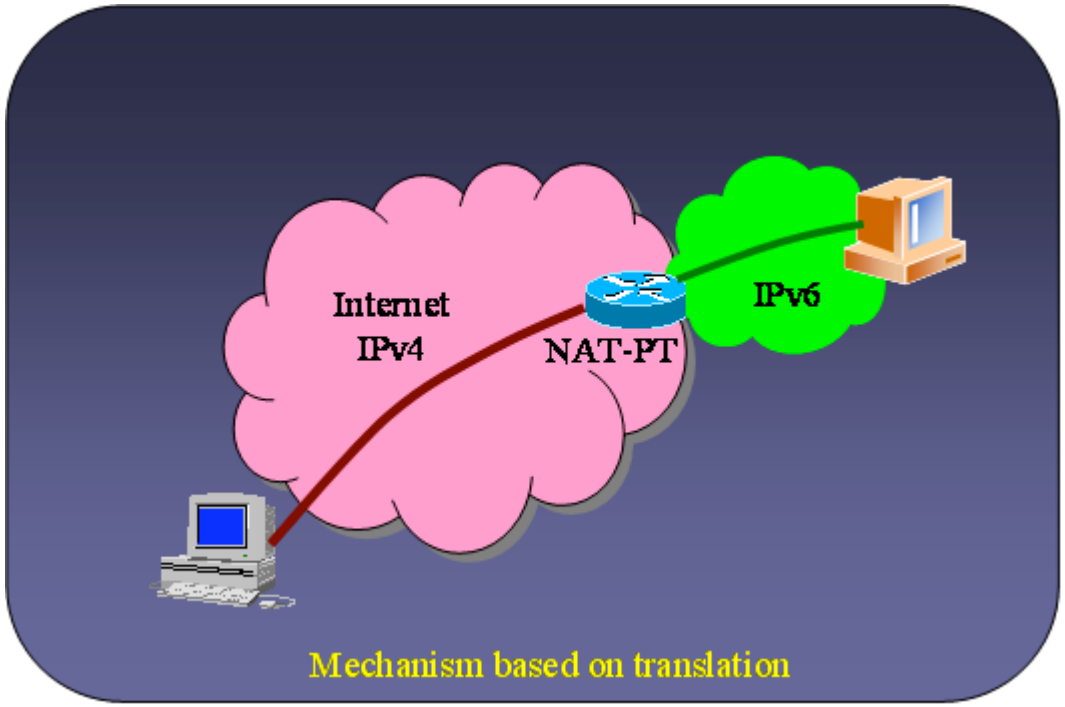

IPv6 host
Native IPv6
IPv6 Internet
Native IPv6
IPv6 host
TEREDO relay
TEREDO relay
TEREDO tunnel
TEREDO tunnel
TEREDO server
IPv4 Internet
TEREDO setup
TEREDO tunnel
NAT BOX
Private LAN
NAT BOX
Private LAN

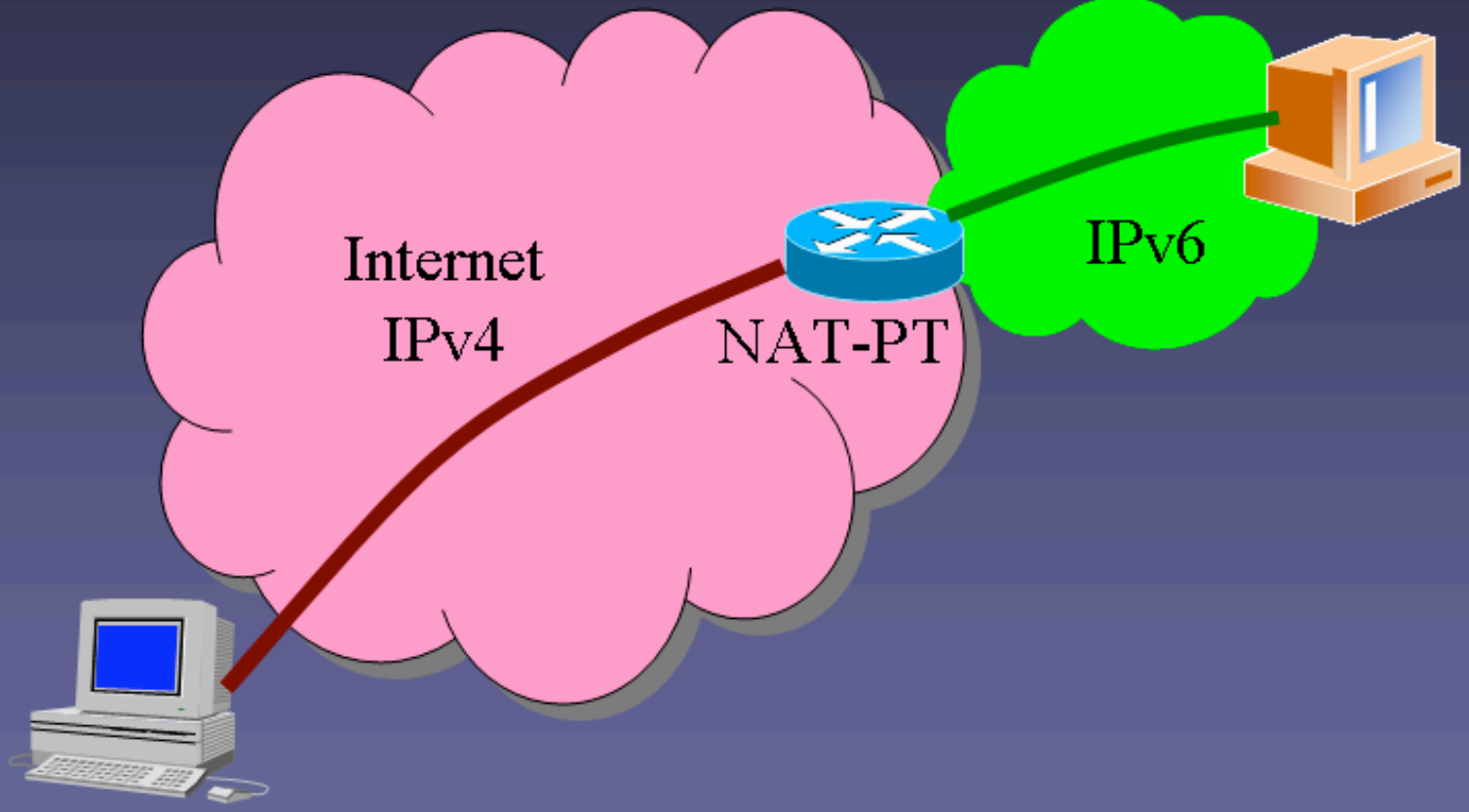
Teredo (2)

- 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 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

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.

