IPv6 Applications

6DEPLOY – IPv6 Deployment and Support

Copy …Rights

This slide set is the ownership of the 6DEPLOY project 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.org

Looking for a contact ?

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

Agenda

IPv6 Application overview

Enabling application for IPv6

VoIP/Conferencing

Grid

Introduction

All major Operating systems are IPv6-enabled

- WinXP/Vista/Win7, MacOSX, Linux, FreeBSD, AIX, HPUX

There are already many IPv6-enabled applications

- E.g. Internet Explorer, Firefox, Apache, SSH, …

It is not hard to provide basic IPv6-support

- A little more difficult to do it well!

Content: Now available over IPv6!

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

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

- **0.2% (~2000) of top 1 Million sites have IPv6**
  - From Alexa list - They have IPv6 DNS records (AAAA)

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

Applications/Services

Core applications

- Web browsers & servers,
- Mail User Agents and Transport Agents
- FTP, SSH, Telnet

Advanced applications

- Videoconferencing tools, streaming, …
- Grid, P2P, Games, …
- Management and monitoring tools

Core applications: Web

Client:

- Firefox (all platforms)
- Internet Explorer (Windows)
- Safari (MacOSX)
- Wget (Unix/Linux/xBSD)
- …

Server:

- Apache2 (All platforms)
- IIS (Windows)
- …

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>
```

Core applications: Mail

Client:

- Thunderbird (all platforms)
- Mail (Mac OSX)
- Outlook Mail (Windows)
- Inframail (Windows/xBSD)

Server:

- Qmail (Unix/Linux/xBSD)
- Sendmail 8.10
- Postfix 2.2+
- Exim 4.3+…

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 !!!

Core applications: File transfer

Client:

- Filezilla (All Platforms)
- Ncftp (All Platforms)
- Fget (Unix/Linux/xBSD)
- Rsync (All platforms)

Server:

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

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

Core applications: SSH,telnet

Client:

- Openssh (all platforms)
- PuTTY (all platforms)
- telnet (all platforms)

