

Project Number:	IST-2001-32603
Project Title:	6NET
CEC Deliverable Number:	32603/IBM/DS/D5.1/A1
Contractual Date of Delivery to the CEC:	30 April 2002
Actual Date of Delivery to the CEC:	27 May 2002
Title of Deliverable:	Specification of IPv6 applications to be developed within the project
Work package contributing to Deliverable:	WP5
Type of Deliverable*:	S
Deliverable Security Class**:	PU
Editors:	IBM France
Contributors:	Workpackage 5

* Type: D - Demonstrator, O – Other, S – Specification, P - Prototype, R - Report,

** Security Class: PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Keywords:

IPv6, Applications, Specifications, Real-time video-conferencing and media streaming, Online games, E-business, Edge services.

Abstract:

This deliverable lists and briefly describes the applications identified as potential candidates to run on 6Net's IPv6 network. In future releases of this document, detailed specifications will be provided for the applications eventually selected by the Consortium as key 6Net applications. New applications brought by partners from Newly Associated States who will join an extension of the project will also be added.

The document is contains two main sections: the first section contains texts that briefly describe the domain and functions of the applications; the second one contains for each application a 'summary sheet' that details its technical environment.

Both sections are structured in four parts, mapped no the four activities of Workpackage 5, namely:

- Real-time video-conferencing and media streaming
- Online games
- E-business
- Edge services

The first part that contains a large number of applications has been divided in several sub-categories.

Table of Contents

Introduction	5
1. Description of the applications.....	5
1.1. Real-time video-conferencing and media streaming.....	5
1.1.1. Streaming applications - Audio and Video	5
Storage and Retrieval	5
– Video Over IP - VIP.....	5
– Video distribution - MPEG4IP.....	6
– Real-time conferencing tools – GnomeMeeting	6
– Music distribution – TUR on IPv6.....	7
– MultiMedia Conference Recorder – MMCR	7
– Video Lan.....	8
– Multicast Radio	8
– Unified Messaging System – 6UMS.....	9
– Kasenna MediaBase XMP streaming server.....	9
– FreeAMP	9
Conversational.....	10
– Open H323	10
– Robust Audio Tool – RAT.....	11
– Video Conference Tool – VIC	12
– Network Text Editor - NTE	12
– Whiteboard – WBD.....	13
– High-quality Audio Tools - HAT.....	13
– ISABEL CSCW.....	14
– Digital Video Transport System - DVTS	14
1.1.2. Protocols.....	15
– Quadapt	15
– SIP server	15
Bonephone-SIP client.....	15
1.1.3. Session.....	18
Monitoring.....	18
– RTP Quality Matrix – RQM.....	18
Handling.....	18
– Multicast Streaming Tools - MUST.....	18
– Session Directory Tool – SDR.....	19
– Secure Conference Store – SCS.....	19
– SDP Parser Applet – SPAR.....	20
1.2. On-line Games.....	21
– Quake	21
– Experimental Gaming Platform - EGP.....	21
– XPilot	22
– MUD gaming environment	23
1.3. E-business solutions.....	23
– IBM Websphere Portal Technology.....	23
– GLOBUS 2.0.....	25
– Agent framework - SoFAR, SLITE	25
– Hypermedia link services	25

–	FunnelWeb	26
–	Transcoding Active Gateway – TAG.....	26
–	TZI Stargate.....	27
–	LightWeight Directory Access Protocol - OpenLDAP	28
–	Public Key Infrastructure – PKI (University of Murcia)	28
1.4.	Edge-services for IPV6	30
–	Edge Server Proxy.....	30
–	Contents Delivery Networking - CDN.....	34
2.	Summary sheets.....	35
1.	Video over IP - VIP.....	36
2.	Video distribution - MPEG4IP.....	37
3.	Real-time conferencing tools - GnomeMeeting.....	38
4.	Music distribution - TUR on IPv6.....	39
5.	Multimedia Conference Recorder - MMCR	40
6.	Video Lan.....	41
7.	Multicast Radio	42
8.	Unified messaging system - 6UMS.....	43
9.	Kasenna Mediabase XMP streaming server.....	44
10.	FreeAMP	45
11.	OpenH323	46
12.	Robust Audio Tool - RAT.....	47
13.	Video Conference Tool - VIC.....	48
14.	Network Text Editor - NTE	49
15.	Whiteboard - WBD	50
16.	High-quality Audio Tools - HAT.....	51
17.	ISABEL CSCW.....	52
18.	Digital Video Transport System - DVTS	53
19.	Quadapt	54
20.	Bonephone – SIP client.....	55
21.	RTP Quality Matrix - RQM	56
22.	Multicast streaming tools - MUST.....	57
23.	Session Directory Tool - SDR.....	58
24.	Secure Conference Store - SCS.....	59
25.	SDP Parser Applet - SPAR	60
26.	Quake	61
27.	Experimental Gaming Platform - EGP.....	62
28.	XPilot	63
29.	MUD gaming environment	64
30.	IBM Websphere Portal Technology.....	65
31.	GLOBUS 2.0.....	66
32.	Agent Framework – SoFAR, SLITE.....	67
33.	Hypermedia Link Services	68
34.	Funnel Web	69
35.	Transcoding Active Gateway - TAG	70
36.	TZI Stargate.....	71
37.	LightWeight Directory Access Protocol - OpenLDAP	72
38.	Public key infrastructure - PKI.....	73
39.	Edge Server Proxy.....	74
40.	Contents Delivery Networking - CDN.....	75

Introduction

This deliverable lists and briefly describes the applications identified as potential candidates to run on 6Net's IPv6 network. In future releases of this document, detailed specifications will be provided for the applications eventually selected by the Consortium as key 6Net applications. New applications brought by partners from Newly Associated States who will join an extension of the project will also be added.

The document is contains two main sections: the first section contains texts that briefly describe the domain and functions of the applications; the second one contains for each application a 'summary sheet' that details its technical environment.

Both sections are structured in four parts, mapped no the four activities of Workpackage 5, namely:

- Real-time video-conferencing and media streaming
- Online games
- E-business
- Edge services

The first part that contains a large number of applications has been divided in several sub-categories.