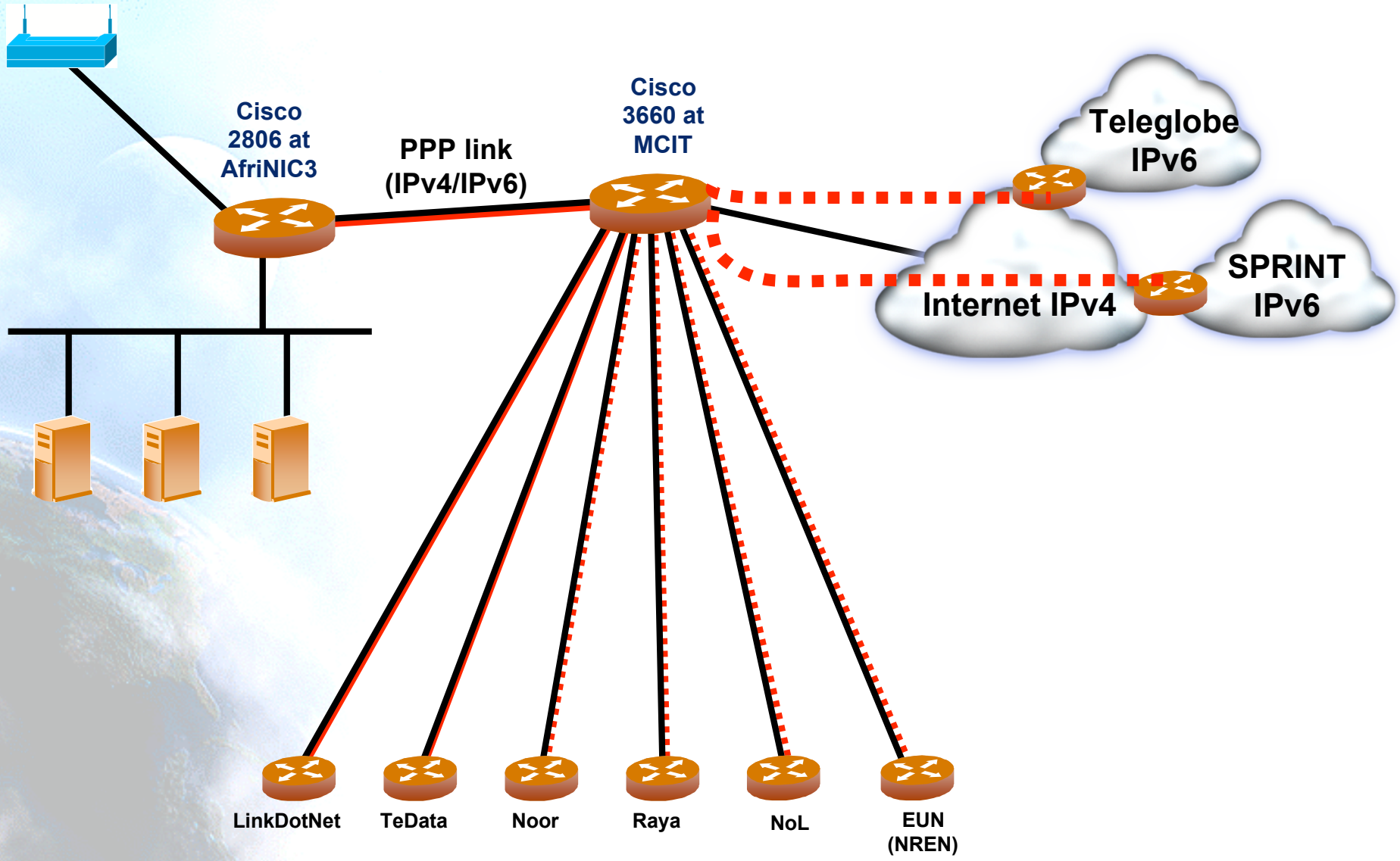




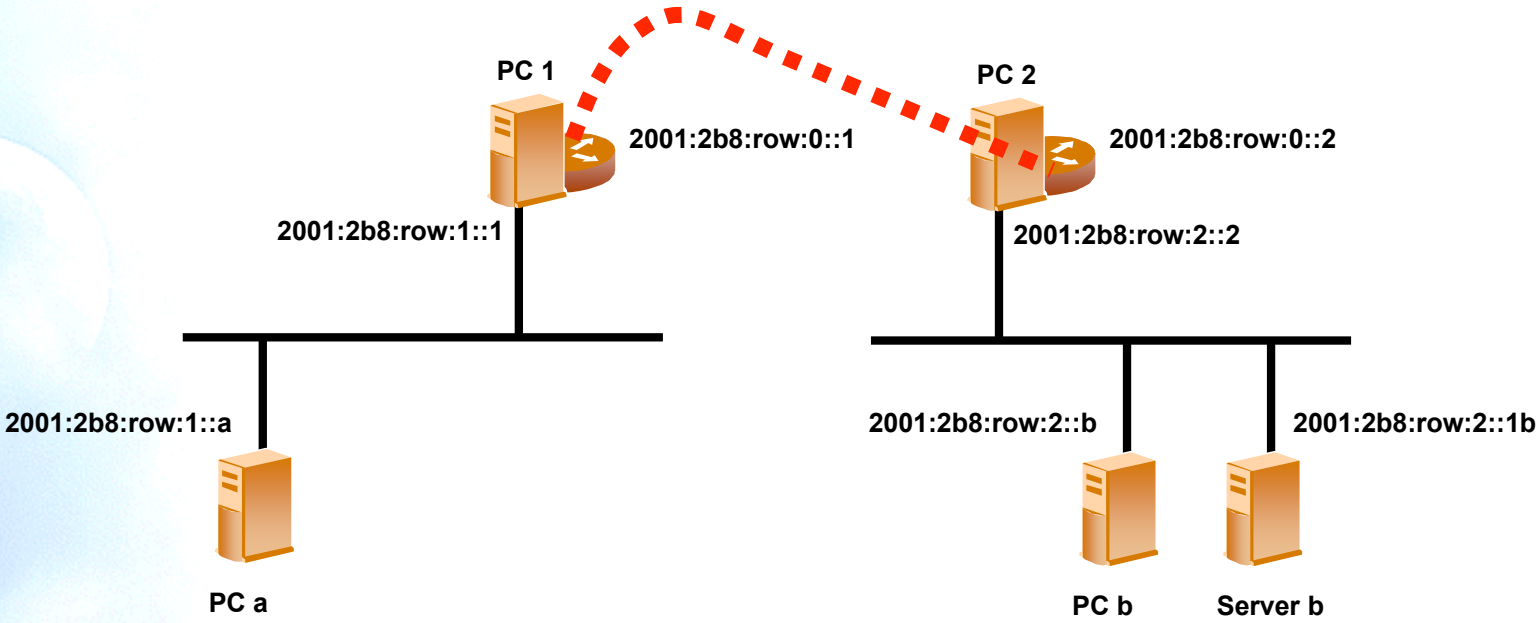
Mechanism based on translation

Building IPv6 Networks

Our Network at AfriNIC3



Our Local Lab at AfriNIC3



Backbone Networks

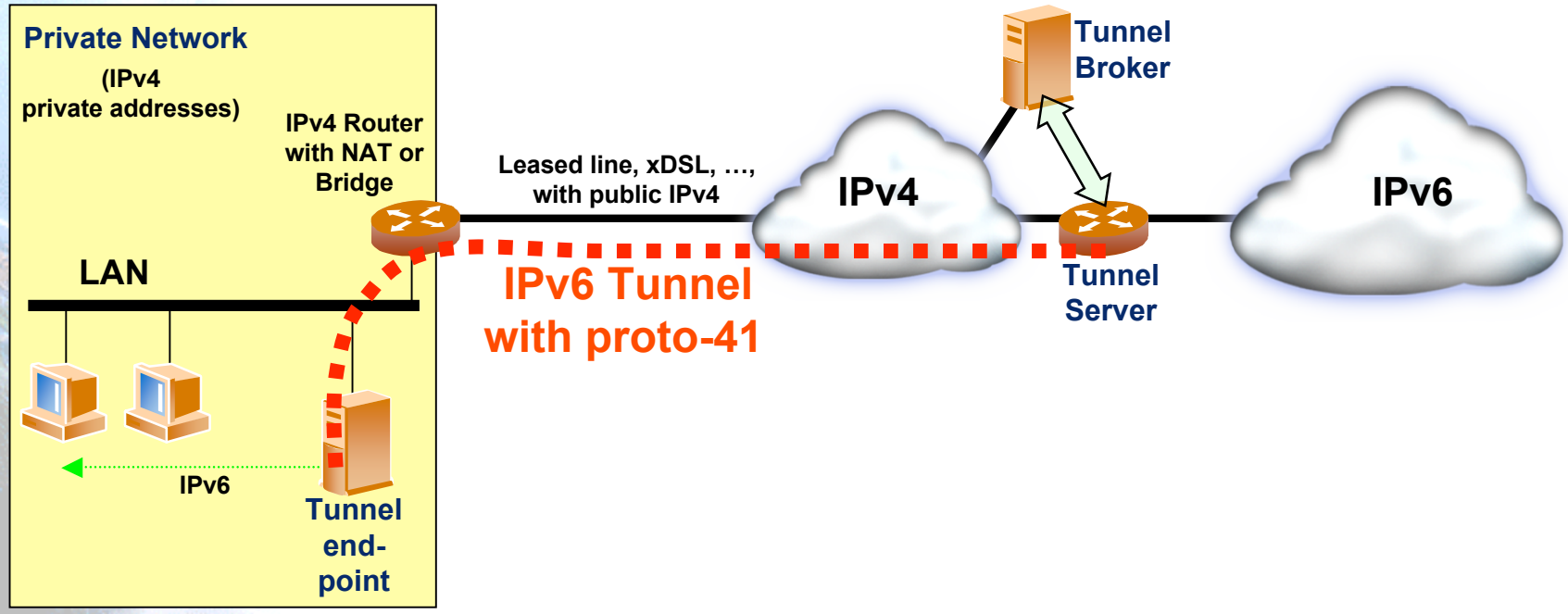
- Typically with a high degree of O&M:
  - Today IPv6 comes as a value added with no additional cost when doing other upgrades
- The deployment of MPLS and Ethernet makes this transition even easier
  - Today all the big “intercontinental” networks offer IPv6 support
- In some cases it may be required to use tunnels, may be temporarily in some PoPs
  - Waiting for some equipments to be upgraded or because special “features”
- IPv6 can be deployed in a record time

Access Networks

- Trouble may be generated by two elements:
  - PoP equipment (technology-dependant)
  - CPEs (cheaper ones often don't offer native IPv6 support)
- However this is also changing quickly
- Despite that, the IPv6 benefits are already visible when using transition mechanisms
- References:
  - http://www.ipv6-es.com (tutorial, deployment in broadband networks)
  - draft-ietf-v6ops-bb-deployment-scenarios