Server:

- Openssh (All platforms)
- sshd (Unix/Linux/xBSD)
- telnetd (All platforms)

Advanced applications

Videoconferencing tools

- Ekiga (Win, Linux) – dev version
- Media tools VIC & RAT (All Platforms)
- ISABEL (Linux) – Managed conferencing
- LinPhone (Linux, Win)

Streaming

- Windows Media player (v9 onwards)
- Quicktime (Win, OSX)
- VLC (All platforms)
  - IPv6 unicast/multicast streaming
- MPlayer

Advanced applications (2)

Peer to peer applications

- Bittorrent (All platforms)
  - uTorrent, Vuze…
- Three degrees (Windows) – early beta p2p

Games

- Quake3 (all platforms)
- Xtris (Unix, Linux, xBSD)

Grid computing

- Globus toolkit (java based)

Monitoring/diagnostics

- Ping6, Traceroute6, Iperf, Wireshark, Microsoft Network monitor3.0

Management

- Apple's AirPort manager for 'Time Machine' backup system
  - Also provides IPv6 tunnel, routing and RADV

Available IPv6 Enabled Applications

Many were tested under 6NET.org Project

- Application Database: http://6net.iif.hu/ipv6_apps
  - Slightly out of date
- 6NET Deliverables discuss their use
  - Particularly those of WP5

IPv6 Portal (ipv6tf.org) – IPv6 enabled apps site

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


deploy
Enabling Applications for IPv6

Enabling application for IPv6

Most IPv4 applications can be IPv6 enabled

- Appropriate abstraction layers used

Providing 'Dual stack' IPv4 and IPv6 is best

- Run-time (preferable) or compile-time network mode (v6 and/or v4)

All widely used languages are IPv6-enabled

- E.g. C/C++, Java, Python, Perl
- Some languages make it particularly easy
  - E.g Java

Benefiting from IPv6 is a little more difficult

- Though most functionality is the similar to IPv4
- Add special functionality for IPv6 features

IPv4 and IPv6 APIs have largely converged

Effects on higher layers

Affects anything that reads/writes/stores/passes IP addresses

- Most IETF protocols have been updated for IPv6 compliance

Bigger IP header must be taken into account when computing max payload sizes

Packet lifetime no longer limited by IP layer (it never was, anyway!)

