IPv6 Applications

<Location>, <Country> - <Month> <Year>

Copy ...Rights

This slide set is the ownership of the 6DEPLOY project (and previous projects 6NET, EURO6IX and 6DISS) via its partners

The Powerpoint version of this material may be reused and modified only with written authorization

Using part of this material must mention 6DEPLOY courtesy

PDF files are available from www.6deploy.eu

Looking for a contact ?

- ***Mail to : martin.potts@martel-consulting.ch***
- ***Or bernard.tuy@renater.fr***

Contributors

- Jim Bound, HP
- Brian Carpenter, IBM, Switzerland
- Tim Chown, UoS, UK
- Johann Fiedler, FhG, Germany
- Ian Foster, Argonne National Labs
- Tony Hain, Cisco, USA
- Sheng Jiang, Peter Kirstein, Piers O'Hanlon, Socrates Varakliotis, UCL, UK
- R. Ruppelt, FhG, Germany
- Jacques Saint Blancat, IBM, France
- Laurent Toutain, ENST-Bretagne – IRISA, France
- Bernard Tuy, Renater, France
- Carlos Friaças, FCCN, Portugal

Agenda

- Porting Applications
- Heterogeneous Environments
- Available Applications Listings
- Services and Network Applications

Porting Applications

<Location>, <Country> - <Month> <Year>

Introduction

- **Most IPv4 Applications can be IPv6 enabled**
  - If certain precautions are taken
  - Good Programming discipline is applied
- **If there are IPv4 and IPv6 versions, most can be made dual stack**
- **Particularly satisfactory if written in a language that allows for IPv6**
  - Java is a good example
- **Affects anything that reads/writes/stores/passes IP addresses (just about every higher protocol)**
- **New DNS record type: AAAA**

Sockets API Changes

Name to Address Translation Functions

Address Conversion Functions

Address Data Structures

Wildcard Addresses

Constant Additions

Core Sockets Functions

Socket Options

New Macros

Core Sockets Functions

Core APIs

- Use IPv6 Family and Address Structures
- socket() Uses PF_INET6

Functions that pass addresses

- bind()
- connect()
- sendmsg()
- sendto()

Functions that return addresses

- accept()
- recvfrom()
- recvmsg()
- getpeername()
- getsockname()

Name to Address Translation

getaddrinfo()

- Pass in nodename and/or servicename string
  - Can Be Address and/or Port
- Optional Hints for Family, Type and Protocol
  - Flags – AI_PASSIVE, AI_CANNONNAME, AI_NUMERICHOST, AI_NUMERICSERV, AI_V4MAPPED, AI_ALL, AI_ADDRCONFIG
- Pointer to Linked List of addrinfo structures Returned
  - Multiple Addresses to Choose From

freeaddrinfo()

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

```
struct addrinfo {
    int ai_flags;
    int ai_family;
    int ai_socktype;
     int ai_protocol;
    size_t ai_addrlen;
     char *ai_canonname;
    struct sockaddr *ai_addr;
     struct addrinfo *ai_next;
     };
```

Address to Name Translation

getnameinfo()

- Pass in address (v4 or v6) and port
  - Size Indicated by salen
  - Also Size for Name and Service buffers (NI_MAXHOST, NI_MAXSERV)
- Flags
  - NI_NOFQDN
  - NI_NUMERICHOST
  - NI_NAMEREQD
  - NI_NUMERICSERV
  - NI_DGRAM

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

Porting Environments

Node Types

- IPv4-only
- IPv6-only
- IPv6/IPv4

Application Types

- IPv6-unaware
- IPv6-capable
- IPv6-required

IPv4 Mapped Addresses

Porting Issues

Running on ANY System

- Including IPv4-only

Address Size Issues

New IPv6 APIs for IPv4/IPv6

Ordering of API Calls

User Interface Issues

Higher Layer Protocol Changes

Specific things to look for

Storing IP address in 4 bytes of an array.

Use of explicit dotted decimal format in UI.

Obsolete / New:

AF_INET	replaced by	AF_INET6
SOCKADDR_IN	replaced by	SOCKADDR_STORAGE
IPPROTO_IP	replaced by	IPPROTO_IPV6
IP_MULTICAST_LOOP	replaced by	SIO_MULTIPOINT_LOOPBACK
gethostbyname	replaced by	getaddrinfo
gethostbyaddr	replaced by	getnameinfo

IPv6 literal addresses in URL’s

From RFC 2732

Literal IPv6 Address Format in URL's Syntax To use a literal IPv6 address in a URL, the literal address should be enclosed in "[" and "]" characters. For example the following literal IPv6 addresses: FEDC:BA98:7654:3210:FEDC:BA98:7654:3210

3ffe:2a00:100:7031::1

::192.9.5.5

2010:836B:4179::836B:4179

would be represented as in the following example URLs:

http://[FEDC:BA98:7654:3210:FEDC:BA98:7654:3210]:80/index.html

http://[3ffe:2a00:100:7031::1]

http://[::192.9.5.5]/ipng

http://[2010:836B:4179::836B:4179]

Other Issues

Renumbering & Mobility routinely result in changing IP Addresses –

- Use Names and Resolve, Don’t Cache

Multi-homed Servers

- More Common with IPv6
- Try All Addresses Returned

Using New IPv6 Functionality

Porting Steps -Summary

Use IPv4/IPv6 Protocol/Address Family

Fix Address Structures

- in6_addr
- sockaddr_in6
- sockaddr_storage to allocate storage

Fix Wildcard Address Use

- in6addr_any, IN6ADDR_ANY_INIT
- in6addr_loopback, IN6ADDR_LOOPBACK_INIT

Use IPv6 Socket Options

- IPPROTO_IPV6, Options as Needed

Use getaddrinfo()

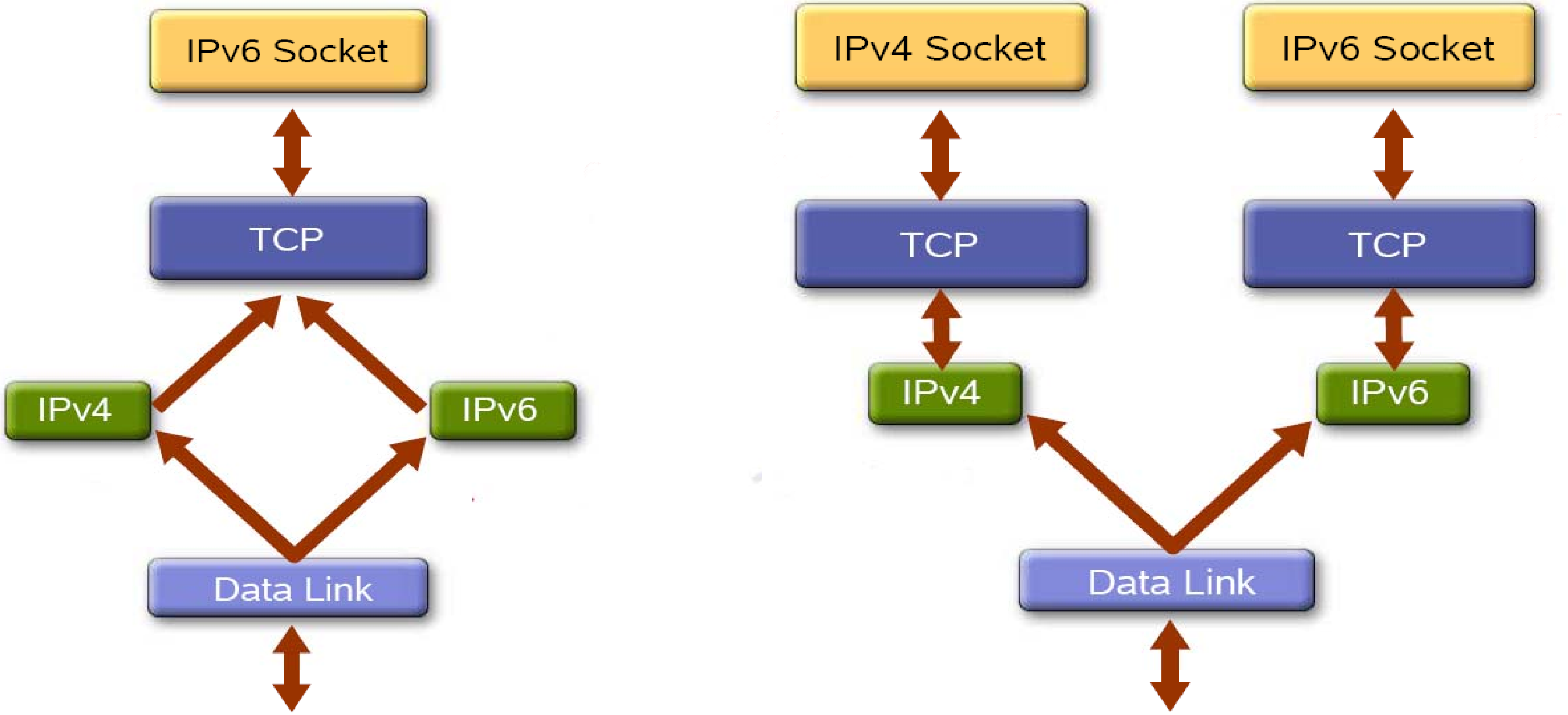
- For Address Resolution

Dual stack

Both IPv4 and IPv6 stacks will be available during the transition period

Dual network stack machine will allow to provide a service both for IPv4 and IPv6