1. Description of the applications

1.1. Real-time video-conferencing and media streaming

1.1.1. Streaming applications - Audio and Video

Storage and Retrieval

– Video Over IP - VIP

VIP aims to set up an integrated chain solution for video-over-IP services, making it possible to provide full-screen, high-quality video (MPEG2 or TV quality) that is fully scaleable over the Internet. This solution will cover the entire chain from video production to delivery of video over the Internet.

The following components are involved:

- Production of digital video material;
- Production of content (video material plus metadata) and the supervision of this production process; digitalization of video and other material in various formats; addition of extra specifications (metadata) to the video material for the purpose of data interchange;
- Storage of the content in a video store taking care of indexing and security;
- Providing search, retrieval and browsing facilities with an attractive user interface;
- Distribution of video over the IP network; implementation of a network architecture; configuration of the IP network;
- Content-based billing & accounting.

VIP is an integration project, where existing methods, techniques and algorithms are incorporated into a new chain for digital content production and deployment. This solution consists of an architecture

32603	Deliverable D5.1 – v2	
-------	-----------------------	--

for the production of content and an architecture for content deployment. In addition, the IP-network environment (Internet) naturally plays an important role. With the availability of large amounts of bandwidth, we will be able to provide end users with high-quality video.

More information can be found at:

<http://www.telin.nl/CE/VIPvideo/index.html>

<https://doc.telin.nl/dscgi/ds.py/Get/File-10636/isma2000-1.1.pdf>

– **Video distribution - MPEG4IP**

MPEG4IP provides an end-to-end system to explore MPEG-4 multimedia. The package includes many existing open source packages and the "glue" to integrate them together. This is a tool for streaming video and audio that is standards-oriented and free from proprietary protocols and extensions.

Provided are an MPEG-4 AAC audio encoder, an MP3 encoder, two MPEG-4 video encoders, an MP4 file creator and hinter, an IETF standards-based streaming server, and an MPEG-4 player that can both stream and playback from local file.

Development is focused on the Linux platform, and has been ported to Windows, Solaris, FreeBSD, BSD/OS and Mac OS X, but it should be relatively straight-forward to use on other platforms. Many of the included packages are multi-platform already.

Three tools are our main interest:

– *Mp4live* :

This program is designed to make it easy to create MP4 files or transmit live audio/video streams over the network.

– *Mp4player* :

This program is used for playback of multimedia content, either distributed live or from a file.

– *Darwin Streaming Server* :

The Darwin Streaming Server is streaming server technology which allows you to send streaming QuickTime and MPEG4 data to clients across the Internet using the industry standard RTP and RTSP protocols.

More information can be found at:

<http://mpeg4ip.sourceforge.net/index.php>

<http://developer.apple.com/darwin/projects/streaming/index.html>

– **Real-time conferencing tools – GnomeMeeting**

GnomeMeeting is an open source H323 application for Linux based on the OpenH323 platform. Real-time conferencing applications are becoming very popular but few are available for non-Windows OSs.

GnomeMeeting is a Linux based H323 compatible application that claims interoperability with Windows based H323 clients, thus opening the possibility for a universal conferencing platform.

Benefits from IPv6 support will be QoS features and security. Features include:

- GUI interface for popular Linux window managers.
- Gatekeeper and directory support.
- Audio – only mode.
- Automatic device detection

– Music distribution – TUR on IPv6

This application does streaming of MP3 and possibly other formats. At present this is done 24x7 freely available to all the world over IPv4 (see <http://www.turmusic.no/>). This site is named "Trondheim Underground Radio ("TUR for short).

Content will be made available for IPv6, first unicast and later multicast. We expect (but can not guarantee) to distribute this on the same terms throughout the 6NET project period. The setup will be documented, and any software developed will be freely available, so that others can use the same solution

Currently the application offers 48 kbps and 128 kbps IPv4 unicast stream, using the shoutcast/icecast protocol (TCP). There are a number of available MP3 players to be used for decoding such streams, and some of them support both unicast and multicast. TUR has been experimenting with multicast using liveCaster (<http://www.live.com/liveCaster/>). We would like to base the primary service on software developed in the Icecast project (<http://www.icecast.org/>), which has been ported to IPv6. Some of the clients supporting IPv6 will be tested. Icecast has no native support for multicast, but we would like to support multicast streaming of TUR as well as unicast, both on IPv6 and on IPv4.

– MultiMedia Conference Recorder – MMCR

The Multicast Multimedia Conference Recorder is a system capable of recording and playing back multimedia data sent over the Multicast Backbone (Mbone). MMCR is a system specifically designed for recording and playing back multicast multimedia conferences over the Mbone. It has a client - server architecture and consists of the client User Interface and the server (which incorporates the playback, recording and browsing mechanisms); logical component independence simplifies development and component replication.

MMCR exists in two forms; the java application based system, and a FunnelWeb proxylet implementation which is currently in a beta form. MMCR was developed at UCL.

All server components have access to the database archive (see Fig. 24) to store/retrieve recordings and information about them. Much research has considered ways of providing efficient storage/access mechanisms for Video On Demand (VOD) systems, which require high bandwidth delivery. However, the simple disk model used here is adequate considering the current bandwidth limitations on the Mbone and a Redundant Array of Independent Disks (RAID) can be integrated in the system as an enhancement, if necessary.

– The Server:

The server acts as the single point of contact for recording, browsing and playback. Most of the existing implementations have a similar architecture. They consist of independent components; a server manager, the player, the recorder and the browser; some of these components may be missing in specific implementations.

The server manager controls the whole service; it handles the establishment of connections with the clients. It has a separate, independent interface for each task and more interfaces can be added when required (e.g. an editing interface). Depending upon the type of service requested by the client, the server manager starts one of the recording, browsing or playback mechanisms. Once the mechanism required has started, the remote client communicates directly with that mechanism. Each mechanism has its own text-based control protocol.

– The Recorder :

To record the media streams the recorder need not be an active part of the conference; it 'listens' to the specified multicast groups and collects the data. Each stream is stored separately. In the case of RTP media, the RTCP messages transmitted are stored along with the data packets.

Information about each recorded media (e.g. type, name) and each source (e.g. data location) is saved in header files. This information is either provided by the user or it is included in the Session Description of the conference. It is then possible to catalogue and index the descriptions for subsequent retrieval. This indexing may be in text form; some current research projects

[sahouria] are attempting to use non-textual forms of indexing to allow more sophisticated retrieval.