Address scoping for multicast

New DNS record type: AAAA

Advanced mobility

- Mobile IPv6, Network Mobility (NEMO)

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* argument
  - 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

- Use of brackets for literal addresses (RFC2732)
  - e.g.“[2001:D::1]”

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


deploy
Heterogeneous Environments

Precautions for Dual Stack

Avoid any explicit use of IP addresses

- Normally do Call by Name

Ensure that calls to network utilities are concentrated in one subroutine

Ensure that libraries and utilities used support both stacks

Do not request functions that would not exist in both stacks

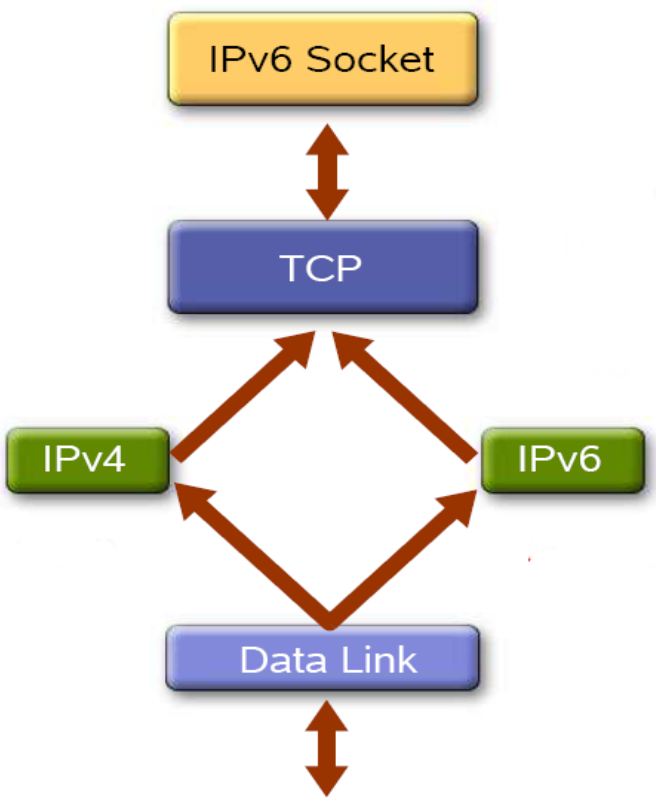
- E.g. IPsec, MIP, Neighbour Discovery may vary

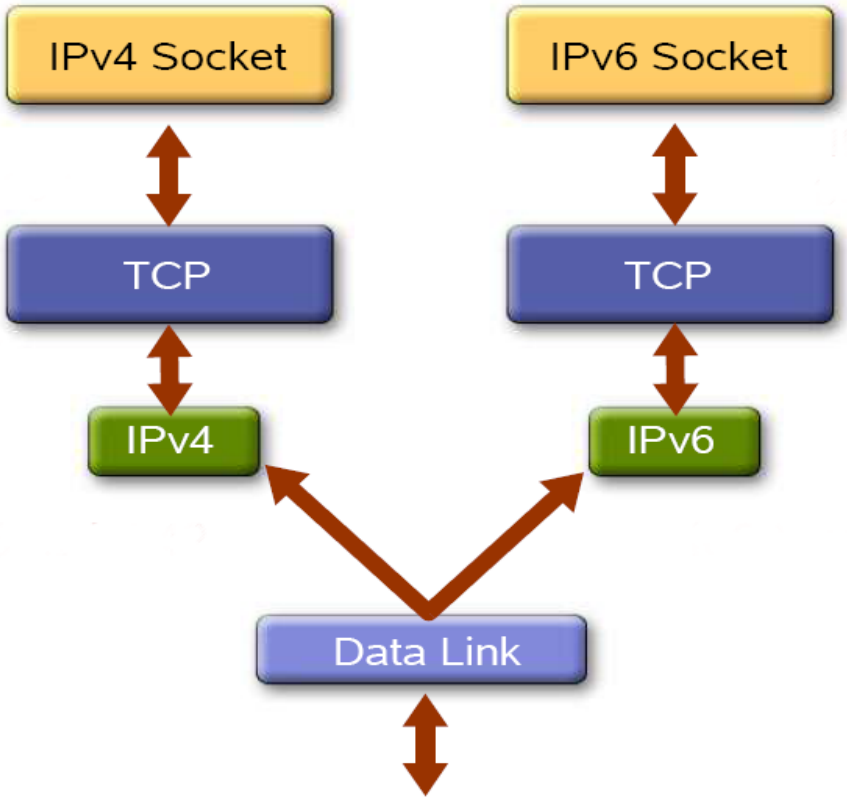
Dual stack configurations

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

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

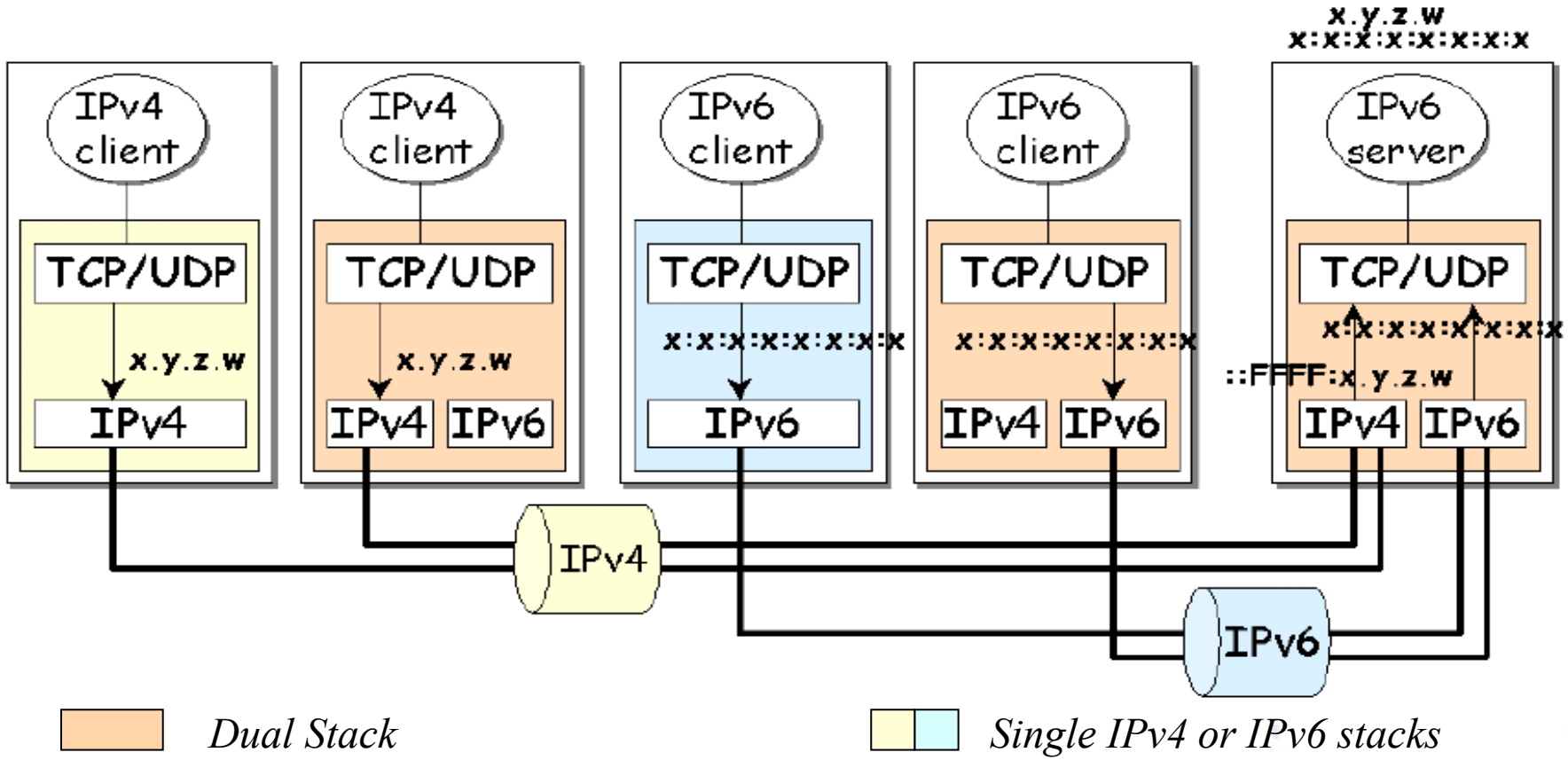
May require transition gateway

- As for example with IPv4 telephones accessing other IPv6 ones

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

Mapping IPv4 address in IPv6

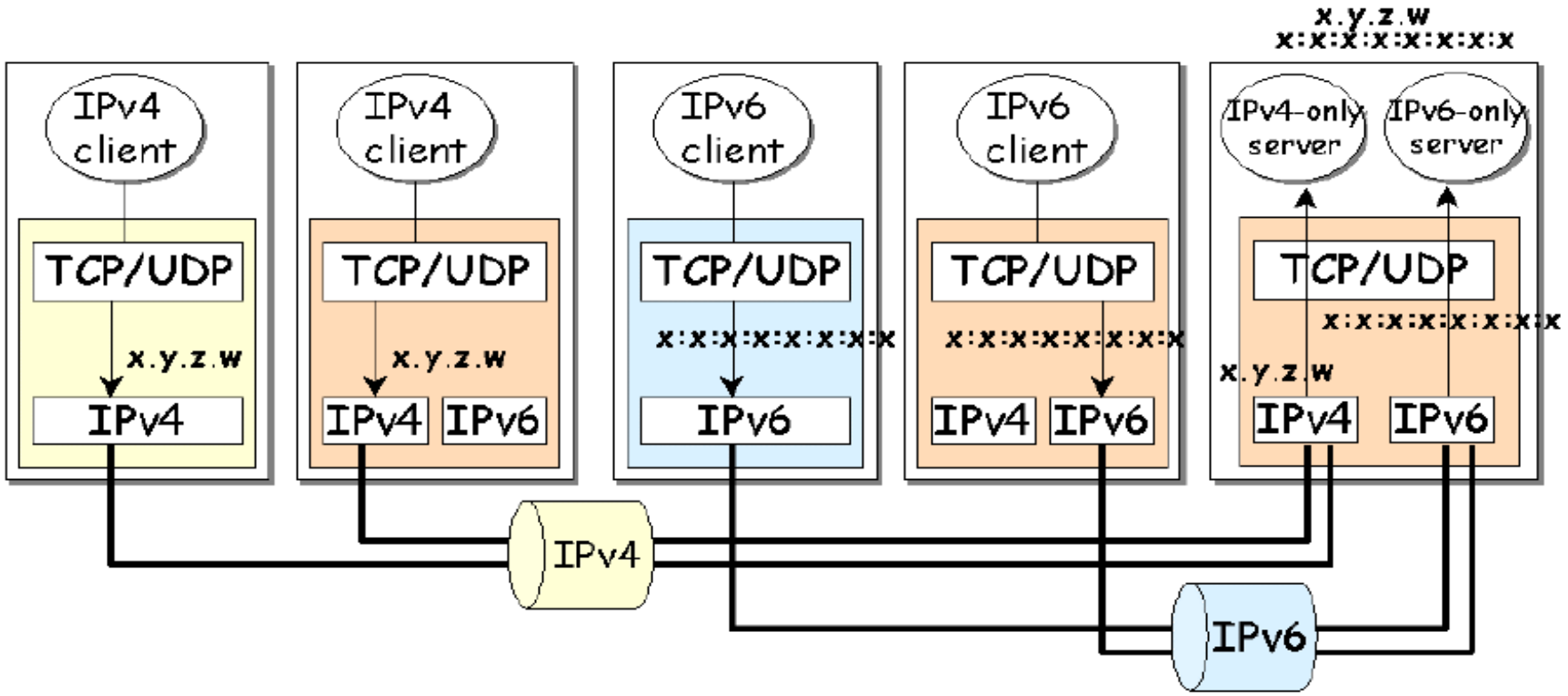
IPv6/IPv4 Clients connecting to an IPv6 server at dual stack node → 1 socket



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



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

New Applications

Simplified by writing apps using a high-level language

- E.g. JAVA seamlessly supports dual stack

Design the application in a protocol independent fashion

Ensure both protocols will be simultaneously operable

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

Re-architect code so that it provides

- Appropriate network abstraction layer

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


deploy
Voip/Conferencing for IPv6

Origins of packet based multimedia

1974

- Realtime Packet Voice demonstrated between USC/ISI and MIT/LL, using CVSD and Network Voice Protocol (NVP[RFC471]) on IPv5 (ST[RFC 1819])

1976

- First packetised speech over SATNET between Lincoln Labs and both NTA (Norway) and UCL (UK).