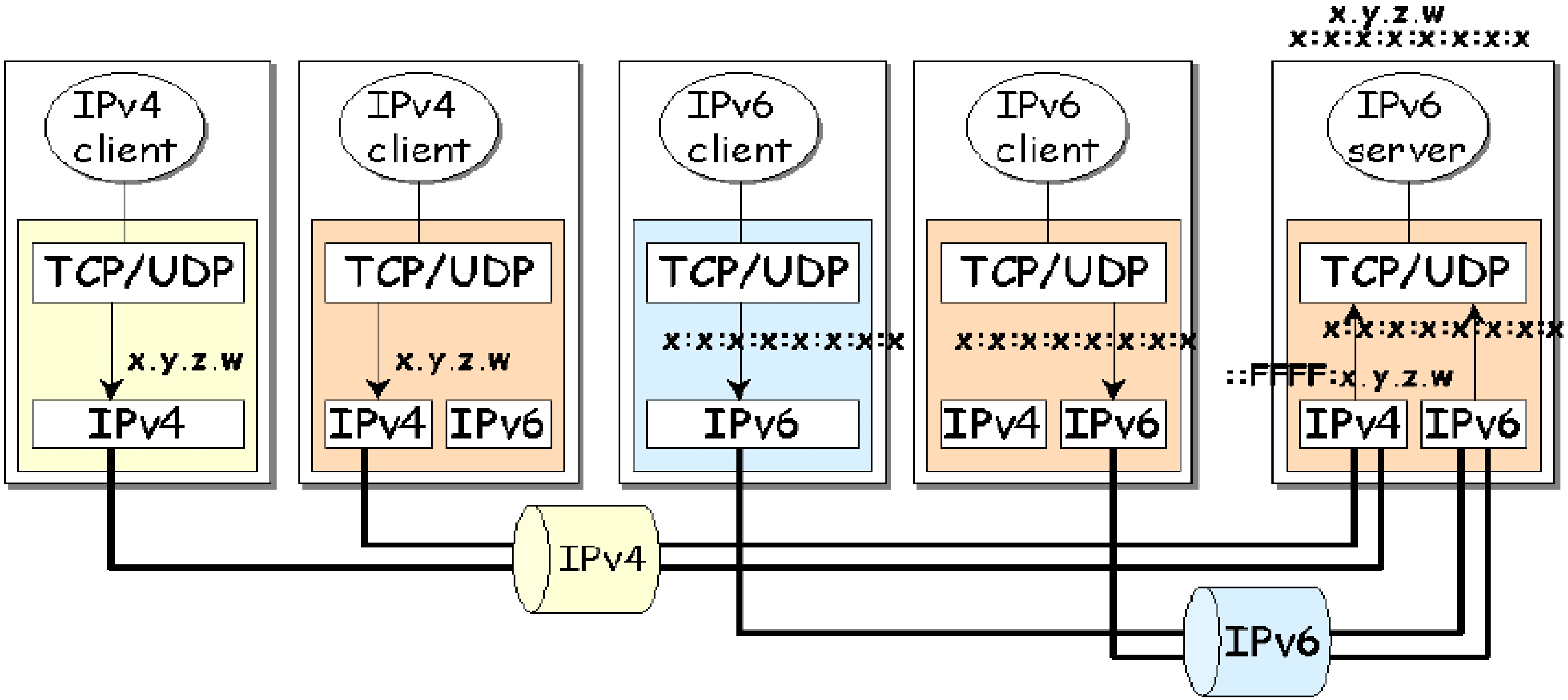
2 different implementations of network stack



Source : Rino Nucara, GARR, EuChinaGRID IPv6 Tutorial

Mapping IPv4 address in IPv6

IPv6/IPv4 Clients connecting to an IPv6 server at dual stack node → 1 socket

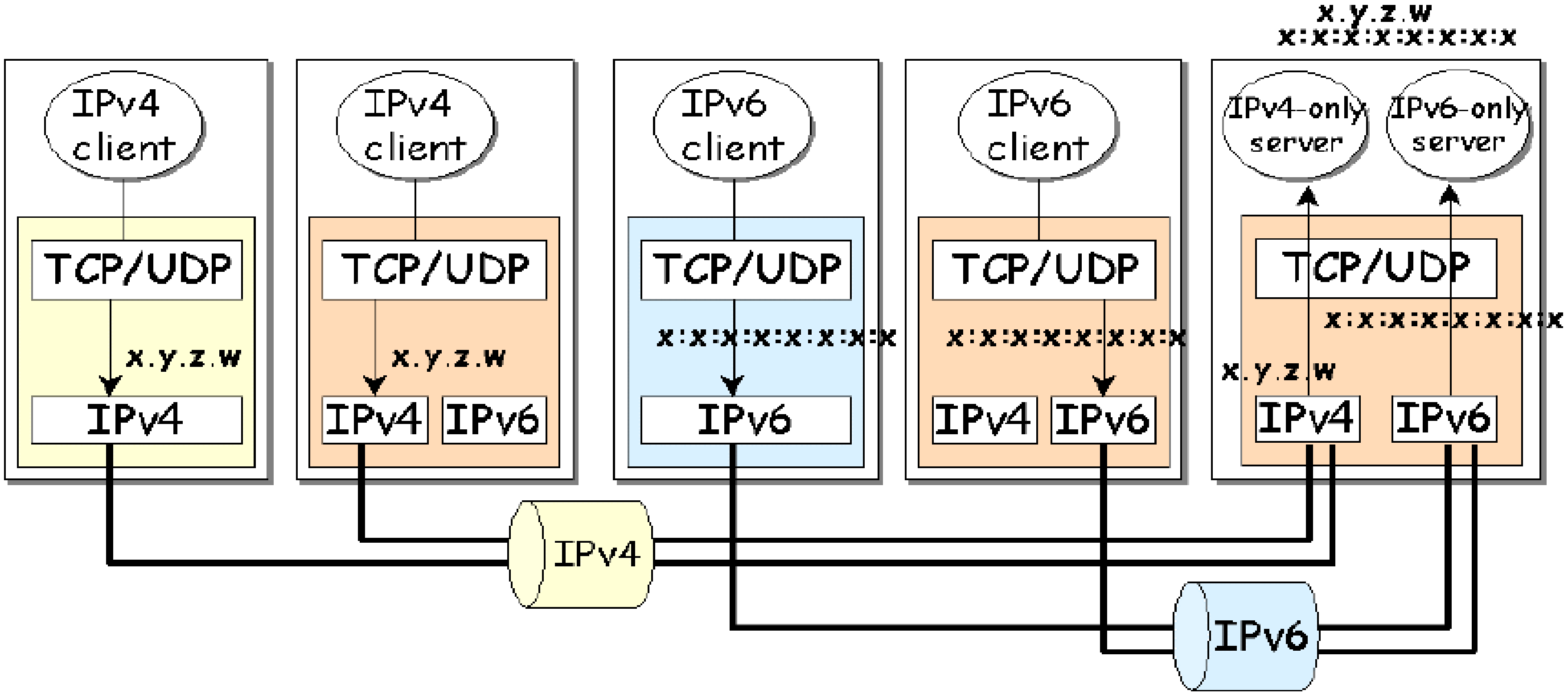


🟧 *Dual Stack* 🟨🟦 *Single IPv4 or IPv6 stacks*

Source : Programming guidelines on transition to IPv6 T. P de Miguel, E. M. Castro