- The Browser

A listing of conferences a server has stored in its archive can be obtained through the browsing mechanism. A title keyword search facility is also available to help identify titles of interest.

Further details about a particular conference can also be obtained to assist a user in deciding which conference to play back. These details include the conference's title and description and the media that constitute the session. Additional information on each media includes the data type (i.e. RTP, wb etc) and the names of the users (where available) to help users select only the required data streams.

- The Player :

The real advantage of storing data on a per source basis is that users can playback only the streams they are actually interested in - ignoring the rest. This allows utilisation of all kinds of networks as users with bandwidth limitations may choose to play a subset of the available streams (e.g. just audio that requires much less bandwidth than video).

The player schedules real-time packet transmission based on the timestamp in the index entry. RTP compliant media provide additional information in the RTP header that can be used for providing smoother playback. Other media (e.g. shared workspace) packets are sent on the network based on their received timestamp (i.e. with the same inter-packet gap as they originally arrived).

The different media characteristics affect the fast forward and rewind operations. Audio and video are continuous media and therefore moving to a random point in the stream simply involves skipping intermediate parts and restarting at the new position. Additionally, the RTP header of the packets must be modified to maintain the continuity in timestamps and sequence numbers. For non-continuous media, such as shared workspace (wb/nte) fast-forward should involve the transmission of intermediate parts so that the data set is complete.

MMCR has not been tested with IPv6 though work is ongoing on an IPv6 version

- **Video Lan**

VideoLAN is an open source application that provides unicast and multicast a media streams from a variety of media sources (See: <http://www.videolan.org/> for details). Video Lan can source from a hard drive, a DVD player, a satellite TV card or an MPEG2 compression card and can create streams with data rates of up to 6-9Mbit/s for DVD, less for MPEG-1. It may be good multicast and QoS demonstrator.

- **Multicast Radio**

Cradio is an MP3 jukebox, home grown and already supporting IPv6. An alternative package, Icecast, is also available with IPv6 support. Here we propose Cradio, but we could run Icecast also with similar MP3 source and user community.

The application performs Web-based MP3 track selection and queuing and multicast operation. It is thus a good test of multicast IPv6 network. The Server can re-reference location of MP3 files via HTTP. Its main modules are:

- Server Side Multicast IPv6 MP3 Provider
- Web Based MP3 Selection Facility
- Client Side Multicast IPv6 Cradio Player
- Registration of Playlists

The application uses existing MP3 player, with mpg123 being favoured.

– **Unified Messaging System – 6UMS**

6UMS is an IPv6-enabled unified messaging system that allows peer-to-peer communication between users using a variety of media. It includes messaging using text, audio, images and video providing the following functions:

Includes

- location awareness
- user context and preferences
- intrusiveness consideration

University of Southampton has SMS relay tools available. During the project lifetime, we expect to communicate with advanced GPRS and 3G devices. Primary focus is WLAN PDA devices.

Major benefits lie in addressability and security.

– **Kasenna MediaBase XMP streaming server**

Kasenna MediaBase XMP is a system for management, distribution and streaming of video and audio assets encoded as MPEG-1, MPEG-2 and MPEG-4 video or MP3 audio. The system supports various MPEG encoders that can act as sources for live streams, and are redistributed by MediaBase XMP handles either multicast streams, unicast streams or both. Stored assets can be scheduled to multicast or made available on-demand. Each asset can have metadata associated with it, and it is possible to create multiformat assets and to define clips and sequences from the assets. MediaBase XMP also includes a content distribution module, for use with several servers, but it is not the intention to set up more than a single server.

– **FreeAMP**

FreeAmp is an extensible, cross-platform audio player. It features an optimized version of the GPLed Xing MPEG decoder, which makes it one of the fastest and best sounding players available. FreeAmp provides a number of the most common features users have come to expect in a clean, easy to use interface. Some of these features are:

- Plays all MPEG 1, MPEG 2, and MPEG 2.5 encoded files.
- Support for Xing's Variable Bitrate Encoding Technology.
- Plays Ogg/Vorbis files
- Plays Audio CD-ROMs, with CD-ROM lookup support via MusicBrainz (an included library).
- Supports Relatable TRM acoustic fingerprinting for playlist and metadata lookup of your MP3/Vorbis collection.
- Supports track lookups at Bitzi and MusicBrainz (web sites for free MP3 downloads).
- Play songs over the Internet through HTTP unicast streaming (ShoutCast), or RTP multicast streaming (Obsequiem).
- Supports Icecast and SHOUTCast style title streaming.
- Supports reading and writing ID3v1 and ID3v2 tags.
- Save ShoutCast and Icecast streams locally to your computer for offline listening...
- A powerful music browser and playlist editor. The playlist view supports customizable headers as well.
- A built in download manager which supports downloading files from sites using the RMP (RealJukebox) download process.
- Flexible theme interface.
- Complete help files.
- A new Watch this directory feature that periodically checks a given folder to see if new tracks need to be imported into MyMusic.
- Support for decoding MP3/Vorbis files to WAV files (using a WAVout PMO).

Conversational

– Open H323

OpenH323 platform is the result of OpenH323 project, which started in September 1998 by Equivalence Pty Ltd, a private company based in Australia (<http://www.openh323.org>) and has as target the development of an open source H.323 protocol stack.

OpenH323 platform is an open source platform, which contains both clients and server, which can be used for H.323 videoconference. The clients that are available include: a command line H.323 videoconference application and a GUI H.323 videoconference application. The servers that are available include: a H.323 MCU (Multipoint Control Unit), a H.323 Gatekeeper and a H.323 to PSTN Gateway. In addition, many tools based on the Openh323 library have been developed independently, like ISDN to H323 Gateways, Call Generator, and GnomeMeeting. GnomeMeeting is a graphical H323 client for Linux and BSD. It is the first free H323 videoconference tool with a powerful GUI interface available under GNU/Linux.

The porting of H.323 to IPv6 will enable the use of H.323 over native IPv6. In addition the H.323 will benefit from the advantages that IPv6 can offer to real time applications (like QoS support, security support, etc). OpenH323 platform is extensively tested on Linux and Win32 platforms. In addition it has been compiled on Solaris, FreeBSD, and BeOS. No special hardware is required to use OpenH323 except of a sound card for audio communication and camera (connected to the computer) for visual communication.