1991

- LBL's audio tool vat released for DARTnet use.

1992

- First IETF MBone audiocast (San Diego, CA)

1993

- Video Conference (VIC) tool released by LBL

1995

- Robust Audio Tool (RAT) released by UCL

1996

- RTP standardized (RFC 1889/1890)

1996

- H.323v1 published

1999

- SIP standardized (RFC 2543)

VoIP protocol overview

Session control/setup protocols

- Session Initiation Protocol (SIP) – IETF standard RFC3261
  - Widely used for VoIP and conferencing
  - IPv6 support in Proxies: open/SER, Asterisk, CUCM-7
    - Clients: Snom, Ekiga-3.2, linphone
- H.323 – ITU-T Standard
  - Widely used for conferencing
  - Supports IPv6 operation: Tandberg (2008), OpenH323
- Skinny Call Control Protocol (SCCP) – Cisco protocol
  - Used in Cisco VoIP systems, also support in Asterisk
  - Supported IPv6 operation on CUCM 7

Media transport protocols

- Realtime Transport Protocol (RTP) – IETF RFC3550 (in ITU-T H.225)
  - Dominant media transport protocol – used by all of above

UCL Media tools

http://mediatools.cs.ucl.ac.uk/

VIC – Video tool

- Lawrence Berkeley National Lab
  - Initial Funding: ARPA, DoE
  - First LBL release 2.0a (Oct 93)
- Taken over by UCL in 2000

RAT – Robust Audio tool

- University College London
  - Initial funding: MICE EU Project
  - First release: RAT-1 95

Common

- University College London
  - Initial funding: MECCANO EU PRoject
  - First release: common-1.0.0 (Nov 98)

RAT (Robust Audio Tool)

Motivation

- Move beyond existing tools: VAT(LBL), nevot (AT&T), vt(ISI)
- New features; Redundant Audio, Loss concealment schemes, sample-rate conversion, IPv6, Stereo, 3D audio, etc

Origins

- University College London
  - MICE (**M**ultimedia **I**nternational **C**onferencing for **E**urope) EU Project(s): '92-95
  - EU:{MERCI (95-97), MECCANO(98-00), COIAS(98-00)} HICID(97-00)
  - RAT EPSRC Project: '96-99
  - Relate (**Re**mote **L**anguage **Te**aching) BT Project : '94-97
- Credits
  - C.Perkins, O.Hodson, I.Kouvelas, V.Hardman, A.Sasse, M.Handley, S.Varakliotis, and many more

RAT Screenshot

RAT(v4) Current Architecture