IPv4-only and IPv6-only

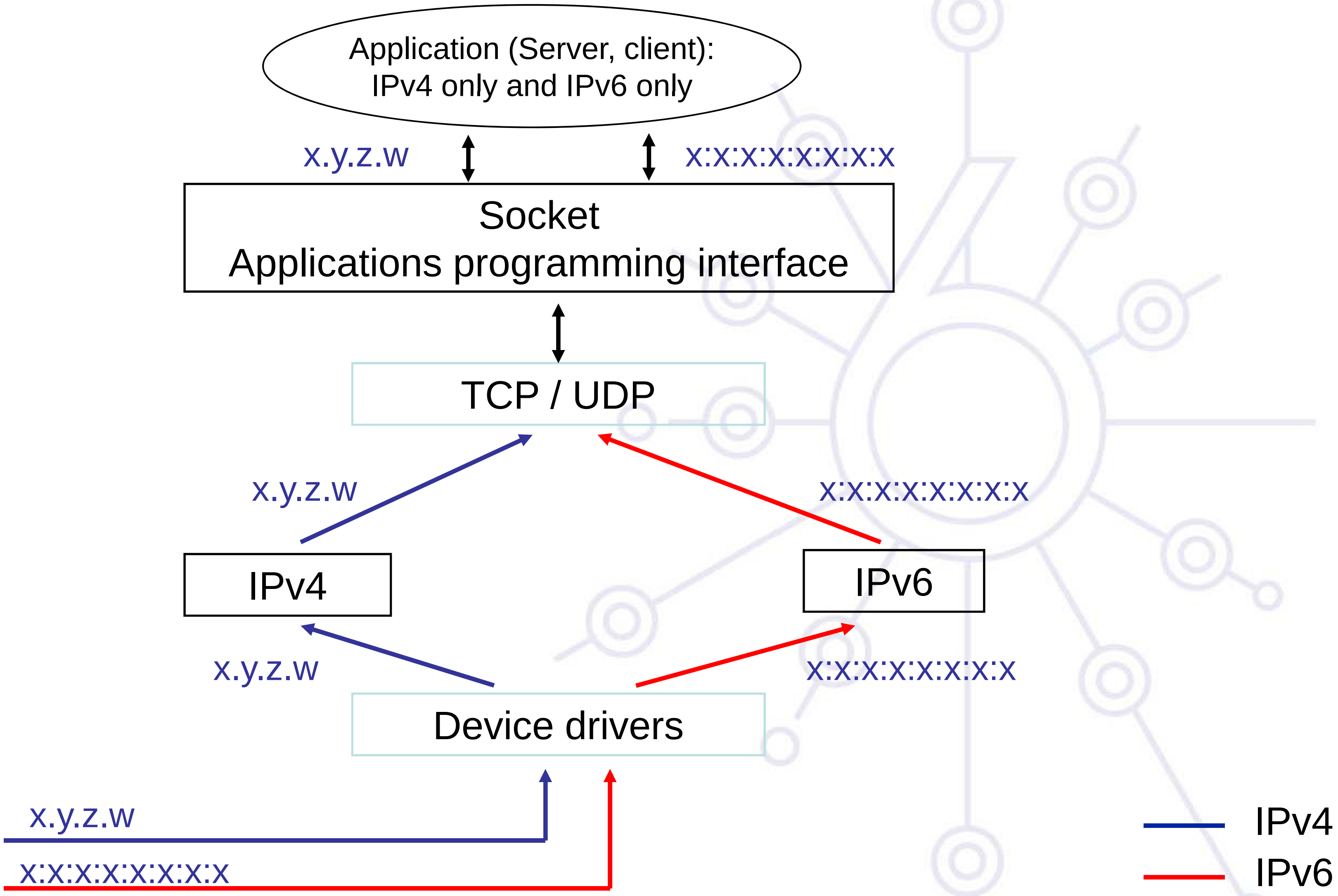
IPv6/IPv4 Clients connecting to an IPv4-only server and IPv6 onlv server at dual stack node → 2 sockets



🟧 *Dual Stack or separated stack* 🟨🟦 *Single IPv4 or IPv6 stacks*

Source : Programming guidelines on transition to IPv6 T. P de Miguel, E. M. Castro

