



IPv6 Applications

Case study: Conferencing

Piers O'Hanlon



Copy ...Rights

- ***This slide set is the ownership of the 6DISS 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 6DISS courtesy***
- ***PDF files are available from www.6diss.org***
- ***Looking for a contact ?***
 - ***Mail to : martin.potts@martel-consulting.ch***
 - ***Or bernard.tuy@renater.fr***



Overview

- Origins of packet multimedia
- Genesis of media tools
- Current status
- IPv6 conversion



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)
- 1996
 - RTP standardized (RFC 1889/1890)
- 1996
 - H.323v1 published
- 1999
 - SIP standardized (RFC 2543)



Genesis of tools

- **VIC:**
 - Lawrence Berkeley National Lab
 - Initial Funding: ARPA, DoE
 - First LBL release 2.0a (Oct 93)
 - Earlier collaborator releases to UCL
- **RAT :**
 - University College London
 - Initial funding: MICE EU Project
 - First release: RAT-1 95
- **Common**
 - University College London
 - Initial funding: MECCANO EU PProject
 - First release: common-1.0.0 (Nov 98)



Other tools

- Integrated systems
 - Ekiga - point to point - (multipoint using MCU)
 - Skype - proprietary encrypted P2P
 - Mstar/Marratech Java system from Lulea University
 - ISABEL from Universidad Politecnica Madrid
- Session directories
 - SD from LBL
 - SDR from UCL – using SDP (RFC2327)
- Shared Whiteboards
 - WB from LBL using SRM reliable multicast
 - WBD from Loughborough and then UCL
 - DBL (Digital Lecture Board) from U of Mannheim
- Shared Text
 - NTE (Networked Text) tool from UCL



UCL Common Library architecture

- Provides a library of functions for media communication
 - Source code: C (in CVS from start)
 - NETUDP - A UDP interface for IPv4 and IPv6.
 - MBUS - Message bus for local coordination (RFC 3259)
 - Used in RAT extensively, in VIC less so. Other apps (Freevo)
 - Crypto/Authentication
 - DES, AES – Data & Advanced Encryption Standards
 - HMAC - Cryptographic message authentication.
 - MD5 - Message Digest-5 algorithm.
 - RTP - Real-Time Transport Protocol (RFC 3550)
 - SAP - Session announcement protocol (RFC 2974)
 - SDP - Session Description Protocol (RFC 2327)
 - Debug - Functions for outputting diagnostics.
 - Memory - Memory allocation and debugging functions.
 - Utility functions – Base64, random numbers, binary trees



UCL Common: IPv6 code

- Configuration
 - ./configure --enable-ipv6
 - uclconf.h, config_unix.h, config_win32.h
- Source files
 - UDP network connections
 - net_udp.c
 - RTP handling
 - rtp.c



Data structures

- UDP socket structure

```
struct _socket_udp {  
    int         mode; /* IPv4 or IPv6 */  
    char        *addr;  
    uint16_t    rx_port;  
    uint16_t    tx_port;  
    ttl_t       ttl;  
    fd_t        fd;  
    struct in_addr  addr4;  
#ifdef HAVE_IPv6  
    struct in6_addr addr6;  
#endif /* HAVE_IPv6 */  
    struct in_addr  iface_addr;  
};
```



IP interface initialisation

- IP-independent entry point to interface initialisation
 - `socket_udp *udp_init_if(const char *addr, const char *iface, uint16_t rx_port, uint16_t tx_port, int ttl) {`
`socket_udp *res;`
`if (strchr(addr, ':') == NULL) {`
`res = udp_init4(addr, iface, rx_port, tx_port, ttl);`
`} else {`
`res = udp_init6(addr, iface, rx_port, tx_port, ttl);`
`} return res; }`
- IPv6 specific interface initialisation
 - `static socket_udp *udp_init6(const char *addr, const char *iface, uint16_t rx_port, uint16_t tx_port, int ttl)`



UDP packet sending

- IP-independent entry point to udp send

```
– int udp_send(socket_udp *s, char *buffer, int buflen) {  
    switch (s->mode) {  
        case IPv4 : return udp_send4(s, buffer, buflen);  
        case IPv6 : return udp_send6(s, buffer, buflen);  
        default  : abort();    }    return -1; }
```

- IPv6 specific version

```
– static int udp_send6(socket_udp *s, char *buffer, int buflen) {  
    struct sockaddr_in6  s_in;  
    memset((char *)&s_in, 0, sizeof(s_in));  
    s_in.sin6_family = AF_INET6;  
    s_in.sin6_addr  = s->addr6;  
    s_in.sin6_port  = htons(s->tx_port);  
    s_in.sin6_len   = sizeof(s_in);  
    return sendto(s->fd, buffer, buflen, 0, (struct sockaddr *) &s_in, sizeof(s_in)); }
```



Conversion functions

- A couple of [standard] conversions not available on all platforms (less of a problem these days)
- Convert from presentation format (text) to network format (e.g. struct in6_addr/in_addr)
 - int **inet_pton**(int af, const char * src, void * dst);
- Reverse function
 - const char ***inet_ntop**(int af, const void * src, char * dst, socklen_t size);



Other functions

- Cleanup and utility functions
 - static void ***udp_exit***(socket_udp *s)
 - IPv6 specific function
 - static void ***udp_exit6***(socket_udp *s)
 - static int ***udp_addr_valid6***(const char *dst)