Source code: C & Tcl/Tk

Two main processes

- Controller process parses arguments and spawns 2 processes
- Communication using MBUS over local multicast
- Built on UCL common library

Media Engine

- Auddev: Drivers to various audio hardware
  - Linux (new: ALSA1.0 & OSS), Win32, OSX, Solaris, BSD
- Packet reception/transmission and RTP de/packetisation
- Mixing, Redundancy support, Layering, Loss concealment schemes, IPv6, Stereo, 3D audio, sample-rate conversion
- Codecs: G.711, G.726, GSM, DVI, LPC, L16..

User interface

- Tcl/tk GUI to control media engine
- Others possible (Java one has been done)

RAT: IPv6

Included IPv6 support since 1998

- On Linux, FreeBSD, & Windows NT4

Minimal changes required as Common is IP-independent

Some changes:

- #includes header files for IPv6 (not necessary now - as they come with std networking #includes)
- Text handling of addresses

VIC (VideoConference tool)

Motivation

- Move beyond existing tools: e.g. ivs(INRIA), nv(Xerox)
- Increased range of codecs, networking options, GUI, packet loss tolerance

Origins

- Lawrence Berkeley National Lab
  - Funding: DoE, ARPA
  - Later University of California, Berkeley
    - NSF, DEC, SUN, SGI
    - MASH Project
- Credits
  - S.McCanne, V.Jacobson, E.Amir, and many more

VIC Screenshot

VIC Current Architecture

Source code: C/C++ with tcl/tk GUI

Single process

- Tcl/C++ components connected using tcl scripting
- Uses UCL common for MBUS and DES
- Integrated some MASH code into UCL vic
  - E.g. Packet-buf, layered codec support (PVH)

Subsystems

- RTP : Session handling
- NET : Network support (IPv6/4, ATM, etc)
- VIDEO : Grabber hardware drivers (new: WDM)
- RENDER : Video rendering/conversion
- CODEC : H.261, H.263, PVH, BVC, JPEG, CellB, NV

VIC: IPv6

Included IPv6 support since 1998 v2.8ucl4

- On Linux, Solaris, & Windows NT4

Quite a few changes required as it doesn't use UCL Common for networking (only MBUS)

IPv6 support initially from UCLA

- Modified by UCL later

UCL added IPv6 SSM support

GLOBAL EU Project

Global Linkage Over BroadbAnd Links