The network demands of H.323 videoconference in terms of bandwidth are the following:

- For endpoints (H.323 clients): In most of the cases 768Kbps is enough (in our knowledge the maximum bandwidth that the available H.323 endpoints can consume is 1.5Mbps).
- For MCUs (H.323 server): In most of the cases $n*768\text{Kbps}$ is enough (were n is the number of the endpoints which are connected in the MCU).

OpenH323 platform is using the following protocols:

- H.323 for videoconference
- G.711, GSM MS-GSM and LPC-10 for audio encoding
- G.723.1, G.728 and G.729 for audio encoding (with the use of appropriate hardware)
- H.261 for video encoding
- H.235 Annex D support for Gatekeeper access
- The RTP/RTCP protocol is used for the transmission of data.

The current version of OpenH323 platform does not support IPv6 networks. As the main developers of OpenH323 platform mentioned: “the way the system was designed was to make the support of different networks as simple as possible”. As we have described, OpenH323 platform consists of a number of modules (endpoints, MCU, gatekeeper, etc). Based on the available resources (resources of CTI and resources of other WP5 partners, which may be interested to participate in OpenH323 porting to IPv6) some (one or more) of the OpenH323 modules will be ported to IPv6 networks.

The ported OpenH323 modules will be tested among the participants of WP5 in order to validate their proper operation. Based on the OpenH323 modules (endpoints, MCU, gatekeeper, etc) which are going to be ported in IPv6 networks, different deployment scenarios can be used like: point to point connection between two H.323 endpoints or multipoint connection among a number of H.323 endpoints with the use of H.323 MCU.

H.323 videoconference has already a big enough user community. Many users groups are using H.323 videoconferencing in order to realize virtual meetings. For example the members of GRNet - VNOC

(Virtual Network Operating Center) in Greece, are using H.323 videoconference in order to arrange scheduled virtual meetings.

The OpenH323 project is based on a central library (called OpenH323 library), where most of the H.323 functionality is implemented. In addition OpenH323 project is based on the PWLib library, a portable (Windows/Unix) library that contains portable classes for I/O, multithreading, GUI, Internet protocols etc. The OpenH323 library is uses the PWLib library extensively.

The applications developed by the project (GUI and command line H.323 clients, a conference server, answering machine, an H.323 gatekeeper and PSTN gateway) are quite simple, because they rely heavily on the basic OpenH323 library.

Our goal is to port the OpenH323 library to IPv6. Porting the central library means that the applications based on the library can relatively easy be ported to IPv6. We have made some modifications to the PWLib and OpenH323 libraries in order to make them IPv6-enabled. Our involvement with this task until now has showed us that porting the libraries is a goal that can be achieved, although much attention has to be paid to details in the source code.

The code is written in C++ and structured in classes. Most of the classes that encapsulate basic transport capabilities seem to be protocol independent, and it comes down to the classes that encapsulate the socket API calls. These calls are IPv4 dependent and need modification, either by making them IPv6 dependent, or by making them protocol independent. The basic socket class is called PIPSocket and is part of the PWLib library. The PIPSocket class inherits a more general socket class, PSocket. The PIPSocket class supports IPv4 and we have either to make another PIPSocketIPv6 class, which will support IPv6, or to make PIPSocket IPv6 compatible.

Our current effort is concentrated on making the library IPv6-enabled, but we have not yet engaged ourselves with making the library IP-protocol independent. The reason is that we believe it is preferable to start with a relatively simpler task, which will reveal hidden protocol-dependencies in the source code and will determine the feasibility of the task.

Protocol dependencies can be found in various parts of the source code, but most of them reside in the PSocket-descendent socket classes (PIPSocket, PUDPSocket, PTCPSocket, etc.). All the aforementioned classes belong to the PWLib library. Some changes are also necessary to classes in the OpenH323 library (H323Transport, H323EndPoint) and some minor changes are required in the applications' source code.

We have also put aside for the moment the source code compatibility between multiple platforms. We have only tested our modifications on Linux machines, and we suspect that for Windows compatibility, some additional effort will be required.

The main points in the OpenH323 source code that needed modifications, according to our experience with the source code, are:

- Declaration and use of socket data structures (sockaddr_in)
 - IPv4-specific functions (inet_addr, inet_pton, gethostbyname, etc.)
 - Hard-coded IPv4 addresses (e.g. 127.0.0.1)
 - IPv4 constants (e.g. INADDR_ANY)
 - Functions that rely on the IPv4 address size (the PIPSocket::Address class for example, contains functions that return one by one, the 4 bytes of an address)
- **Robust Audio Tool – RAT**

The Robust Audio Tool (RAT) is an open-source audio conferencing and streaming application that allows users to participate in audio conferences over the Internet. These can be between two participants directly, or between a group of participants on a common multicast group. RAT was developed at UCL.

RAT requires no special features for point-to-point communication, just a network connection and a soundcard. For multiparty conferencing RAT uses IP multicast and therefore all participants must reside on a multicast-capable network. RAT is based on IETF standards, using Realtime Transport Protocol (RTP) [RFC1889] above UDP/IP as its transport protocol, and conforming to the RTP profile for audio and videoconference with minimal control.

RAT features a range of different rate and quality codecs, receiver based loss concealment to mask packet losses, and sender based channel coding in the form of redundant audio transmission [RFC 2198]. It offers better sound quality relative to the network conditions than most audio tools available. It also features encryption so you can keep your conversations private.

RAT is just an audio application, it does not perform call services like user location, and neither does it listen to session announcements to discover advertised multicast sessions. For these purposes, it is recommended you use RAT in conjunction with the Session Directory (SDR), or a similar application. RAT supports multiple sampling rates, multiple channels, 3D rendering, and the message bus (mbus) [MBUS]. RAT consists of two entities which are connected via the mbus:

- The Media Engine – which provides the bulk of the functionality, including all audio coding systems.
- The User Interface – which provides the GUI for user control over the media engine.

This architecture provides for the possibility of construction of other user interfaces or other controller entities which would communicate via the mbus to the media engine. RAT supports IPv6 operation for multicast and unicast use.

– **Video Conference Tool – VIC**

VIC is an open-source video conferencing and streaming application that allows users to participate in video conferences over the Internet. These can be between two participants directly, or between a group of participants on a common multicast group.