IPv4 / IPv6 interoperability



Application (Server, client):
IPv4 only and IPv6 only

Socket
Applications programming interface

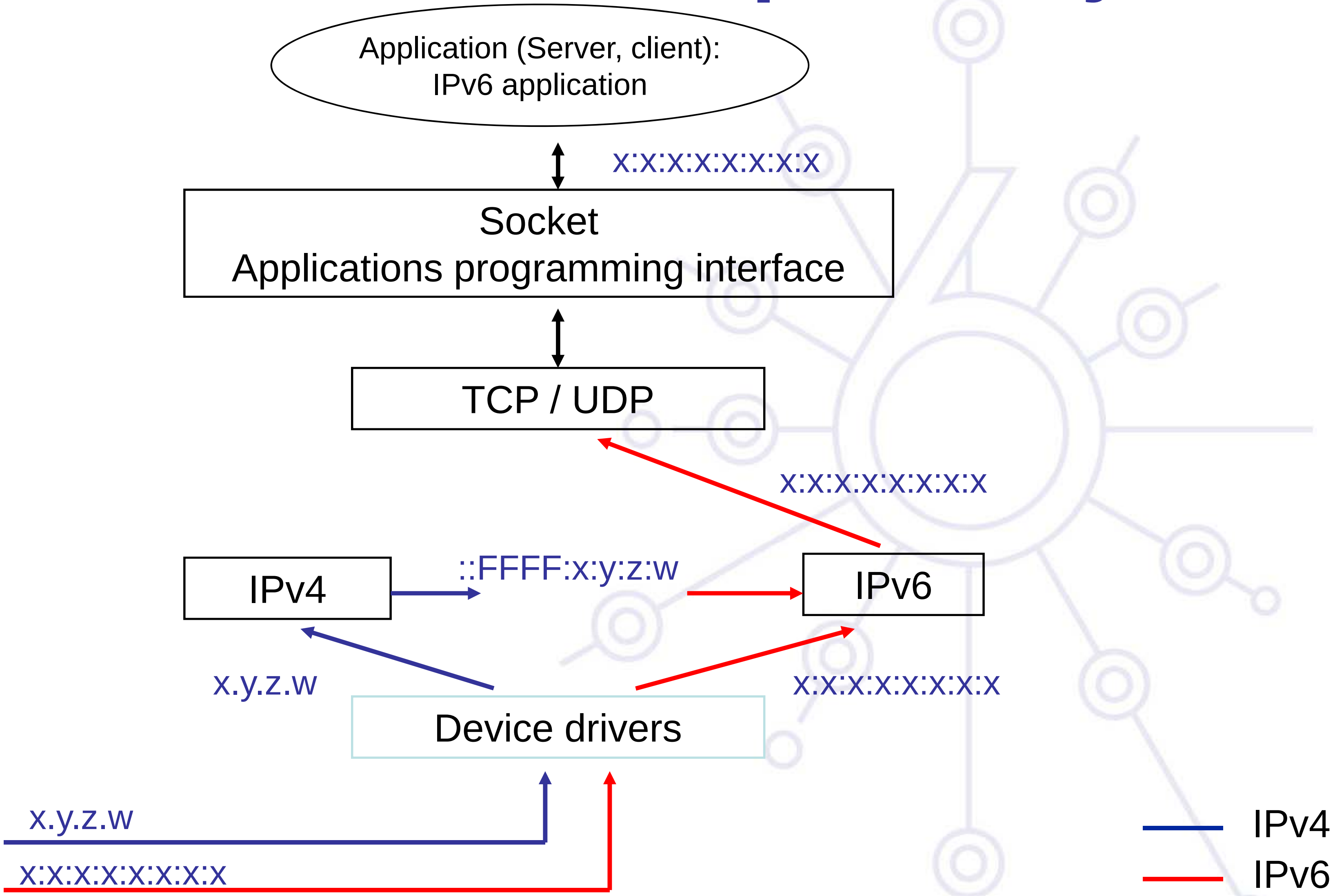
TCP / UDP

IPv4

IPv6

Device drivers

IPv4 / IPv6 interoperability



Application (Server, client):
IPv6 application

x:x:x:x:x:x:x:x

Socket
Applications programming interface

TCP / UDP

x:x:x:x:x:x:x:x

IPv4

::FFFF:x:y:z:w

IPv6

x.y.z.w

x:x:x:x:x:x:x:x

Device drivers

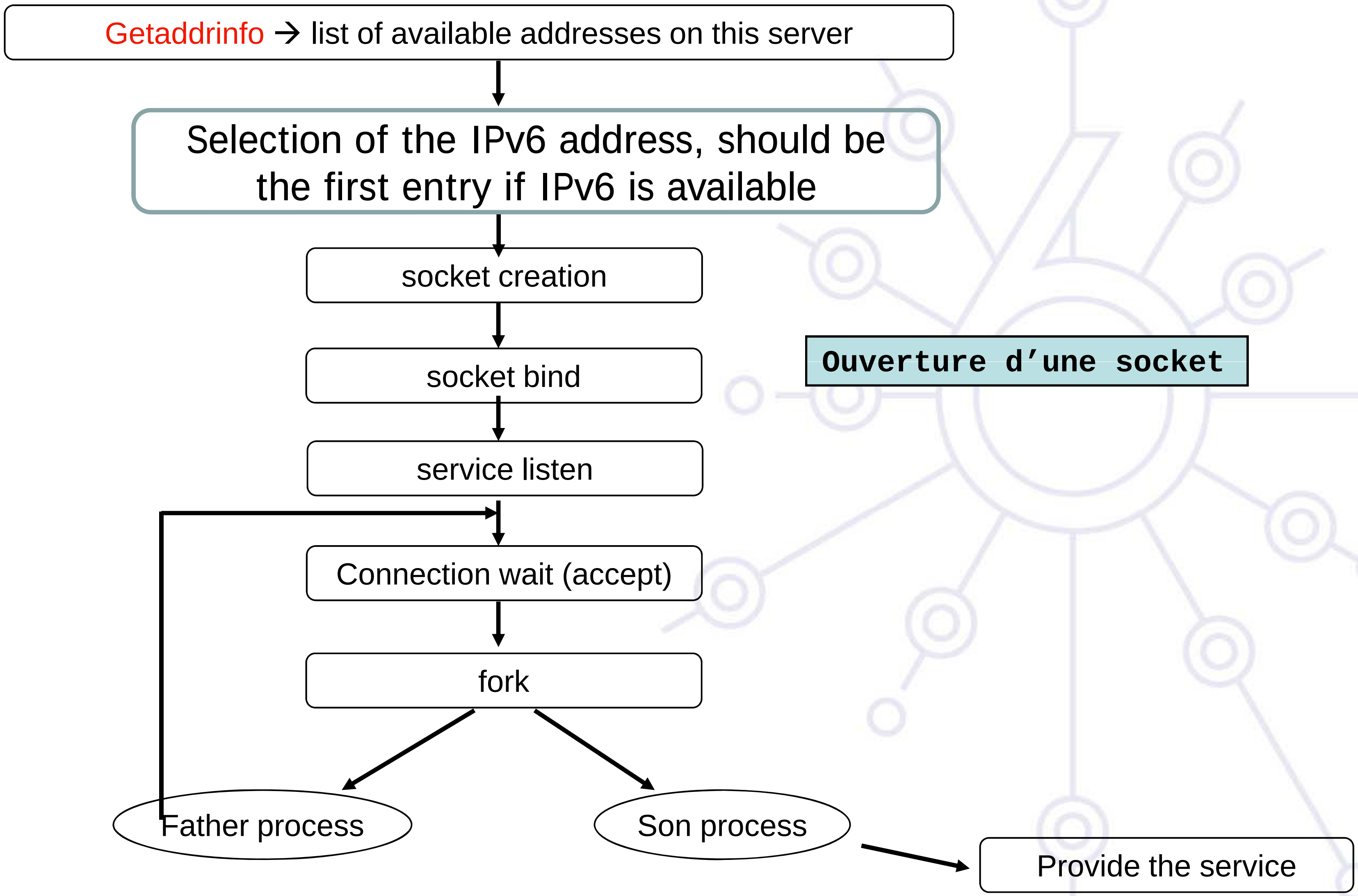
x.y.z.w

x:x:x:x:x:x:x:x

IPv4

IPv6

Algorithm of low level programming of a server 1/2



Getaddrinfo → list of available addresses on this server

Selection of the IPv6 address, should be the first entry if IPv6 is available

socket creation

socket bind

service listen

Connection wait (accept)

fork

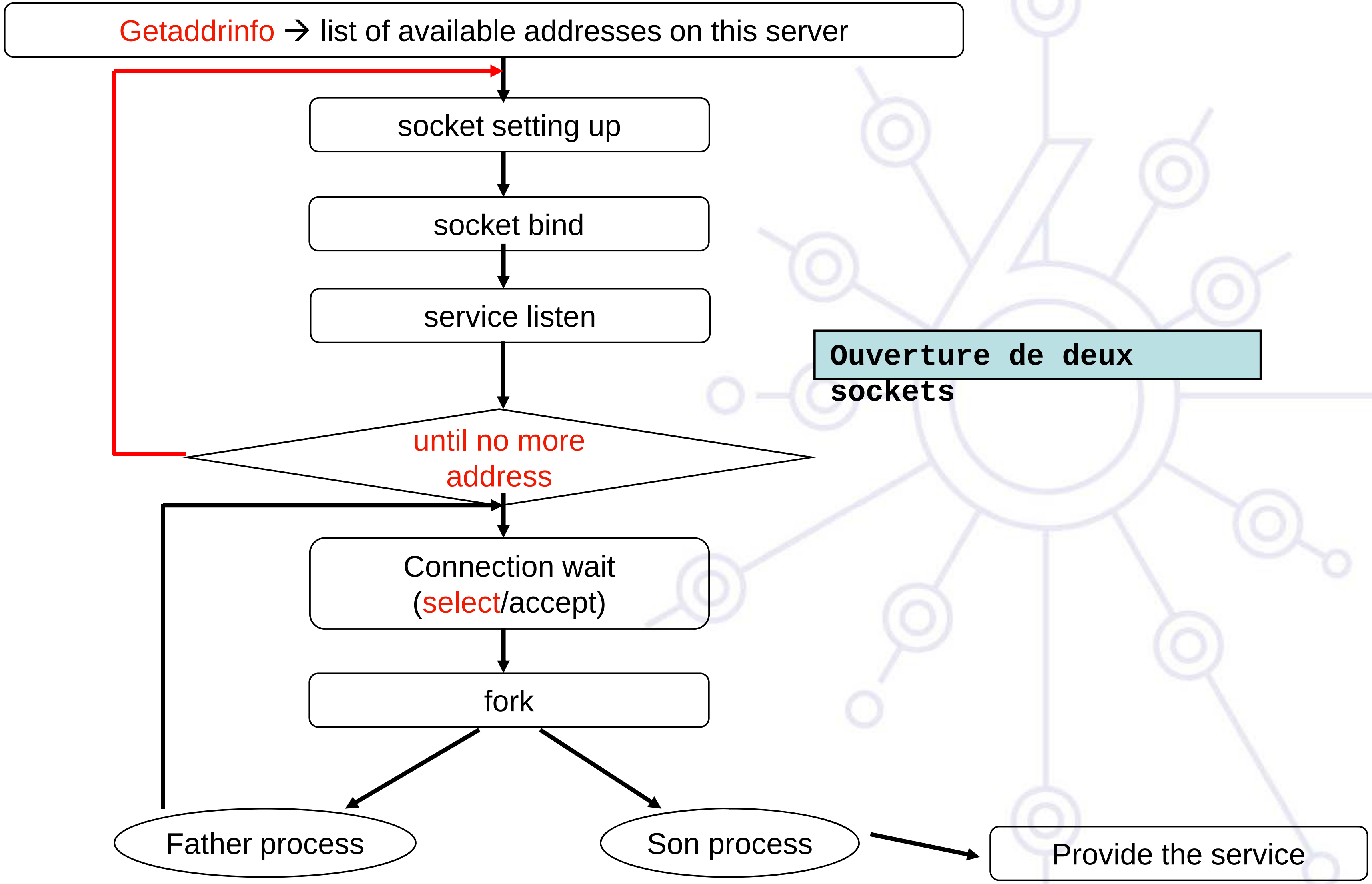
Father process

Son process

Provide the service

```
Ouverture d’une socket
```

Algorithm of low level programming of a server 2/2



Ouverture de deux sockets

Heterogeneous Environments

<Location>, <Country> - <Month> <Year>

Precautions for Dual Stack

Avoid any explicit use of IP addresses

- Normally do Call by Name (i.e. make use of DNS!)

Ensure that calls to network utilities are concentrated in one subroutine

Ensure that libraries and utilities used support both stacks

Do not request utilities that would not exist in both stacks

- E.g. IPsec, MIP, Neighbour Discovery may vary

New Applications

For new Apps, some can use high-level language

- JAVA fully supports dual stack

Examples of utilities that must so support

- DNS, SSH, FTP, Web server, Resource Location

Examples of libraries and applications that must so support

- RTP library, NTP time protocol, Web browser,
  IPsec library

Legacy Applications

If most parts are written in say Java, and small parts in say C, try to rewrite C part to be in Java or at least make sure that I/O is concentrated in certain regions

Potentially re-arrange code so that it fits needs of earlier slide

Adjust I/f to code to fit dual-stack specs

- Or do all networking via a utility which is IPv6-enabled
- VIC, RAT using RTP are good example

Heterogeneous IPv4/IPv6 Environments

May require dual-stack client/server, accessible by both stacks

- Often used, for example, with Web services and with SIP signalling

May require transition gateway

- As for example with IPv4 telephones accessing other IPv6 ones

May be very difficult, as when encrypted IPv4 messages are passed into the IPv6 networks with packet header encrypted, or certificate cryptographically bound to IP4 address