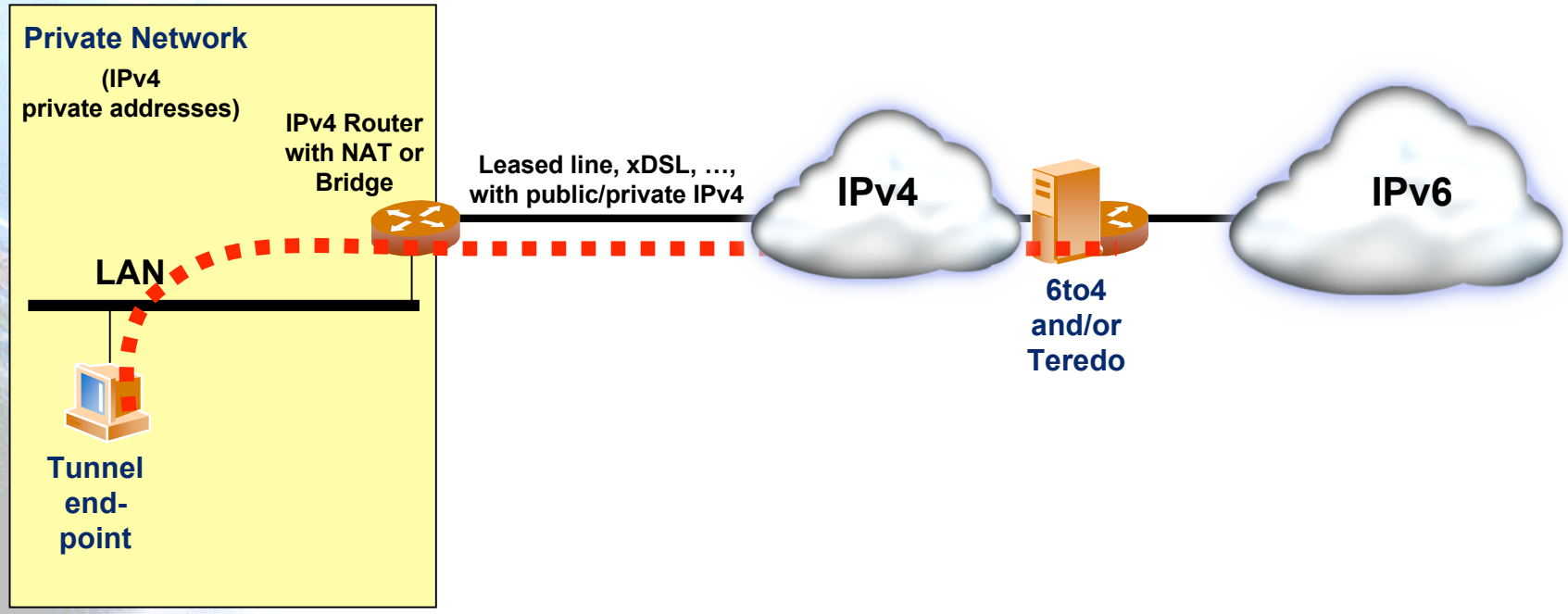
Example of IPv6 Tunnel

- All the devices can use IPv6 with a single tunnel



Example of 6to4 + Teredo

- Each device setup its own tunnel

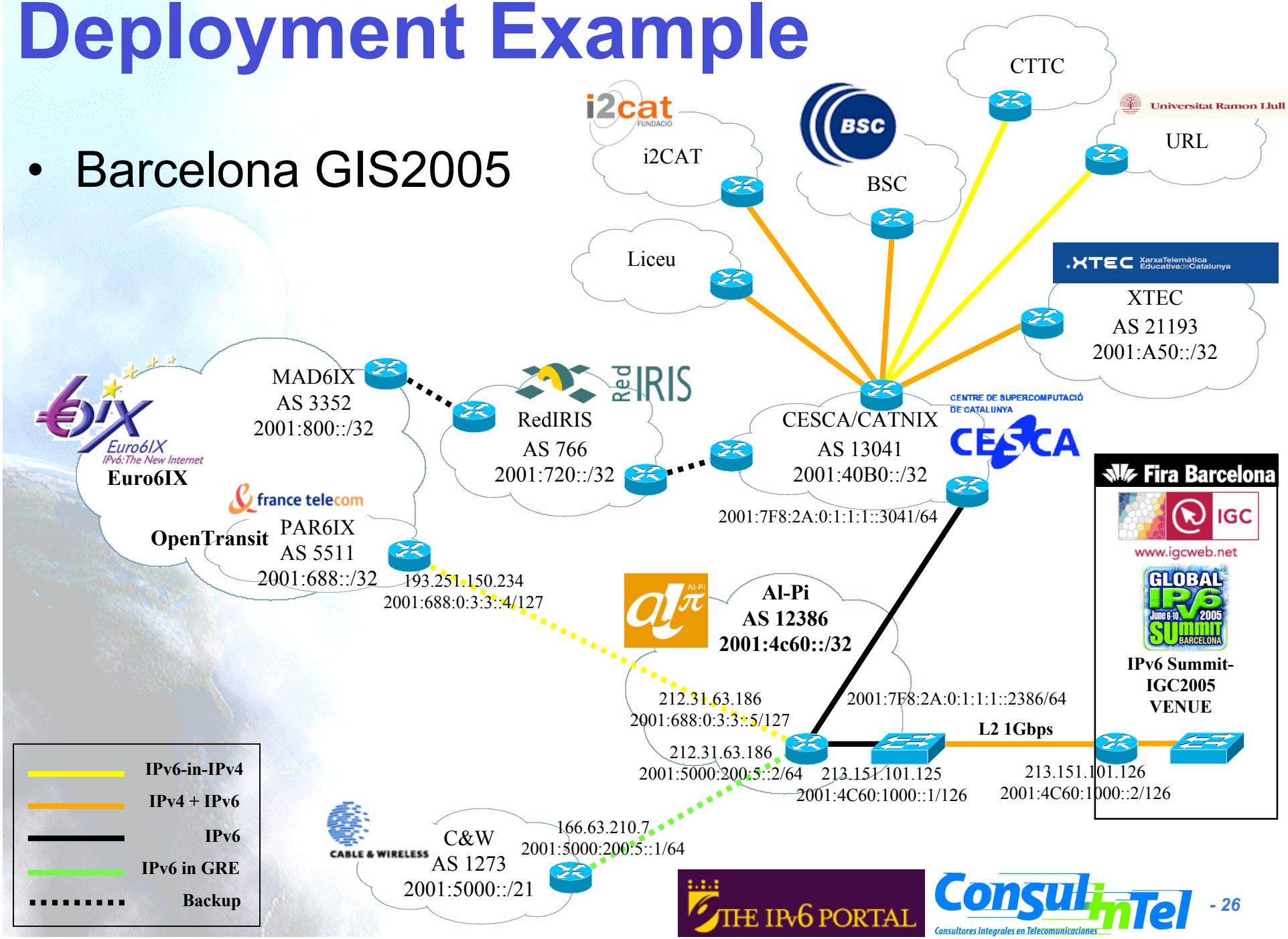


Advantages