The GLOBAL project has set up a collaborative e-Infrastructure, called the Virtual Conference Center (VCC), which enables the regular realisation of virtual events.

The GLOBAL project objectives:

- Providing the "Virtual Conference Centre" Collaborative e-Infrastructure
- Realisation of Global Networking Sessions
- Disseminating the Results and Providing Sustainability
- Providing Services for Third Parties

GLOBAL Project

The project is supported by the European Commission's FRAMEWORK 7 PROGRAMME/THEME: Capacities/ Research Infrastructures.

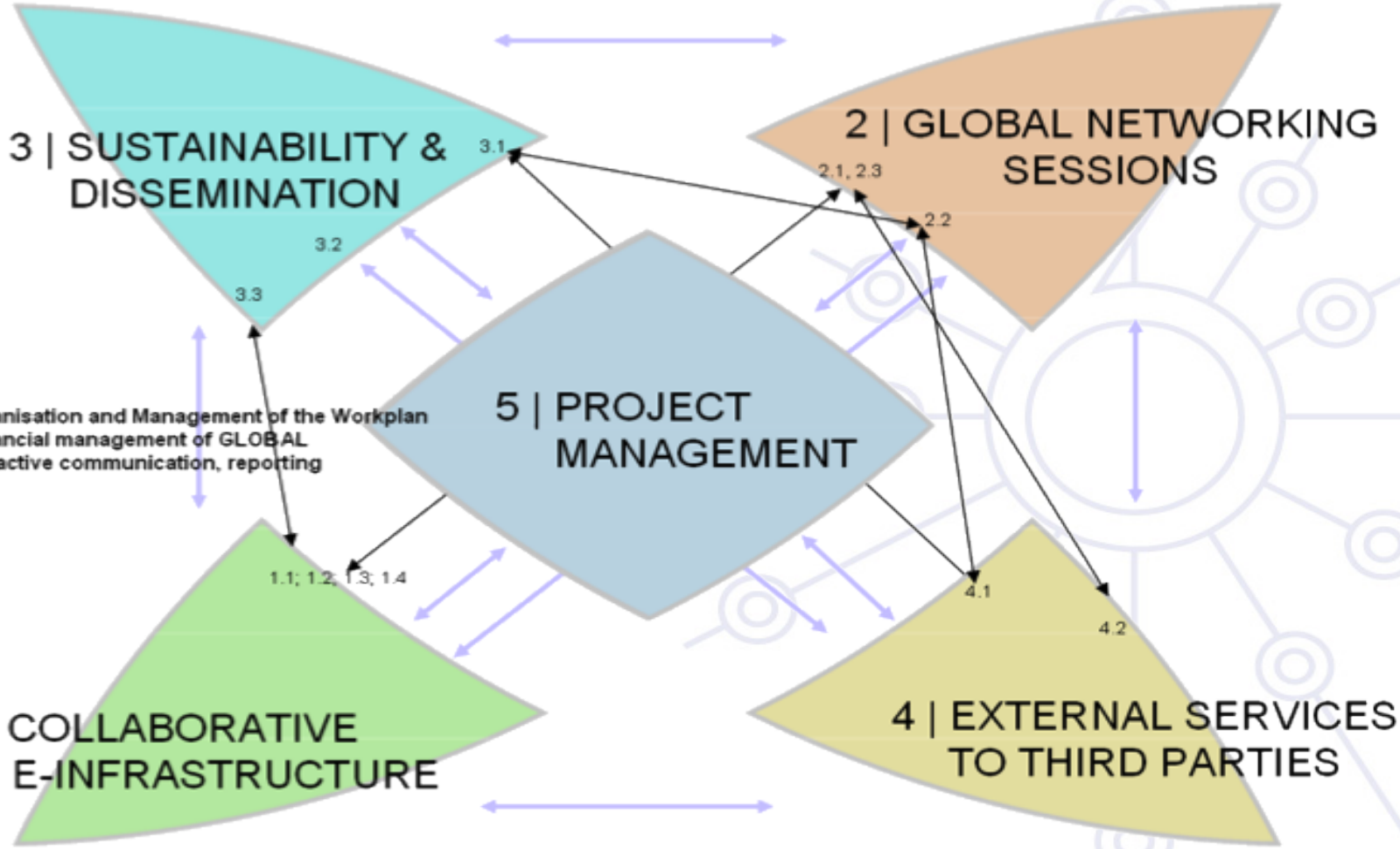
- Project dates: 2008-05-01 – 2010-10-31

Partners

- Zentrum für Soziale Innovation (ZSI), Austria
- U. Politécnica de Madrid (UPM), Spain
- Agora Systems S.A. (ASSA), Spain
- Cooperación Latino Americana de Redes Avanzadas (CLARA)
- University College London (UCL), United Kingdom
- Jozef Stefan Institute (JSI), Slovenia
- Ubuntunet (Ubuntunet), Malawi

Task 3.1.: Development of Project Promotion Material.
Task 3.2.: Virtual Corridor and continuous dissemination channels.
Task 3.3.: Website interfaces and content.

Task 2.1.: Technical Event Helpdesk.
Task 2.2.: Content-wise Organisation of Events.
Task 2.3.: Technical Operation and Management of Networking Sessions.



Task 5.1: Organisation and Management of the Workplan
Task 5.2.: Financial management of GLOBAL
Task 5.3.: Proactive communication, reporting

Task 1.1.: The VCC Website core infrastructure.
Task 1.2.: Virtual Auditorium core infrastructure.
Task 1.3.: Inter-working infrastructure.
Task 1.4.: Narrowband Access to virtual events.

Task 4.1.: Identification of and Liaison with Potential Users and Developers.
Task 4.2.: Logistical support to events organised by third parties.

UCL: Leads WP4 & T1.3

WP4: Liaison with 3rd Party Users and Systems

- Support Organisers of 3rd Party Events
  - Projects, Organisations: TERENA, DANTE, GEANT..
- Support Users of 3rd Party Events
  - CLARA, UBUNTUNET, Silk NRENs (Central Asia)...
- Support Developers of 3rd Party Systems
  - Provide specific Gateway API (GAPI) to ISABEL

Task 1.3: Interworking infrastructures