VIC was developed by the Network Research Group at the Lawrence Berkeley National Laboratory in collaboration with the University of California, Berkeley. VIC has been developed further by UCL.

VIC requires no special features for receiving video from a session. To send video to a session a video capture device is required, which supports the platform specific capture libraries which include; Video4linux [v4l], Video for Windows, and Sunvideo. VIC is based on IETF standards, using RTP above UDP/IP as its transport protocol, and conforming to the RTP profile for audio and videoconference with minimal control.

VIC features a range of different codecs (H.261, H.263, JPEG, H263, H263+, PVH, RAW (YUV), NV, cellb), which allow for the choice of quality and bandwidth employed. Vic provides support for layered video streams using the PVH codec. It now uses the UCL common library for Mbus operations, and cryptographic algorithms. Support for IPv6 from UCLA has been added. It also features application level symmetric encryption for private conferencing.

VIC supports IPv6 operation for multicast and unicast use.

– **Network Text Editor - NTE**

NTE is an open-source shared text editor. The collaborative text editing can be between two participants directly, or between a group of participants on a common multicast group. NTE was developed at UCL.

Using NTE can be very interactive - unless the user locks a block of text, anyone else in the session can edit that text or delete it. This is intentional. Many people can (if they wish) edit the same document simultaneously. Many people can even edit the same block of text simultaneously, but if

more than one person tries to edit the same line at one time, a conflict will occur, which results in only one of the changes being preserved.

NTE tries hard to ensure that the user does not get confused by unexpected events caused by other users - it always tells users who did what if it can. However, it cannot do the impossible, and sometimes network conditions may mean that a change arrives somewhat delayed. If this happens, NTE will reach a consistent result, but this may not be what any individual user expected. Thus we recommend using NTE as part of a multimedia conference in which it is a support tool, rather than as the only channel of communication.

IPv6 is supported through the UCL common multimedia library. It also features application level symmetric encryption for private conferencing. NTE supports IPv6 operation for multicast and unicast use.

– **Whiteboard – WBD**

WBD is an open-source shared whiteboard compatible with the LBL whiteboard, WB. The collaborative whiteboard activities can be between two participants directly, or between a group of participants on a common multicast group. WBD was originally written at Loughborough University and has since been modified at UCL.

WBD provides a shared canvas that may be edited by a number of users at the same time. WBD provides facilities for drawing various shapes, and text, in a variety of different colours. External postscript files may also be imported into WBD for collaborative annotation.

WBD utilises the Ghostscript engine for processing of postscript input. The drawing primitives, which represent the whiteboard state, are distributed between participants using an early version of Scalable Reliable Multicast (SRM) [SRM] protocol from LBNL.

IPv6 is supported through the UCL common multimedia library. It also features application level symmetric encryption for private conferencing. WBD supports IPv6 operation for multicast and unicast use.

– **High-quality Audio Tools - HAT**

The High Quality Audio Tool (HAT) provides for sending and receiving MP3 audio over Realtime Transport protocol (RTP) on IPv6. HAT uses the MP3 encoder, LAME to encode the MP3, which is taken packetised and sent out on RTP. For playback HAT retrieves the MP3 payload from the RTP packets and uses mpg123 to decode the MP3 stream.

Currently HAT works on MSR IPv6 stack (We tested it on Microsoft Windows2000 with MSR IPv6 suite installed). In the near future (hopefully the end of Dec. 2001), another version which works on Microsoft Tech Preview IPv6 will be released.

HAT supports two types of communication: n-to-n (multicast) and 1-to-1 (unicast). When you're in a network with IPv6 multicast capability, you can exchange data via IPv6 multicast simply by typing in a multicast address in the address field of HAT UI or joining an advertised HAT session through SDR. Even if in a multicast-incapable network, you can communicate with one who has an IPv6 address by assigning your counterpart's unicast address to the address field.

HAT allows you to choose the targeted bandwidth among 32, 80, 128 kbps. VBR (variable bit rate) / CBR (constant bit rate) option is also provided.

HAT presents some traffic monitoring data per participant basis: the total number of bytes received, the proportion of lost packets, and the proportion of disordered packets.

– ISABEL CSCW

The ISABEL CSCW application is a group communication tool for the Internet, based on advanced videoconferencing features. Isabel allows efficient organisation of working procedures over the Internet in large enterprises or groups.

ISABEL uses an innovative service concept, which adapts the collaboration sessions to the user needs. It includes service definitions for different usage scenarios. Each service will support the specific behaviours and characteristics of a given set of users:

- A distributed meeting needs a fairly free way of interacting where participants feel like if they were all in the same meeting room. Therefore Isabel includes a tele-meeting service which allows a fairly free way of interacting, allowing each participant to react in a fairly free way.
- A training session or a distributed lecture must allow a much more tight control of participants by the educator or lecturer. Therefore the tele-class service allows a strict control by the lecturer or educator of the interactions taking place. The student or trainee must request intervention to the lecturer before being allowed to participate.
- A distributed congress has a strict program which must be followed. Therefore the distributed congress service is script driven and is based on a centralized control which guarantees a timely delivery of the program.
- The previous services are defined with the Isabel service definition language, which allows an easy modification of the already defined services, as well as the definition of new ones.

The architectural model of ISABEL is a set of media components, which are controlled by a management agent. The management agent implements the management policy of a given service. The media components manage the media flows and the media presentation at the connected sites.

The media components are audio, video, shared workspace, VNC windows OS interconnection.

The data traffic generated by ISABEL is generated by many independent and variable rate sources. The overall traffic sent to the network by an ISABEL workstation is the aggregation of all the individual sources. A special network interface agent has been added to ISABEL, which aggregates the traffic coming from the various sources and which tries to adapt the traffic shape and rate to the requirements of the network service available. This network agent is called the ISABEL Flow Server.

The roles and functions are very similar to the roles and functions performed by the multicast server needed by ISABEL and the flow server has been extended to support multicast server functions.

Therefore the flow server is a key element, which has three fundamental roles in ISABEL:

- Creation of gateways for interconnecting heterogeneous networks.
- Adaptation of the multimedia flows to the quality of service provided by the network.
- Creation of a multicast server which connects a large number of endpoints.