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:{MERCY (95-97), MECCANO(98-00), COIAS(98-00)} HICID(97-00)
 - RAT EPSRC Project: '96-99
 - Relate (**R**emote **L**anguage **T**eaching) 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

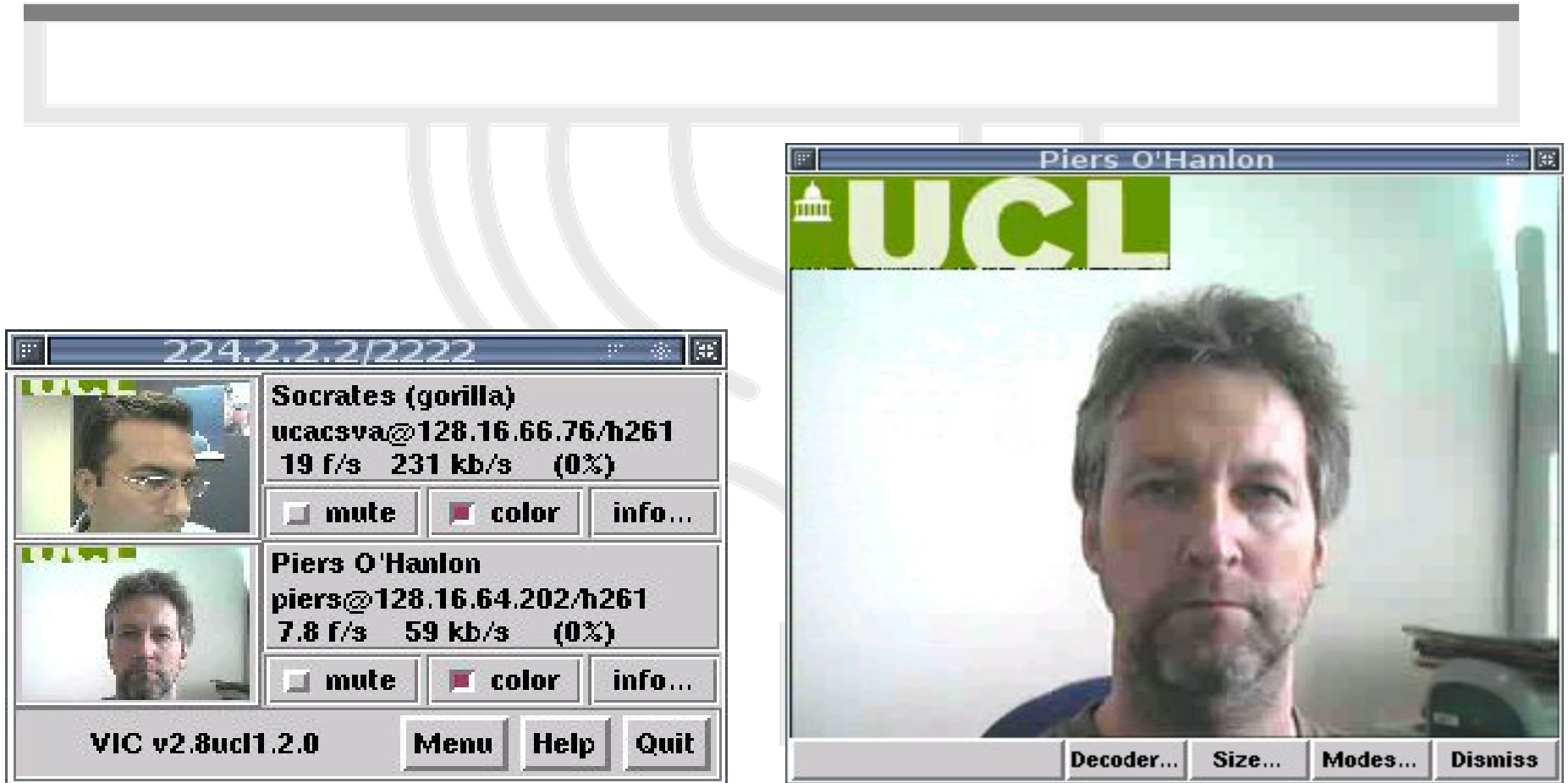


VIC (VideoConference tool)

- 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



VIC: IPv6 code

- Configuration
 - `./configure --enable-ipv6`
 - `config.h`
- Source files
 - UDP network connections
 - `net/net-ipv6.cpp`
 - netdb functions for IPv6 addresses
 - `net/inet6.c`
 - RTP handling
 - `rtp/session.cpp` __IPv6 changed srcid computation
 - `rtp/source.cpp`:* IPV6Q (used to look up on addr)



GUI modifications

– GUI

- tcl/cf-network.tcl: # Auto detect IPv6 addresses
- tcl/cf-network.tcl: # Check for ipv6 as well as ip6 as a netType
- tcl/cf-network.tcl: if { \$IPaddrFamily == "ipv6" } {
- tcl/ui-resource.tcl: # Flow Label info for IPV6
- tcl/ui-resource.tcl: # Interface Index to bind to in IPv6



Current status

- Tools development driven by SUMOVER project
- SUMOVER Project
 - JISC funded project for 2 years (Started: Aug 05)
 - Joint Information Systems Committee (JISC)
 - Directly funded by UK further and higher education funding councils for development of the infrastructure and activities
 - Resources: ~1.5 people for 2 years
- See:
 - <http://mediatools.cs.ucl.ac.uk/>



Demo...