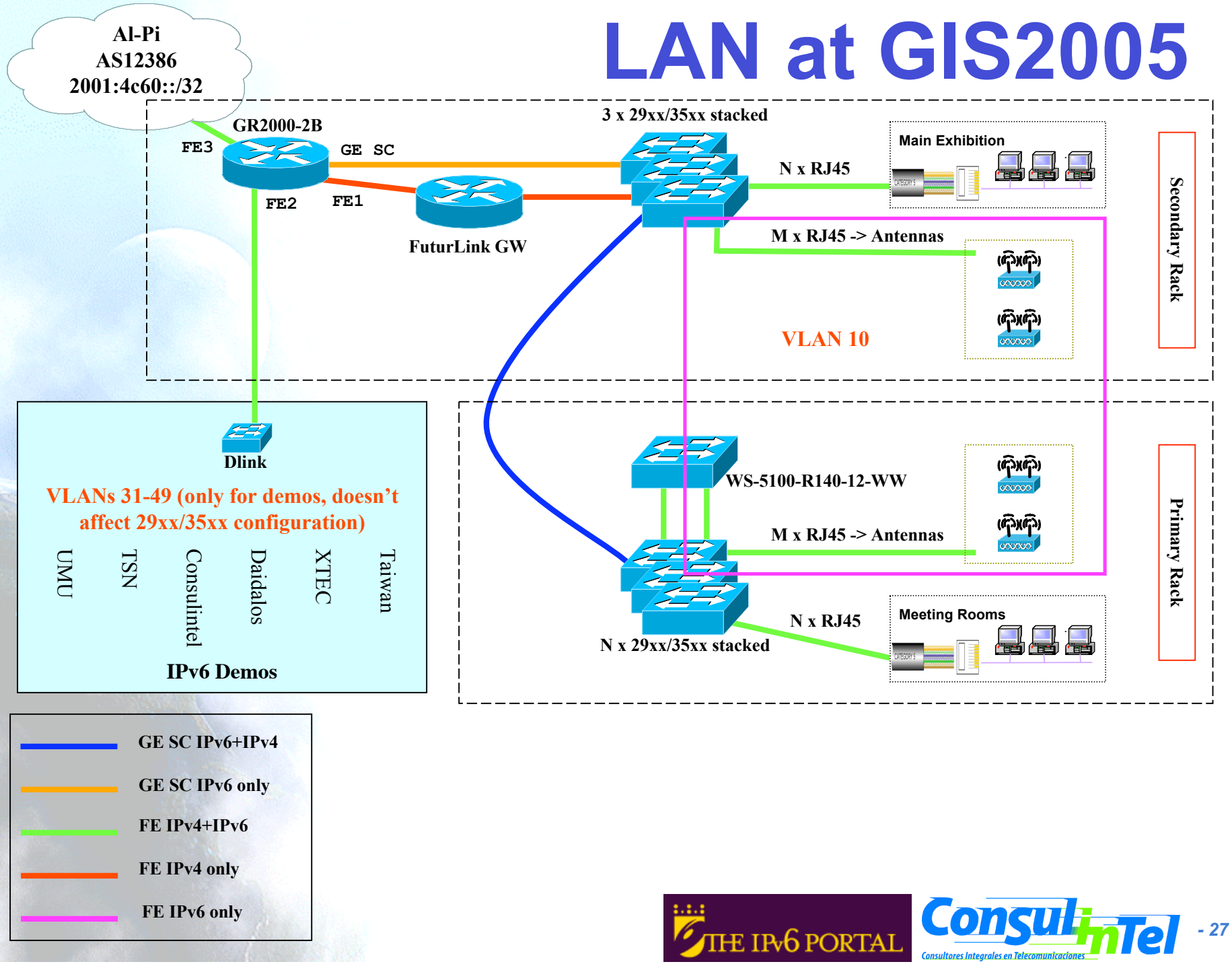
- Restoring the innovation
- Less complexity and troubles because NAT
- O&M Cost reduction
  - Especially if it becomes IPv6-only
  - Confirmed by big operators
- Creation of new applications and services
- New business generation