Available Applications Listings

<Location>, <Country> - <Month> <Year>

Available IPv6 Enabled Applications

Many have been tested under 6NET, Description given in

- http://6net.iif.hu/ipv6_apps

Check if your application is IPv6 enable

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

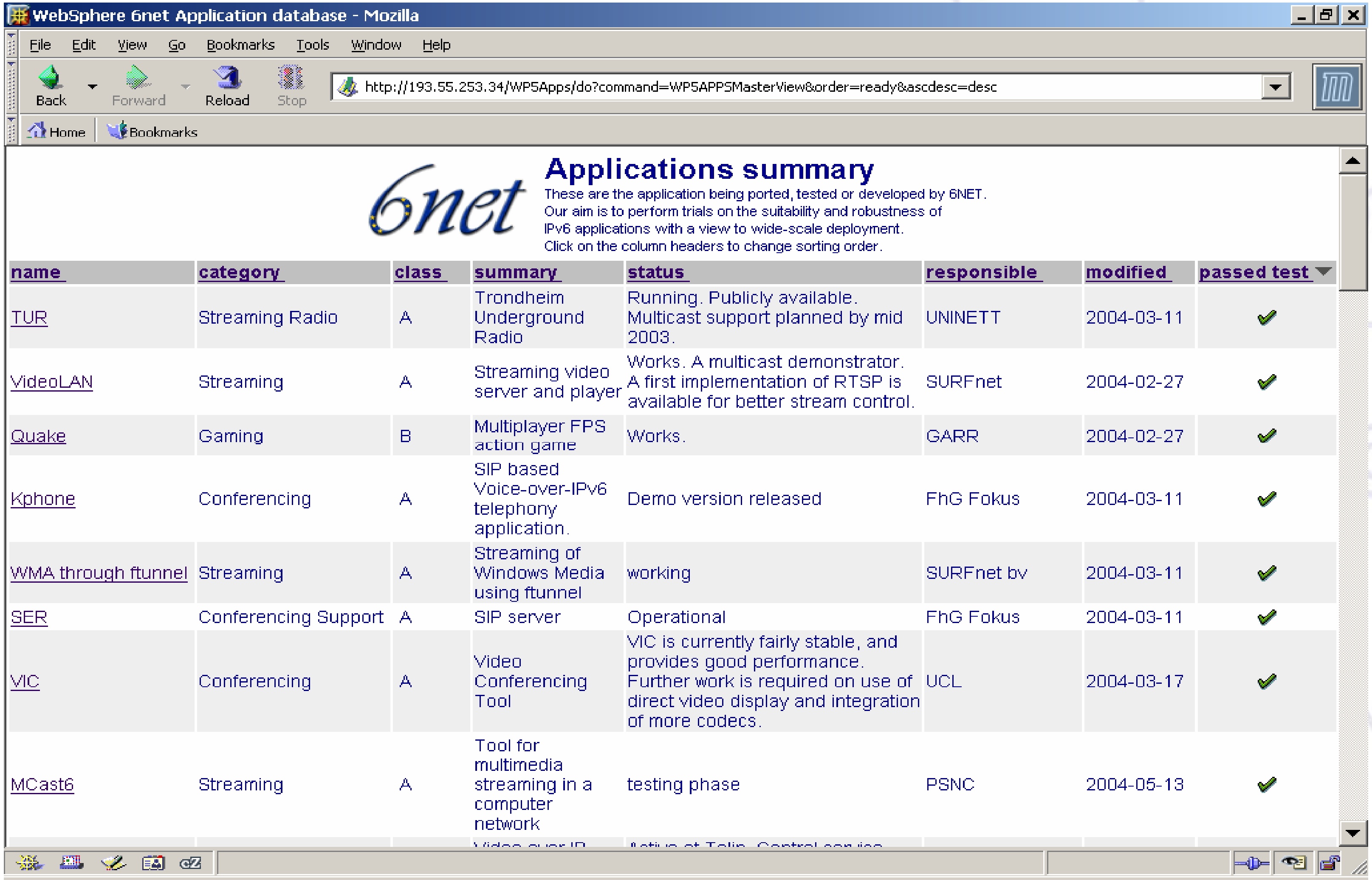
Most currently useful utilities exist, e.g.

- SIP, WWW, RTP, SSH, MIP, IPsec, NTP

6NET Deliverables discuss their use

- Particularly those of WP5

6net Application DB



WebSphere 6net Application database - Mozilla

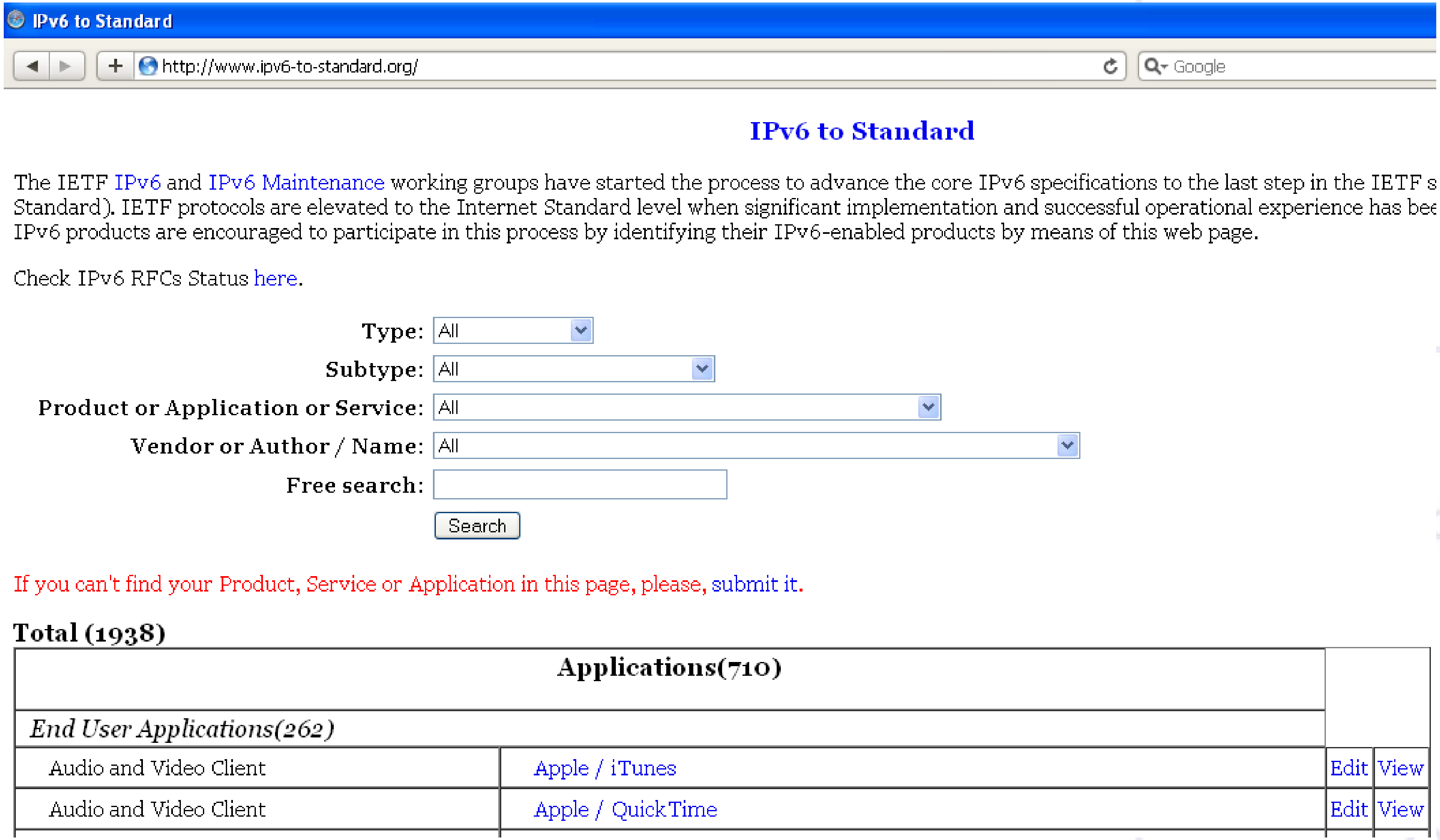
http://193.55.253.34/WP5Apps/do?command=WP5APPSMasterView&order=ready&ascdesc=desc

Applications summary

These are the application being ported, tested or developed by 6NET. Our aim is to perform trials on the suitability and robustness of IPv6 applications with a view to wide-scale deployment. Click on the column headers to change sorting order.

name	category	class	summary	status	responsible	modified	passed test
TUR	Streaming Radio	A	Trondheim Underground Radio	Running. Publicly available. Multicast support planned by mid 2003.	UNINETT	2004-03-11	✓
VideoLAN	Streaming	A	Streaming video server and player	Works. A multicast demonstrator. A first implementation of RTSP is available for better stream control.	SURFnet	2004-02-27	✓
Quake	Gaming	B	Multiplayer FPS action game	Works.	GARR	2004-02-27	✓
Kphone	Conferencing	A	SIP based Voice-over-IPv6 telephony application.	Demo version released	FhG Fokus	2004-03-11	✓
WMA through ftunnel	Streaming	A	Streaming of Windows Media using ftunnel	working	SURFnet bv	2004-03-11	✓
SER	Conferencing Support	A	SIP server	Operational	FhG Fokus	2004-03-11	✓
VIC	Conferencing	A	Video Conferencing Tool	VIC is currently fairly stable, and provides good performance. Further work is required on use of direct video display and integration of more codecs.	UCL	2004-03-17	✓
MCast6	Streaming	A	Tool for multimedia streaming in a computer network	testing phase	PSNC	2004-05-13	✓

IPv6 to Standard



IPv6 to Standard

The IETF IPv6 and IPv6 Maintenance working groups have started the process to advance the core IPv6 specifications to the last step in the IETF s
Standard). IETF protocols are elevated to the Internet Standard level when significant implementation and successful operational experience has bee
IPv6 products are encouraged to participate in this process by identifying their IPv6-enabled products by means of this web page.

Check IPv6 RFCs Status here.

Type:	All
Subtype:	All
Product or Application or Service:	All
Vendor or Author / Name:	All
Free search:	

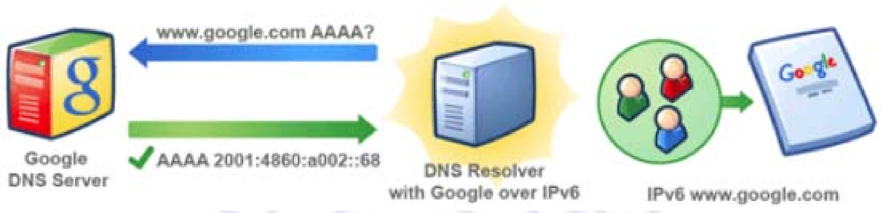
If you can't find your Product, Service or Application in this page, please, submit it.

Total (1938)

Applications(710)			
End User Applications(262)			
Audio and Video Client	Apple / iTunes	Edit	View
Audio and Video Client	Apple / QuickTime	Edit	View

Content: Now available over IPv6!