- Investigation of gateways with other systems
  - SIP, H.323, SCCP

CLARA: leader of Task 1.4

T1.4: Narrowband Access to virtual events

- The goal is to allow access to sites that, for lack of resources, can't normally connect to an Isabel session.
- It's divided in two parts:
- A client that allow access from ADSL lines.
- Access from mobile devices.
- Easy to use interface / no technical personnel required.

Conferencing systems

ISABEL (GLOBAL project)

- Has supported IPv6 for a few years

AccessGrid

- Has IPv6 media support for a few years
  - Due to UCL VIC and RAT
- UCL implemented IPv6-IPv4 gateway in 6NET project

Commercial

- Cisco/Tandberg and others

Grid

Open Grid Forum IPv6-Working group

- Formed in 2003 during 6net project
  - WG Chairs: Myself and Brian Carpenter (IBM/University of Auckland)
- Issued two Standards documents:
  - T. Chown, S. Jiang, P. O'Hanlon, J. Bound, *Guidelines for IP version independence in GGF specifications,GFD.40, Jan '05*
  - Rute Sofia, *Survey of IPv4 Dependencies in Global Grid Forum Specifications,GFD.41, Nov '04*

Globus Toolkit – Initial IPv6 support in GTv3.2

- UCL IPv6-enabled GT4 in 2005 under 6net project
  - http://www.cs.ucl.ac.uk/staff/sjiang/webpage/How-to-IPv6-in-GT4.htm
  - http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=2232

2009: 3TERA Cloud IPv6 enabled

Smart [power] Grid

Smart Grids are being developed globally

- Make grid more efficient – potential large cost savings
  - US estimated $56-112 Billion saving in 20 years
- Earliest examples
  - 2005: Italy - Telegestore project €2.1B – annual savings €500M per year!

2009: US Smart Grid Initiative - $8.1 Billion

- 40 Million smart meters…
- http://www.nist.gov/smartgrid
- Smartgrid BoF at IETF76 in Japan, Nov 2009
- ftp://ftpeng.cisco.com/fred/IETF-SG/
- Happening fast – standards to be ready by end 2010

Large number of addresses => Need for IPv6


deploy
Questions

Conclusion

Many existing applications are available in IPv6

Porting applications to IPv6 is straightforward

- Provided certain guidelines are followed

Heterogeneous environments provide the most challenges