Deployment Example

- Barcelona GIS2005



LAN at GIS2005



Traffic Stats

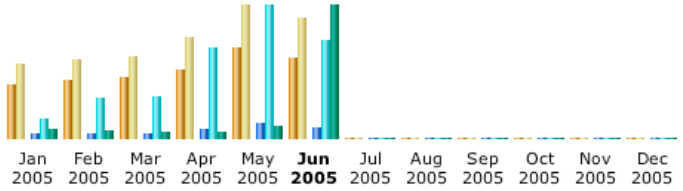
Last Update: 30 Jun 2005 - 00:30
Reported period: Jun 2005 OK

Summary — IPv4 + IPv6

Reported period	Month Jun 2005				
First visit	01 Jun 2005 - 00:12				
Last visit	30 Jun 2005 - 00:10				
	Unique visitors	Number of visits	Pages	Hits	Bandwidth
Viewed traffic *	**3161**	**4735** (1.49 visits/visitor)	**15423** (3.25 pages/visit)	**129591** (27.36 hits/visit)	**49.23 GB** (10901.66 KB/visit)
Not viewed traffic *			**13670**	**14723**	**19.27 GB**

* Not viewed traffic includes traffic generated by robots, worms, or replies with special HTTP status codes.

Monthly history



Month	Unique visitors	Number of visits	Pages	Hits	Bandwidth
Jan 2005	2139	2962	5913	26100	3.54 GB
Feb 2005	2297	3099	7120	53904	3.09 GB
Mar 2005	2387	3204	7392	55884	2.30 GB
Apr 2005	2719	3974	13646	120189	2.70 GB
May 2005	3576	5234	19739	176331	4.57 GB
Jun 2005	3161	4735	15423	129591	49.23 GB
Jul 2005	0	0	0	0	0
Aug 2005	0	0	0	0	0
Sep 2005	0	0	0	0	0
Oct 2005	0	0	0	0	0
Nov 2005	0	0	0	0	0
Dec 2005	0	0	0	0	0
Total	16279	23208	69233	561999	65.43 GB

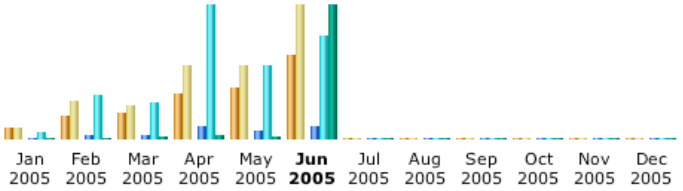
Last Update: 30 Jun 2005 - 00:30
Reported period: Jun 2005 OK

Summary — IPv6

Reported period	Month Jun 2005				
First visit	01 Jun 2005 - 01:42				
Last visit	30 Jun 2005 - 00:10				
	Unique visitors	Number of visits	Pages	Hits	Bandwidth
Viewed traffic *	**211**	**337** (1.59 visits/visitor)	**1323** (3.92 pages/visit)	**10830** (32.13 hits/visit)	**7.37 GB** (22939.12 KB/visit)
Not viewed traffic *			**237**	**304**	**1.45 GB**

* Not viewed traffic includes traffic generated by robots, worms, or replies with special HTTP status codes.

Monthly history



Month	Unique visitors	Number of visits	Pages	Hits	Bandwidth
Jan 2005	27	29	69	725	12.13 MB
Feb 2005	57	97	411	4604	36.30 MB
Mar 2005	67	86	373	3806	128.30 MB
Apr 2005	113	186	1252	13967	200.64 MB
May 2005	131	185	822	7728	147.58 MB
Jun 2005	211	337	1323	10830	7.37 GB
Jul 2005	0	0	0	0	0
Aug 2005	0	0	0	0	0
Sep 2005	0	0	0	0	0
Oct 2005	0	0	0	0	0
Nov 2005	0	0	0	0	0
Dec 2005	0	0	0	0	0
Total	606	920	4250	41660	7.89 GB

Transition Mechanisms Configuration

Cisco Config of Manual Tunnels

- Router> enable
- Router# configure terminal
- Router(config)# interface tunnel tunnel-number
  - Example: Router(config)# interface tunnel 0
- Router(config-if)# ipv6 address ipv6-prefix/ prefix-length [eui-64] (assign an IPv6 network to the interface and enable IPv6 on it)
  - Example: Router(config-if)# ipv6 address 2001:DB8:1:1::1/126
- Router(config-if)# tunnel source { ip-address | type number} (Tunnel source can be an IPv4 address or an interface)
  - Example: Router(config-if)# tunnel source ethernet 0
- Router(config-if)# tunnel destination ip-address (Tunnel destination can be an IPv4 address or an interface)
  - Example: Router(config-if)# tunnel destination 192.168.30.1
- Router(config-if)# tunnel mode {aurp | cayman | dvmrp | eon | gre | gre multipoint | gre ipv6 | ipip [decapsulate-any] | iptalk | ipv6ip |mpls | nos}
  - Example1: Router(config-if)# tunnel mode ipv6ip (6in4 tunnel)
  - Example2: Router(config-if)# tunnel [mode gre ip] (GRE tunnel)
- Router(config-if)# exit
- Router(config)# ipv6 route ipv6-prefix/ prefix-length tunnel tunnel-number (configures a static route)
  - Example: Router(config)# ipv6 route ::/0 tunnel 0

Cisco Config of 6to4

- Router> enable
- Router# configure terminal
- Router(config)# interface tunnel tunnel-number
  - Example: Router(config)# interface tunnel 0
- Router(config-if)# ipv6 address ipv6-prefix/ prefix-length [eui-64] (assigns an IPv6 address to the interface and enables IPv6 on it)
  - Eample: Router(config-if)# ipv6 address 2002:c0a8:6301:1::1/64
- Router(config-if)# tunnel source { ip-address | type number} (Tunnel source can be an IPv4 address or an interface)
  - Example: Router(config-if)# tunnel source ethernet 0
- Router(config-if)# tunnel mode ipv6ip 6to4 (6to4 tunnel)
- Router(config-if)# exit
- Router(config)# ipv6 route ipv6-prefix/ prefix-length tunnel tunnel-number (configures an static route for the tunnel interface)
  - Example: Router(config)# ipv6 route 2002::/16 tunnel 0

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
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

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.simphalempin.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 guarantied 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