– Digital Video Transport System - DVTS

DVTS (Digital Video Transport System) is an application for sending and receiving DV (Digital Video) streams using the Internet. IEEE1394 (Firewire) cables are used for connecting DV devices. However, the length of a single IEEE1394 cable can not be longer than 4.5 meters. Using DVTS, DV data can be sent anywhere using the Internet.

The DV data is sent over the Internet using RTP (Real-Time Transportation Protocol). The isochronous stream packet of DV is encapsulated in RTP/UDP/IP, with audio and video in the one stream.

The design and implementation is adaptable on the Internet regarding jitter and packet loss. DVTS also has ability to adapt to variety of network bandwidths. At the highest quality of communication, the system consumes over 35Mbps as network bandwidth, however the system can also adaptively change the bandwidth according to the end to end network conditions.

1.1.2. Protocols

– **Quadapt**

Quadapt is a heterogeneous mobile streaming platform that uses Mobile IP to offer streaming services to mobile clients. Research issue is the solution for mobility handoff. Two alternatives are investigated and compared against each other: a network level solution (Mobile IP) and an application level solution (VIC with extensions ie. SIP).

Quadapt represents a platform in which mobile devices directly connect to a multimedia-streaming server through a unicast connection. The research efforts concentrate on the function of the end-to-end platform that allows mobile devices to roam between different network technologies, possible involving different administrative domain.

The objective of this project is to investigate the pros and cons of the pure end-to-end model where mobile clients directly connect to a streaming server through a unicast connection. There are several main issues during the investigation:

- Design and implement a network layer mobility solution for end-to-end model.
- Do some measurement of performance issues of “mobility” and
- Try to improve the quality level and flexibility in application layer solution.

The design of an application layer protocol that allows mobile clients to roam between networks and domains while receiving a multimedia stream. The protocol will need to be able to dynamically handoff clients between different versions of the same stream, either from the same domain (intra-domain handoff) or from a different domain (inter-domain handoff).

A goal is to implement the protocol and to try to improve the performance in mobility handling such as reduce the handoff time, and decrease the loss of packets during roaming, and then to add functions which can improve the flexibility of multimedia streaming quality regarding to QoS issue.

– **SIP server**

Bonephone-SIP client

As a basis for any SIP communication a SIP user agent is required that is capable of initiating sessions and conveying user requests. Furthermore, as a multimedia capable application the user agent application acts as the interface of the user to the audio or video engines which generate and receive multimedia content.

Bonephone is a PC based SIP phone that acts as a SIP user agent and has the following features:

- It is capable of sending and receiving SIP messages and reacting to them.
- It provides the user with a GUI to enable him to start, answer or terminate calls as well as maintain a phonebook with SIP addresses of possible callees.
- It allows the user to establish and maintain parallel calls whereas it displays the status of the different calls.
- It integrates the audio engine needed to interact with the audio device of the system and allows the user to generate packetized audio and display incoming audio streams.

The general architecture of the Bonephone application consists of the following components:

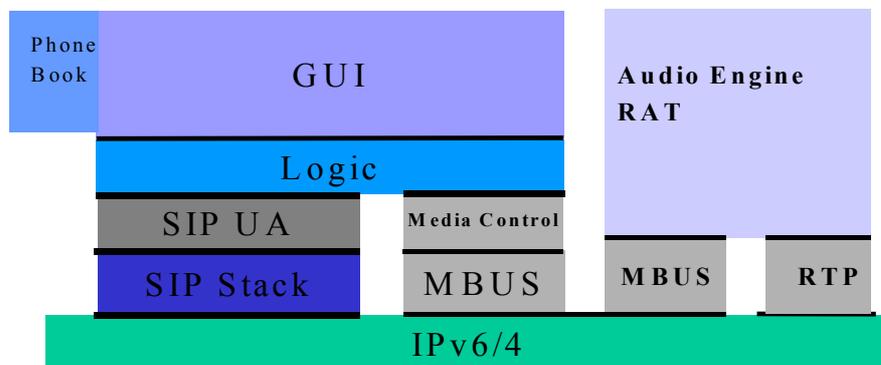


Figure 1: Bonephone architecture

Audio Engine

As an audio engine, Bonephone utilizes the Robust Audio Tool (RAT)¹. RAT is an open-source audio conferencing and streaming application that allows users to participate in audio conferences over the internet. These can be between two participants directly, or between a group of participants on a common multicast group.

Inter Process Communication Bus

The Mbus² has been designed as a basis for interaction between software components with the aim of enabling construction of complex multimedia conferencing systems out of simple (standalone) components and simplifying coordination of these various system modules.

In the context of IPv6 based SIP the Mbus carries IPv6 addresses between the Mbus entities instead of the common IPv4 addresses. Various enhancements are needed to the Mbus to avoid incompatibilities between the different Mbus implementations used by different entities in bonephone.

SIP Stack

The SIP stack is responsible for generating SIP messages, receiving them and parsing their content.

As the basis for an IPv6 infrastructure bonephone is based on the NIST SIP stack³. The stack implements an open interface based on the SIP JAIN specification⁴.

In the context of IPv6 based SIP the parser has to be extended to allow the generation of SIP messages containing IPv6 addresses as well as reading such messages.

SIP User Agent Logic

The SIP stack is primarily responsible for sending and receiving SIP messages as well as parsing the content. As an access interface it supports the JAIN SIP API. The stack does not handle issues of reliability or session management.

¹ <http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/>

² <http://www.mbus.org/>

³ <http://www.antd.nist.gov/proj/iptel/>

⁴ John de Keijzer, Douglas Tait and Rob Goedman, "JAIN: A New Approach to Services in Communication Networks," IEEE Communications Magazine, vol. 38, no. 1, Jan. 2000.

Application Call Logic

The call control is the central class for issuing outgoing calls and accepting incoming calls. It provides two interfaces: CallListener and SIPPhone. The CallListener interface is provided to the SIP-UA. The SIP-UA uses it to signal changes of existing calls or the arrival of new calls.

Media control

The media control uses the underlying Mbus package for communication with the RAT audio tool.

Graphical User Interface

The GUI is designed to be replaceable. It implements an interface “Uinotifyable” to get messages from the SIPPhone (CallCenter) and uses the SIPPhone interface to modify or create calls. The GUI’s main part manages the main frame of the application, i. e. the menu bar and the control buttons at the bottom. The Bonephone GUI displays the status of several parallel calls the user is involved in.