- **Google (+YouTube, +...)**
  - www.google.com/ipv6
  - For whitelisted networks
    - (i.e. DNS resolvers)



- **Facebook**
  - www.v6.facebook.com
    - different DNS name path

facebook

Source : L.Colitti (Google); D.Lee (Facebook); Google IPv6 Implementors Conference 2010

Services & Network Applications

<Location>, <Country> - <Month> <Year>

Applications / Services

Basic applications

- MUAs, MTAs
- Web browsers & servers,
- FTP, SSH, Telnet

Advanced applications

- Videoconferencing tools, streaming, ...
- Editors, Games, ...
- Management and monitoring tools

Basic applications: Mail

Server:

- Qmail (Unix/Linux/xBSD)
- Sendmail (Unix/Linux/xBSD)
- Postfix (Linux)
- Dovecot (Linux)

Client:

- Thunderbird (all platforms)
- Inframail (windows/xBSD)
- Evolution (Linux)

E-Mail/Postfix

- Postfix >= 2.2 supports IPv6
- Example: /etc/postfix/main.cf

```
inet_protocols = ipv4, ipv6
smtp_bind_address6 = 2001:db8:1:1::1600
smtp_bind_address = 172.16.250.1
inet_interfaces = 2001:db8:1:1::1600, localhost
mynetworks = [2001:db8:1:1::]/64 172.16.250.0/24 [::1]/128 127.0.0.0/8
```

- Answers on port 25, both in IPv4 and IPv6

E-Mail

- It’s not only the MX(s) server(s) who need IPv6 addresses...
  - The servers from where your users retrieve e-mail (POP, IMAP, ...) can also start operating with IPv6 enabled

- Transparency !!!

Basic applications: Web

Server:

- Apache2 (all platforms)
- thttpd (Unix/Linux/xBSD)
- IIS (windows)

Client:

- Firefox (all platforms)
- Internet Explorer (windows)
- Safari (MacOS, windows)
- Opera (windows, ...)
- Wget (Unix/Linux/xBSD)
- Lynx (Unix/Linux/xBSD)
- Symbian S60 webclients

Web / Apache

- Apache >= 2.x supports IPv6
- Directives
  - Listen 80 (place only *port* and not an IP address)
  - NameVirtualHost <address> (place [ ] before and after the IPv6 address)
  - VirtualHost <endereço> (place [ ] before and after the IPv6 address)
- Example: httpd.conf

```
Listen 80
NameVirtualHost [2001:690:1fff:200:20e:cff:fe31:c81f]
<VirtualHost [2001:690:1fff:200:20e:cff:fe31:c81f]>
        DocumentRoot /usr/local/apache2/htdocs/lg
        ServerAdmin ip6adm@fccn.pt
        ServerName lg.ip6.fccn.pt
        ServerAlias lg.tbed.ip6.fccn.pt
        ServerSignature email
</VirtualHost>
```

Basic applications: FTP

Server:

- Ftpd(Unix/Linux/xBSD)
- vsFTP (all platforms)
- Pure-ftpd(all platforms)

Client:

- Filezilla (all platforms)
- Ncftp (Windows, MAC, Linux)
- Fget (Unix/Linux/xBSD)
- Wget (Unix/Linux/xBSD)

FTP

- VsFTP >= 2.0.x supports IPv6
- Example: /etc/xinetd.d/vsftpd

```
service ftp
{
        socket_type     = stream
        wait             = no
        user             = root
        server           = /usr/local/sbin/vsftpd
        server_args     = /etc/vsftpd.conf
      flags             = IPv6
        nice             = 10
        disable          = no
}
```

- Answer on port 21, both in IPv4 and IPv6

Basic applications: SSH,telnet

Server:

- sshd (Unix/Linux/xBSD)
- Openssh (Unix/Linux/xBSD)

Client:

- puTTY (all platforms)
- telnet (Unix/Linux/xBSD), unsecure...

Advanced applications

Videoconferencing tools, streaming:

- Videolan (all platforms)
    - IPv6 unicast/multicast streaming, www.videolan.org
- Ekiga (former GnomeMeeting)
    - H323 application
- OpenH323 (all platforms)
- ISABEL
- DVTS

Multicast

Peer to peer applications

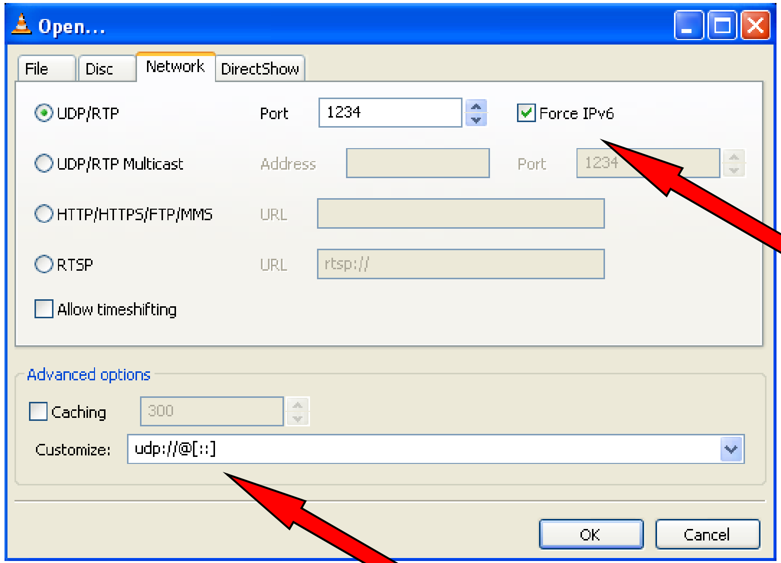
- Gnutella (all platforms)
- Azureus
- uTorrent
- ...

Video *On-Demand*

- *Windows Media Services* 9 (>Win2003 *Server*)

- VideoLan Tool
  - www.videolan.org

0:01:11 / 0:10:53 x1.00 E:\ELEPHA~2.AVI



● UDP/RTP Port 1234 ☑ Force IPv6

○ UDP/RTP Multicast Address Port 1234

○ HTTP/HTTPS/FTP/MMS URL

○ RTSP URL rtsp://

□ Allow timeshifting

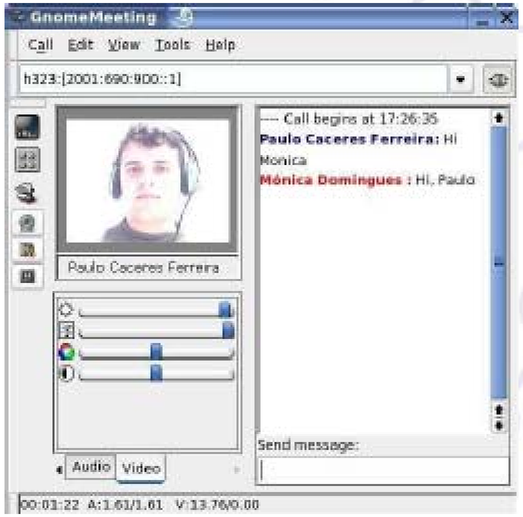
Advanced options

□ Caching 300

Customize: udp://@[::]

Videoconference

- Several Components
  - H.323 Managers: OpenMCU
  - Clients: Ekiga (former GnomeMeeting), ConferenceXP



h323:[2001:690:9D0::1]

--- Call begins at 17:26:35
Paulo Caceres Ferreira: Hi Monica
Mónica Domingues : Hi, Paulo