Examples of Applications

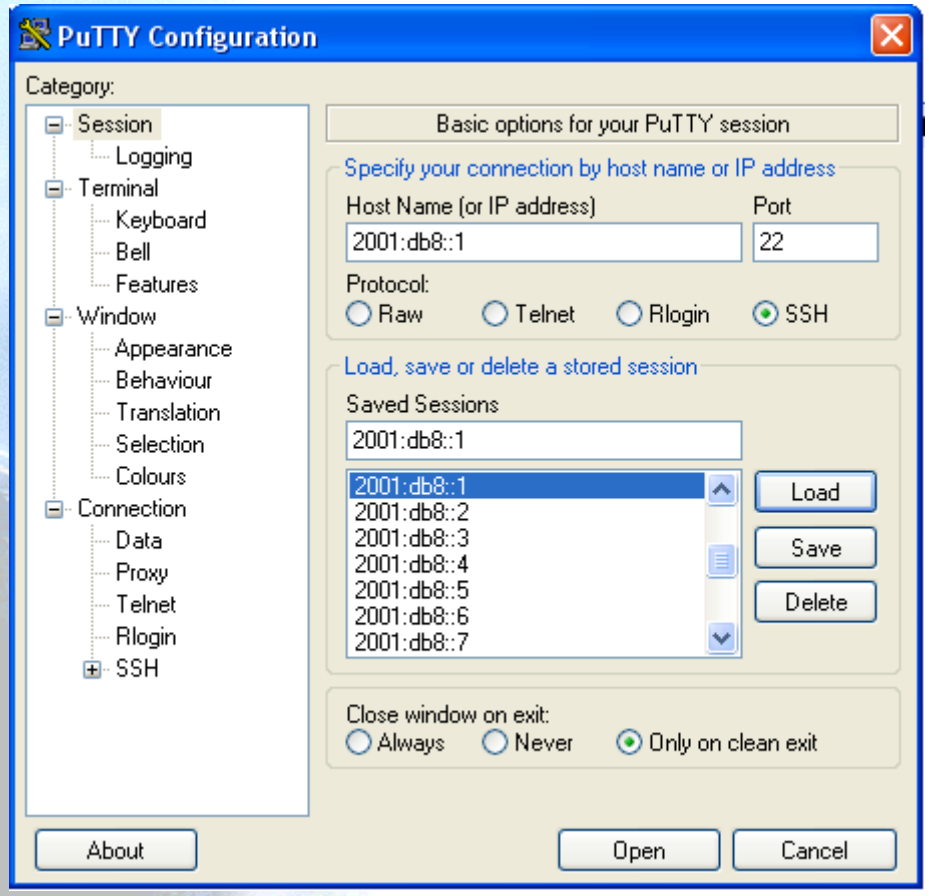
IPv6 Applications (1)

- 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

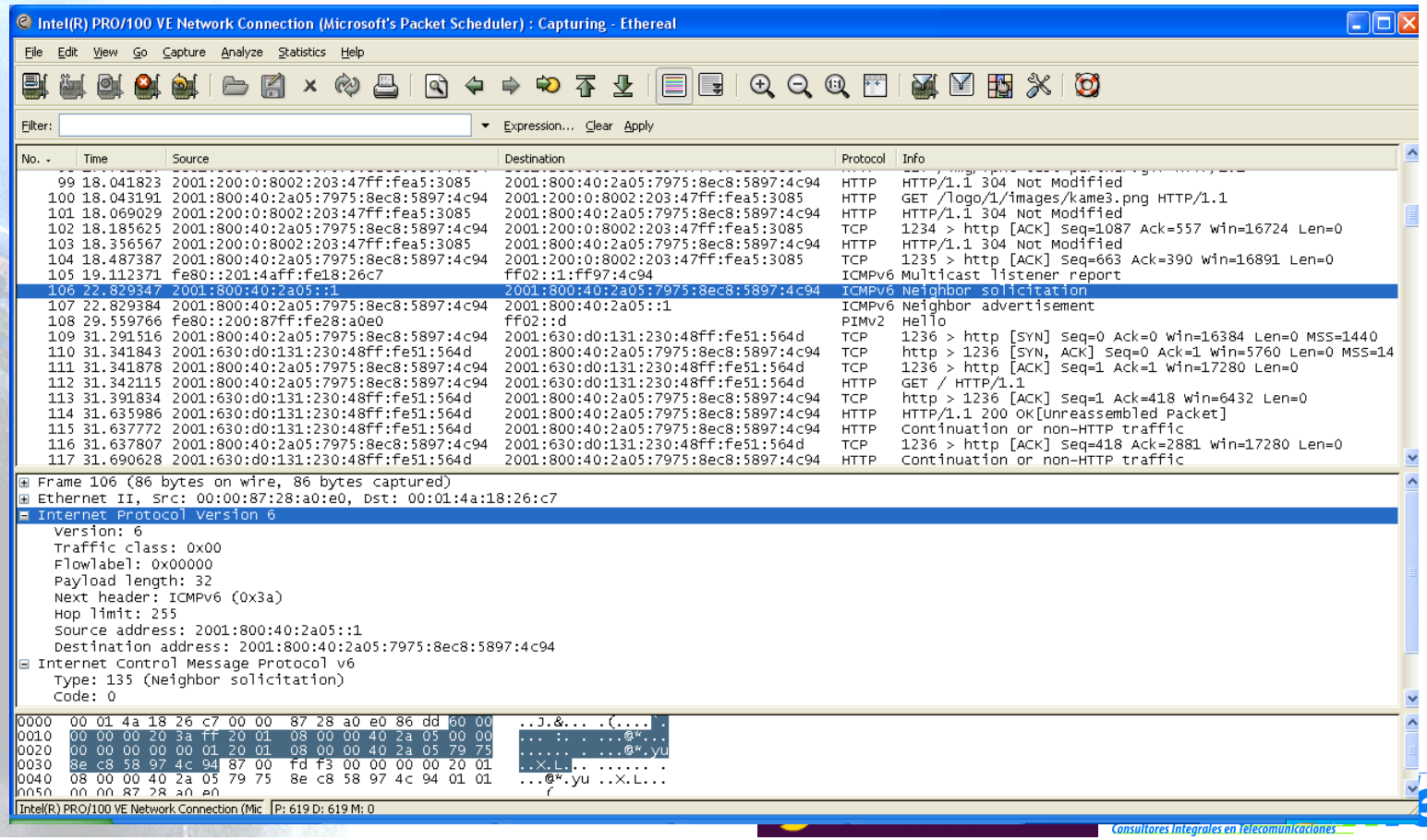
IPv6 Applications (3)



- **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

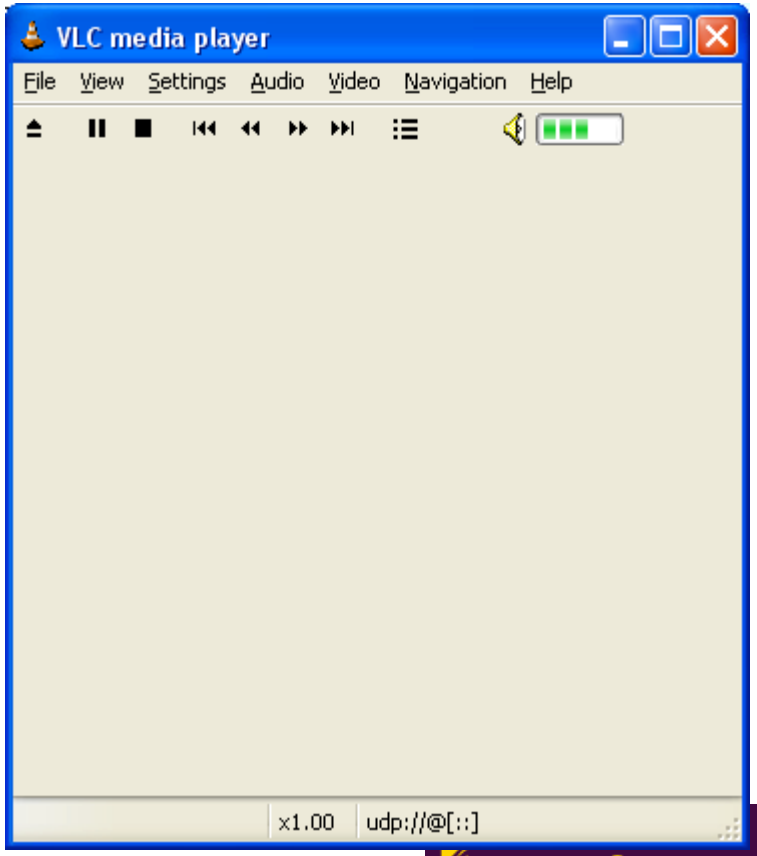
IPv6 Applications (4)