A SIP bonephone communication scenario:

To enable a communication between two parties the callee first needs to register his current location (i.e., IP address, another URI ...), the caller then needs to start a SIP session.

1. To start an invitation a caller needs to send a SIP INVITE message to either the callee directly if his address is known, to a proxy that controls the domain which the callee belongs to or to an outbound proxy, as shown in Figure . An outbound proxy needs to be addressed when some local service such as firewall control needs to be used, or the user’s identity and authorization status need to be checked in the local network.
2. The invite is then forwarded to the proxy responsible for the callee. The address of this proxy is obtained using DNS. Note: SIP proxies can have special entries called SRV records.
3. After receiving at the next proxy, the proxy checks a location server about the contact address of the callee.
4. The INVITE gets forwarded to the callee.
5. After accepting the call the callee generates a “200 OK” message which traverses the opposite path as the INVITE did (see step 5 and 6 in Figure).
6. Forward the reply to the caller.
7. The caller acknowledges the reception of the reply. At this stage both caller and callee are sure that the session establishment has succeeded. With the INVITE or ACK, the caller includes a description of the media types it can accept and the IP address it wishes to receive the media on. The callee adds his preferences in the reply message.
8. The end systems can exchange data. Note that the data will most probably take a completely different path than the signalling as the proxies can be located anywhere and be responsible for users distributed all over the world.

In case the user needs to be authenticated before being allowed to start a session, the initial invitation is not sent directly to the proxy responsible for the callee but to an outbound proxy which checks the credentials of the user using some AAA mechanisms.

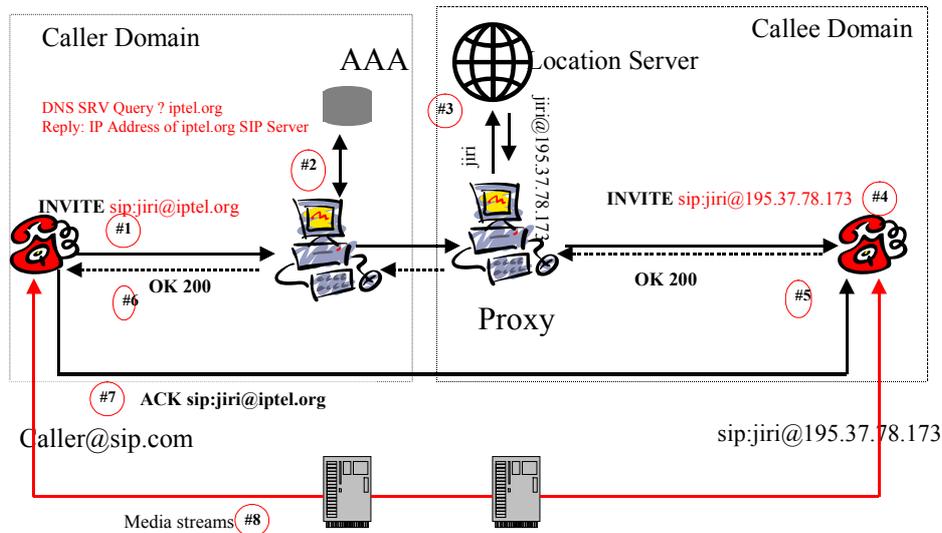


Figure 2: SIP invitation procedures

1.1.3. Session

Monitoring

– RTP Quality Matrix – RQM

The Real-time Transport Protocol, RTP, provides quality of service feedback with reception reports sent alongside the media stream. If the media is sent via IP multicast it is possible for a third party to snoop on these reception reports, displaying reception quality for all members of a group. The RQM application performs such snooping. RQM was developed at UCL.

When running RQM displays a matrix, in a window, with participant details on the left, and a number of cells to the right of these. Each row of cells denotes the packet loss rate observed for data sent from the participant indicated at the left of that row (point to a cell and a popup will appear giving the names of the source and destination of the traffic represented by that cell). The colours of the cell start as green (no loss) and fade to red (20% loss). A white cell indicates that no information is available. A light blue cell indicates that the receiver is not receiving media data from a particular sender (at present light blue is only used when an empty reception report is received, indicating that a receiver can hear no-one). Clicking on a cell will initiate an mtrace between the indicated participants, if you have mtrace installed and available in your path.

IPv6 is supported through the UCL common multimedia library.

Handling

– Multicast Streaming Tools - MUST

This application is a Web interface for simplified MBONE access using unicast or multicast connections.

The toolkit is composed of two parts:

- A multicast enabled listener application that monitors MBONE announcements and stores session info
- a cgi script for serving MBONE announcements over a web page to multicast or unicast enabled clients

– **Session Directory Tool – SDR**

SDR is a session directory tool designed to allow the advertisement and joining of multicast conferences on the Mbone. It was originally modelled on *sd* written by Van Jacobson at LBNL, but implements a later version of the session description protocol than *sd*. SDR was developed at UCL.

When SDR is running it lists all the announced sessions (including authenticated and encrypted sessions, after checking the signature and decrypting the sessions) that are currently scheduled on the Mbone. SDR listens on the standard SAP announcement multicast address for SAP packets and displays the SDP sessions in the main window. SDR allows the user to join the sessions, where the relevant tools are automatically started up on the correct addresses/ports.

SDR provides facilities for announcement of (secure) sessions. For secure sessions SDR can be used to generate symmetric and asymmetric keys. Smart Cards can be used to encrypt personal information to be used for authentication and encryption.

SDR also allows the user to make *quick calls* or multimedia “phone calls” to other SDR users, using a version of the SIP protocol.

SDR supports IPv6 announcing and announcements, though its operation is not fully tested.

– **Secure Conference Store – SCS**

The UCL Secure Conference Store is a web-based system for secured creation, storage and access to conference information. Currently, the system provides a store for session details for multicast conferences, so arranged that users can join the sessions easily via their web browser, using the SPAR Java applet to start the media tools in secure mode. SCS has been developed within UCL.

The system is mainly designed to manage secure conferences, but may be used to manage any conference. The server also maintains a cache of Session Announcement Protocol (SAP) [RFC2974] announcements, created by other announcement tools such as SDR, and allows public sessions to be created. Any user can view such public announcements and join the relevant sessions.