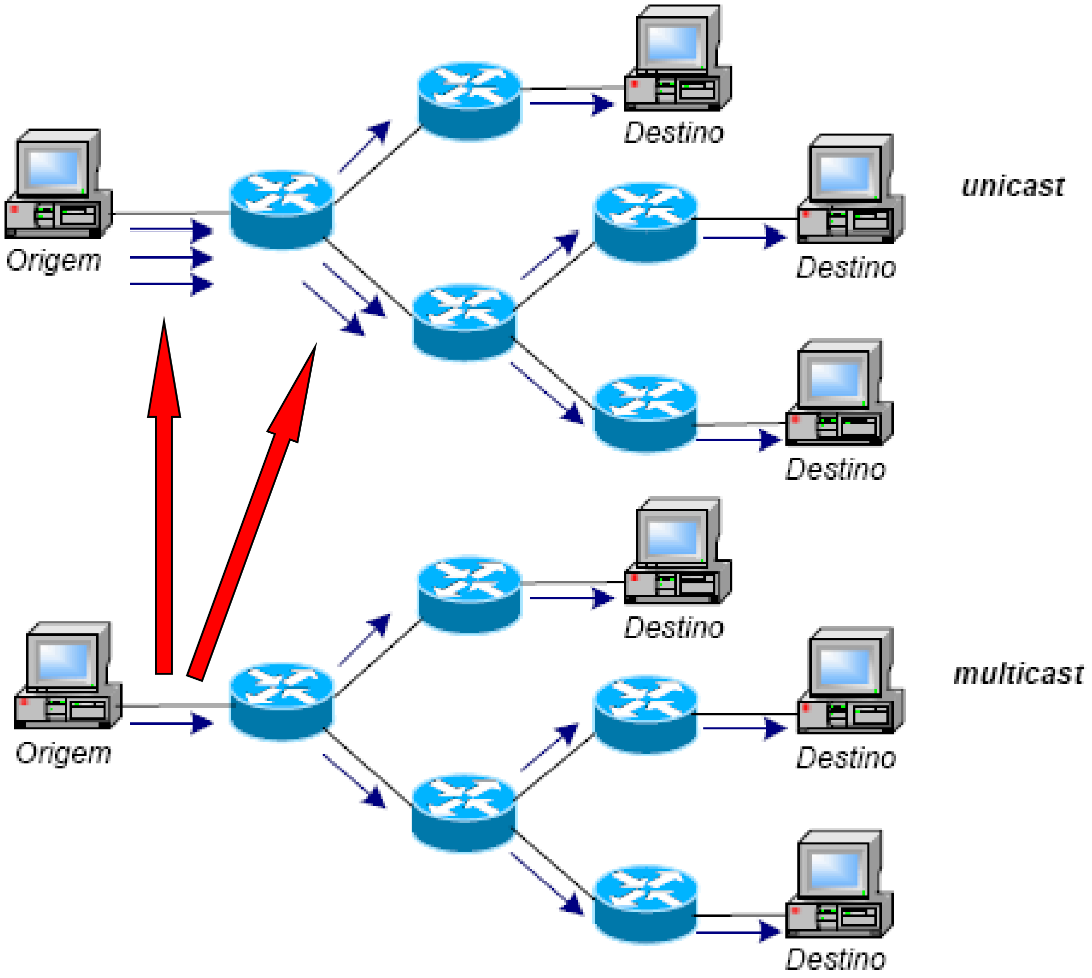
Paulo Caceres Ferreira

Send message:

00:01:22 A:1.61/1.61 V:13.76/0.00

Multicast

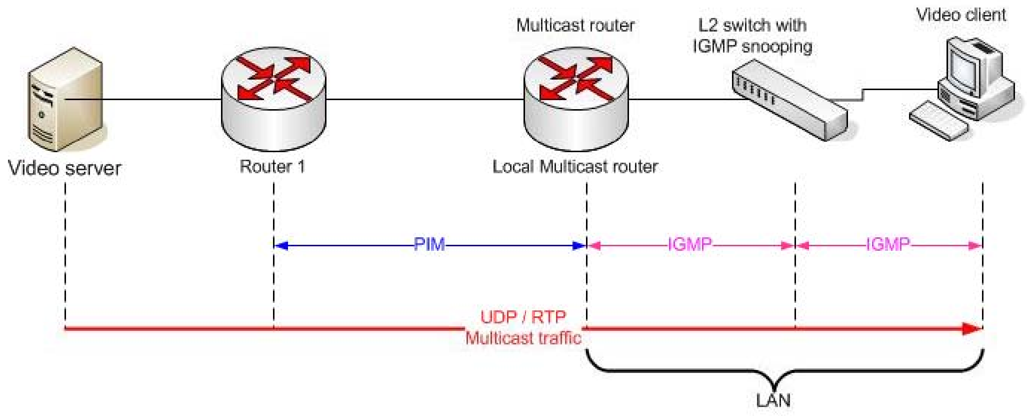
- Goal: Traffic flows savings
- The architecture is difficult to maintain and operate at the interdomain level (between different ISP networks)
- With IPv6, the concept of *Source Specific Multicast* (SSM) starts to be used



Multicast

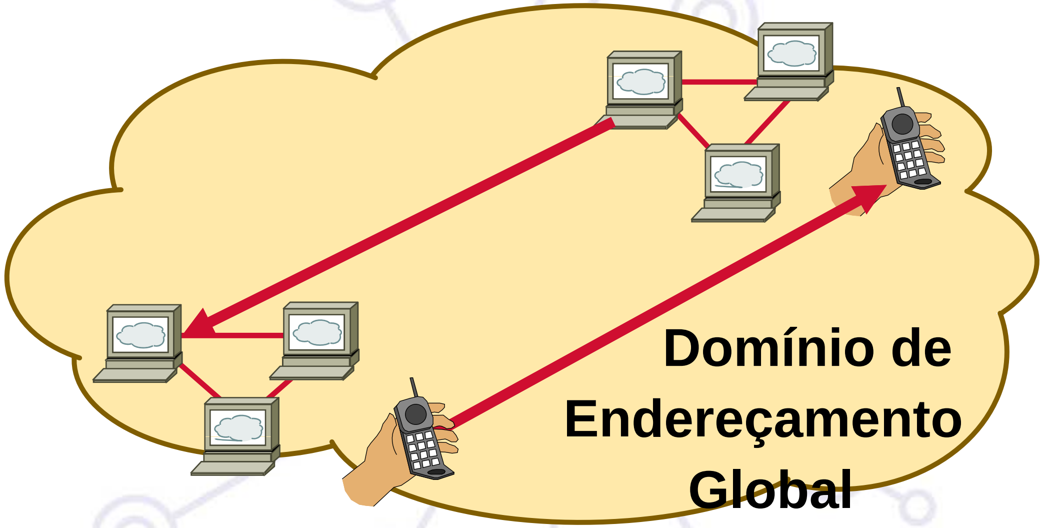
- IPv4: IGMP, *Internet Group Management Protocol*
- IPv6: MLD, *Multicast Listener Discovery*

- Protocols to manage Multicast groups
  - used between the client and the *gateway*
  - stop undesired traffic to hit switch ports which are not acessing Multicast groups



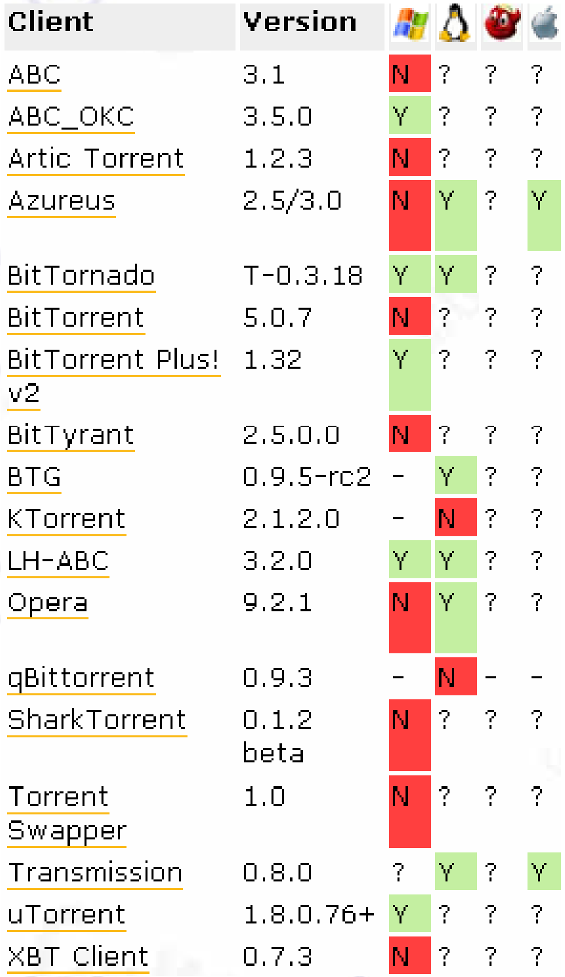
P2P - *Peer-to-Peer*

- Virtually all nodes provide a service
- Critical Infrastructure:
    - DNS
    - «Rendez-Vouz» Service/«Tracker»
- Without restrictions regarding the initiator party
- All participants share a consistent network vision