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



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

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

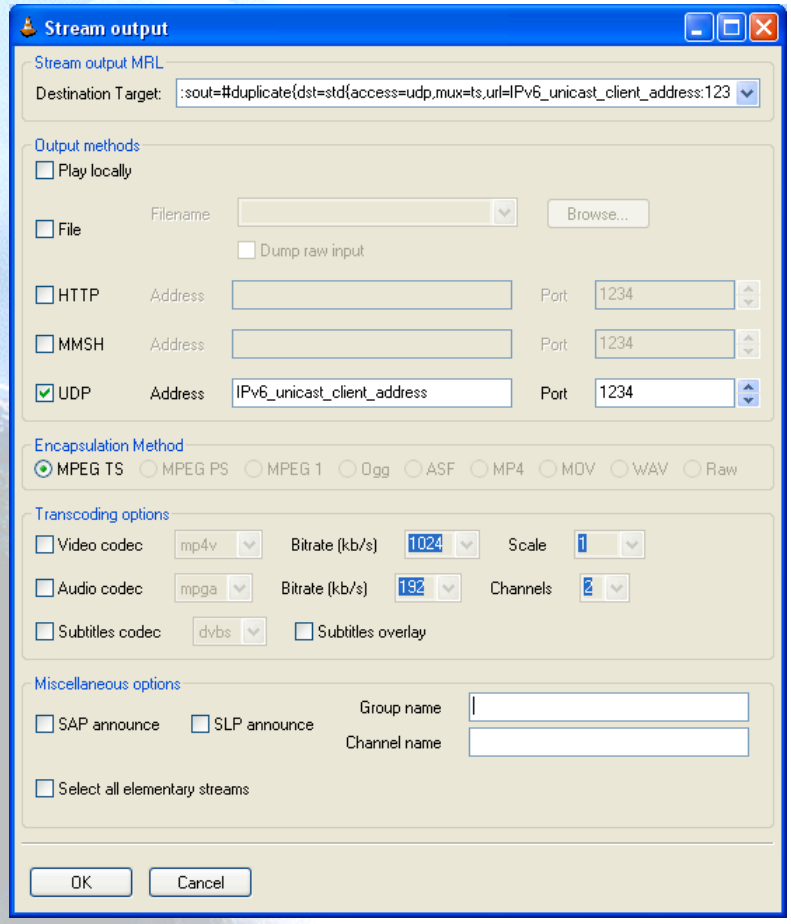
- To install (in case those are not already installed):
  - SSH Client with IPv6 support (Putty)
  - FTP Client (Command line on BSD, Linux, Windows)
  - Web Browser (Firefox, IE)
  - Ethereal
  - VLC
  - VNC

IPv6 Applications: Exercise 3

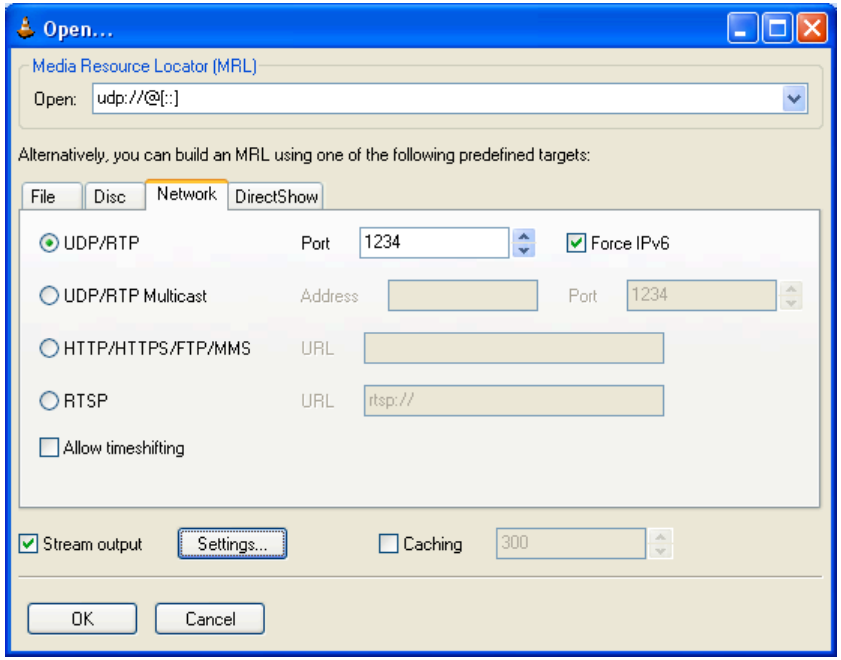
- To use the different services while Ethereal (or tcpdump) is used, in order to capture packets
- To use the SSH client to access by v4 or v6 choosing by means of DNS resolution
- To use the SSH client to access by v4 or v6 choosing by means of an application parameter (linux: #ssh -6|-4)(XP: ping -6|-4)

IPv6 Applications: Exercise 4 (1)

- VLC with Unicast



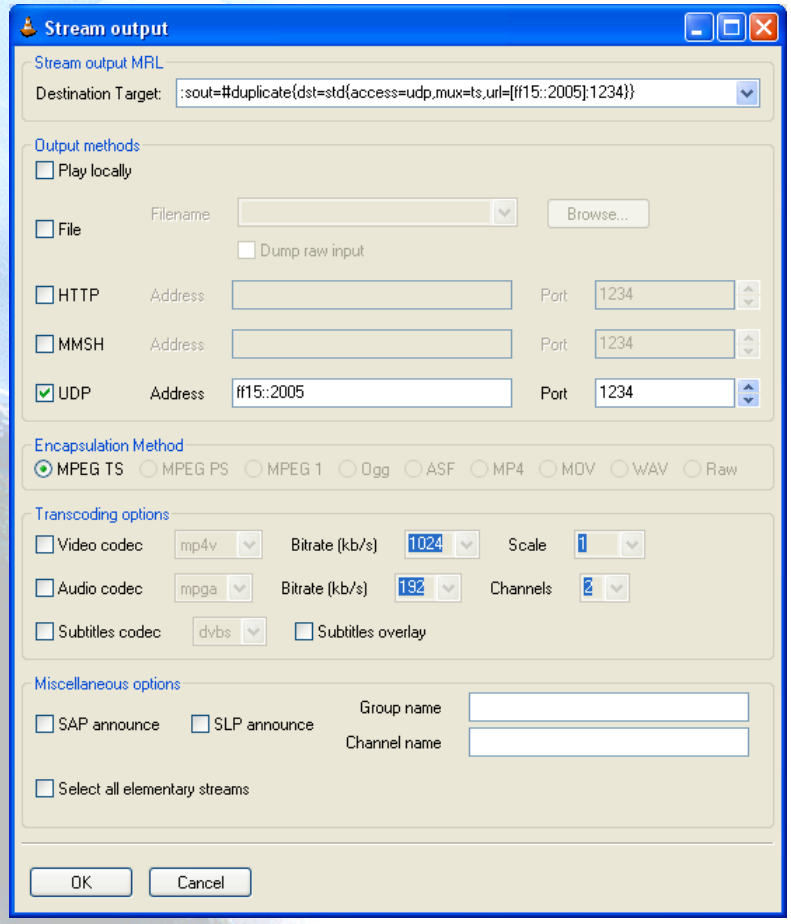
Server



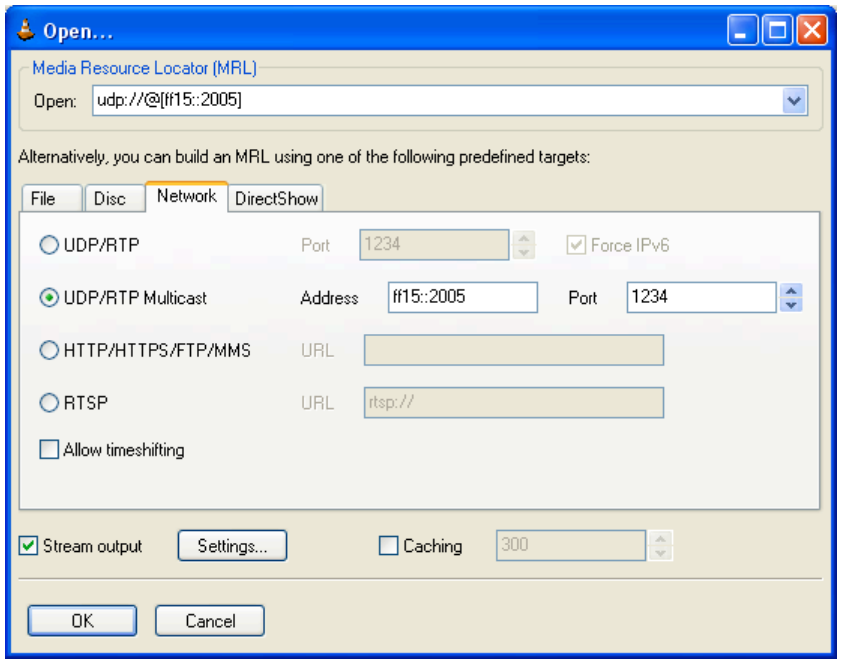
Client

IPv6 Applications: Exercise 4 (2)

- VLC with Multicast

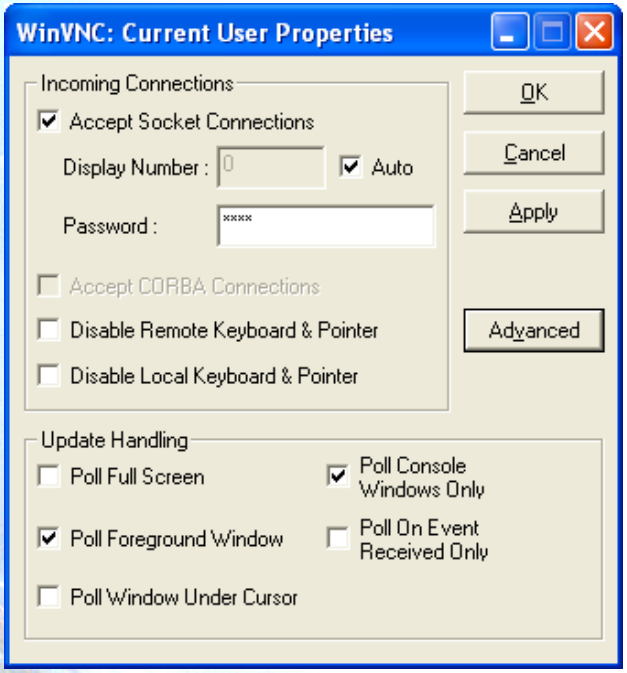


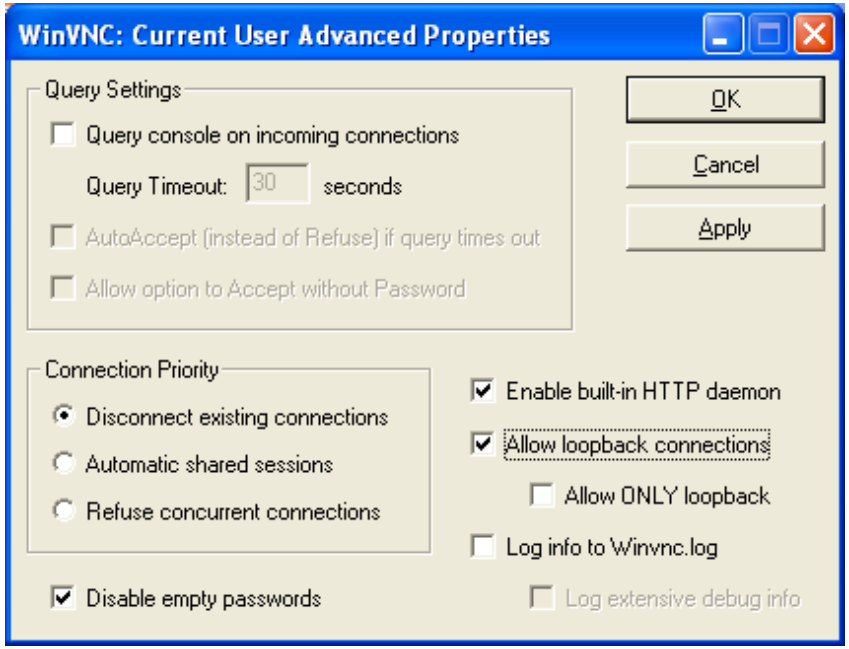
Server



Client

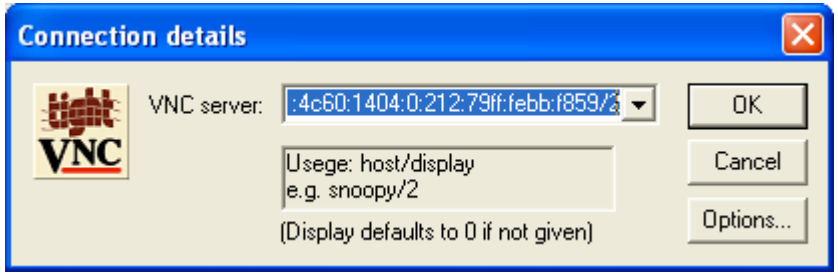
IPv6 Applications: Exercise 5 (1)





- **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 ‘/’

Thanks !

Contact:

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

The IPv6 Portal:

- http://www.ipv6tf.org

Madrid 2005 IPv6 Summit, info available at:

- http://www.ipv6-es.com