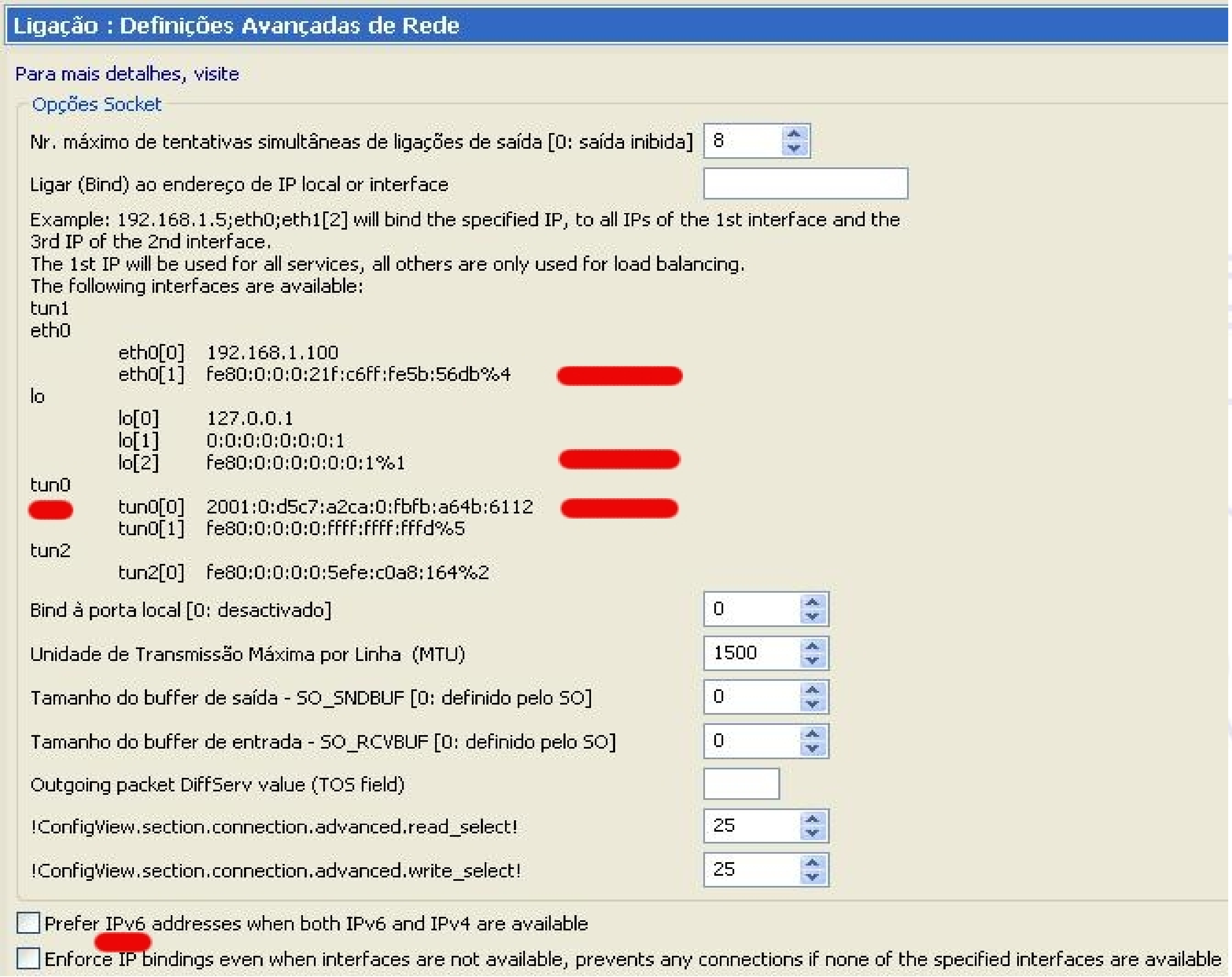
P2P - Bittorrent

- Protocol created in 2002
- There are loads of legal content accessible through this protocol:
  - http://fedoraproject.org/en/get-fedora
- IPv6 Support in some clients
- Always platform dependent
  - Win/Linux/BSD/Mac
- IPv6 Communication with:
  - «Tracker»
  - Other clients



Client	Version				
ABC	3.1	N	?	?	?
ABC_OKC	3.5.0	Y	?	?	?
Artic Torrent	1.2.3	N	?	?	?
Azureus	2.5/3.0	N	Y	?	Y
BitTornado	T-0.3.18	Y	Y	?	?
BitTorrent	5.0.7	N	?	?	?
BitTorrent Plus! v2	1.32	Y	?	?	?
BitTyrant	2.5.0.0	N	?	?	?
BTG	0.9.5-rc2	-	Y	?	?
KTorrent	2.1.2.0	-	N	?	?
LH-ABC	3.2.0	Y	Y	?	?
Opera	9.2.1	N	Y	?	?
qBittorrent	0.9.3	-	N	-	-
SharkTorrent	0.1.2 beta	N	?	?	?
Torrent Swapper	1.0	N	?	?	?
Transmission	0.8.0	?	Y	?	Y
uTorrent	1.8.0.76+	Y	?	?	?
XBT Client	0.7.3	N	?	?	?

P2P - Azureus



Ligação : Definições Avançadas de Rede

Para mais detalhes, visite

Opções Socket

Nr. máximo de tentativas simultâneas de ligações de saída [0: saída inibida] 8

Ligar (Bind) ao endereço de IP local or interface

Example: 192.168.1.5;eth0;eth1[2] will bind the specified IP, to all IPs of the 1st interface and the 3rd IP of the 2nd interface.
The 1st IP will be used for all services, all others are only used for load balancing.
The following interfaces are available:

```
tun1
eth0
        eth0[0]   192.168.1.100
        eth0[1]   fe80:0:0:0:21f:c6ff:fe5b:56db%4
lo
        lo[0]     127.0.0.1
        lo[1]     0:0:0:0:0:0:0:1
        lo[2]     fe80:0:0:0:0:0:0:1%1
tun0
        tun0[0]   2001:0:d5c7:a2ca:0:fbfb:a64b:6112
        tun0[1]   fe80:0:0:0:0:ffff:ffff:fffd%5
tun2
        tun2[0]   fe80:0:0:0:0:5efe:c0a8:164%2
```

Bind à porta local [0: desactivado] 0

Unidade de Transmissão Máxima por Linha (MTU) 1500

Tamanho do buffer de saída - SO_SNDBUF [0: definido pelo SO] 0

Tamanho do buffer de entrada - SO_RCVBUF [0: definido pelo SO] 0

Outgoing packet DiffServ value (TOS field)

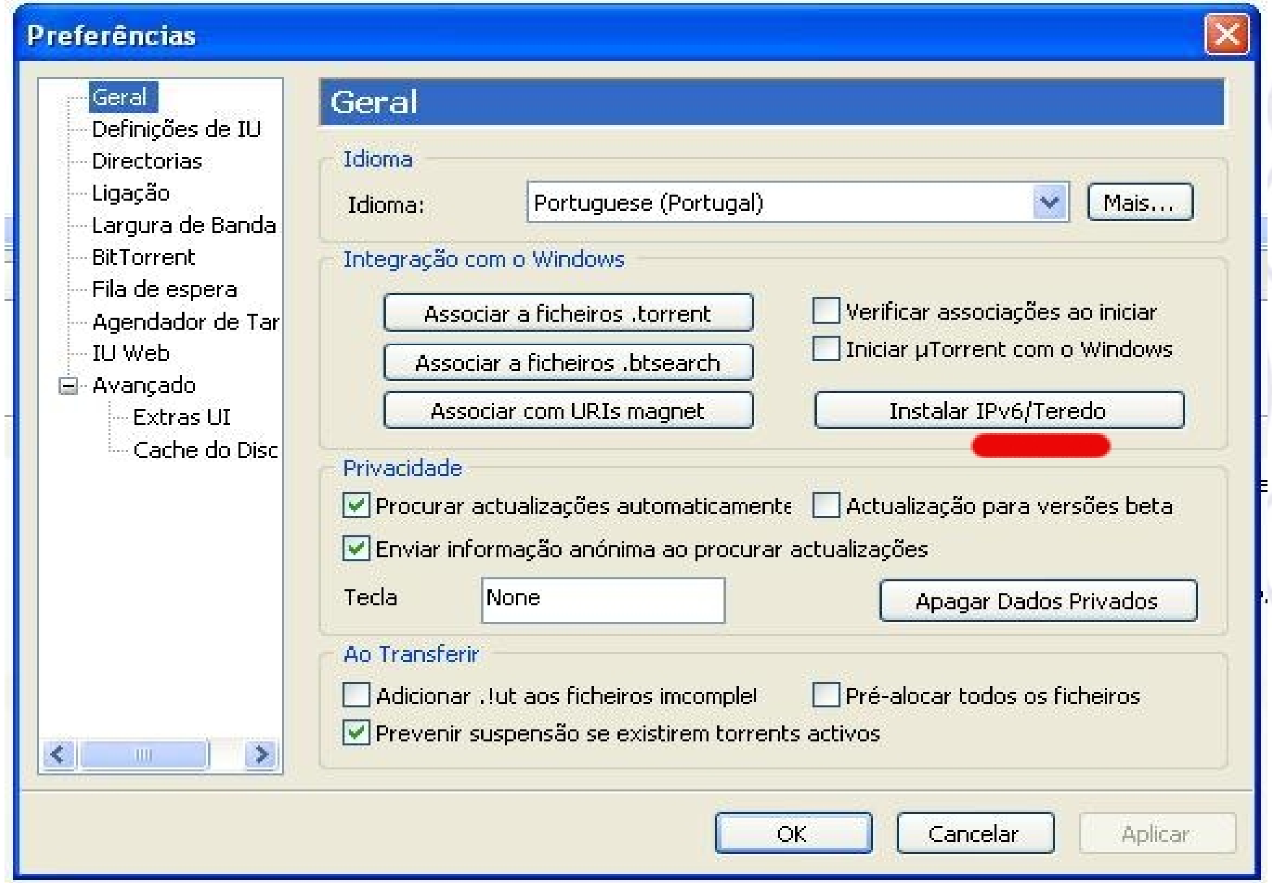
!ConfigView.section.connection.advanced.read_select! 25

!ConfigView.section.connection.advanced.write_select! 25

□ Prefer IPv6 addresses when both IPv6 and IPv4 are available

□ Enforce IP bindings even when interfaces are not available, prevents any connections if none of the specified interfaces are available

P2P - uTorrent



Geral

Idioma

Idioma: Portuguese (Portugal)

Integração com o Windows

□ Verificar associações ao iniciar

□ Iniciar µTorrent com o Windows

Privacidade

☑ Procurar actualizações automaticamente □ Actualização para versões beta

☑ Enviar informação anónima ao procurar actualizações

Tecla None

Ao Transferir

□ Adicionar .!ut aos ficheiros imcomplel □ Pré-alocar todos os ficheiros

☑ Prevenir suspensão se existirem torrents activos

Conclusion

IPv4-only applications are now IPv6-enabled

- Basic & Advanced

New services/applications/paradigms will take a step forward into IPv6 ?

- Grid Computing
- Cloud Computing

Questions

<Location>, <Country> - <Month> <Year>