The browser-server dialogue is protected by HTTP-S over IPv4 and IPv6. It can be used either with a simple user chosen “username” and “password” or with a user certificate stored in the browser. Mechanisms have been provided for managing groups of users. Each potential conference participant must register with the conference store by using a certificate. This certificate is verified in the normal way. The conference organiser can then set up an authorised group of participants by simple entries on the list of registered participants; the group managers can add/remove users, create/delete sessions and change keys. Group managers can also grant management capabilities to other group members. Only these authorised members of the group can access the web store containing the private session announcements pertaining to a particular group. Users have access only to the session information for groups to which they belong and to those pertaining to the public sessions; however, they may request to join a group that is not secret (i.e. one that they can see on the server).

The server maintains a database of conference sessions, created using user-supplied details plus a random set of multicast addresses, ports and encryption keys. When setting up sessions, the server selects multicast addresses at random from a “GLOP” range [RFC2770]; these addresses are unique across all active sessions defined. Random, even numbered ports are selected, which are unique for

each of the media defined for the session. The session encryption keys are random alphabetic strings - they can be regenerated at any time (e.g. when a group manager removes a user). The server automatically removes a session after the expiry time (as set when created). When a user joins a session, a server script generates an HTML page referencing the SPAR Java applet and encodes the session details as Session Description Protocol (SDP) [RFC2327] data (passed in the HTML as a parameter to the applet). The applet is run by the browser and parses the SDP to execute the media tools on the host with the correct addresses/ports/keys required for the session. The server can run on both IPv4 and IPv6 networks and manage sessions for either. It is available as an open facility, accessible by anyone, on a secure web server at UCL.

– SDP Parser Applet – SPAR

New multicast capable multimedia applications, such as RealPlayer and Murratech Pro, can accept files containing session descriptions (in SDP format) to join a multicast conference. This allows SDP files to be served from a web page and a conference joined, using a registered SDP MIME type associated with the application.

Alternatively popular Mbone applications, such as VIC, RAT, WB and NTE rely on the Session Directory Tool (SDR) to convert the SDP into command line parameters and thus cannot take advantage of joining conferences via a web page. The solution offered here is to accept the SDP from the web browser, convert it to command line parameters at the client side and then to start up the tools with the relevant parameters. This could be implemented in several possible ways:

- a platform dependent binary running on the client machine with a MIME type for the SDP
- a plug-in installed in the browser
- a Java applet with execution privileges

The Java applet was implemented because it requires no installation or configuration by the user and is platform independent. Simplistically, the Java applet parses the SDP, extracts essential parameters and then starts the tools with the parameters on the client machine. The SDP content is embedded within the HTML as a parameter to the applet. The field terminator used in the SDP is replaced with a 'browser friendly' alternative, as SDP's CR/LF field terminator is removed by the browser. An added advantage of using HTTP to communicate the SDP content between client and server is that by using a secure web server, the SDP content (with possible encryption keys) will also be secure.

For obvious security reasons standard Java applets do not have permissions to access local resources and thus cannot execute software on the client machine. To overcome this, both Netscape's Communicator and Microsoft's Internet Explorer 4 allow applets to be digitally signed with a private key associated to a RSA object-signing certificate. If the user accepts the certificate, therefore trusting the applet, then the browser allows the applet access permissions outside the Java security sandbox.

Communicator and Internet Explorer implement different methods and technologies for digitally signing and distributing objects:

- Communicator requires Java applets to be signed using Netscape's Netscape Object Signing software. The certificate and Java code are then packaged using the JAR file structure. Signed Java applets need to explicitly request permission to access local resources, such as executing software using the Netscape's Capabilities API extensions. The request causes Communicator to prompt the user, asking them to either accept or deny the relevant permission (see Figure 2). The dialog box also contains the certificate as verification of the source and authenticity of the code.
- Internet Explorer requires Java applets to be signed using Microsoft's Authenticode software and packaged using a CAB file structure. A signed Java applet also has to request permission to access local resources by using Microsoft's Com API extensions. However unlike Communicator, the specific request doesn't prompt any user action. Instead the user is asked to accept the applets' certificate when the applet is encountered by the browser (see Figure 3) and by doing so grants universal access to local system resources.

Alternatively, browser's that do not support signed applets but do have a 'plug-in' architecture can use SUN's Java Plugin to view the applet. The Plugin requires the applet to be signed and packaged using Netscape's Netscape Object Signing software but it doesn't implement Netscape's Capabilities API. Since the API is not supported by the Plugin, any certificate accepted by the user, grants universal access to local system resources.

SPAR supports IPv6 operation.

1.2. On-line Games

– Quake

There are many different varieties of computer games on the market today. One of the fastest growing game-types is the First-Person Shooter. This sub-category of action games is based upon a relatively simple premise: You see the world through the eyes of your character.

First-person shooters have led the industry in implementing the latest technology and they usually provide very good support for multi-player gaming.

The release of Quake in 1996 by id Software marked the true beginning of the 3D age of first-person shooters. Quake was the first game to include fully-featured, real-time Internet play out of the box (<http://www.idsoftware.com/games/quake/quake>).

Quake's story revolves around the "slipgate," a device the US army has developed to instantaneously transport people anywhere. Of course, an evil fellow who calls himself Quake gets hold of a slipgate and is sending his troops to wreak havoc on Earth. The US army launches "Operation Counterstrike," of which the player is part of, to seek out Quake and annihilate him. The player must hop into a slipgate and head for Quake's home world in a desperate attempt to save the planet. Of course, this is not as easy as it sounds !

To solve this task the game player has to steer his character through a virtual world full of dangers and secrets and to fight his way to find and destroy Quake.

When playing in multiplayer mode, the different players steer their characters through the same virtual world. They can play in teams or against each other.

After three years, id Software released the source code to the Quake engine under the GNU General Public License, this lead to the start of many projects improving and enhancing the game engine (<http://www.quakeforge.net>). In this context also the multi-player capabilities were enhanced (<http://www.quakeworld.net/>). The game play of multiple players is thereby coordinated by so-called game servers.

In 2000 the game engine was also ported to IPv6 by the Canadian company Viagenie (<http://www.viagenie.qc.ca/en/ipv6/quake/ipv6-quake.shtml>). We intend to use this version in the context of 6net. The games' client and server software is available for FreeBSD 3.4/KAME Solaris 8 for SPARC and Windows 2000 and NT 4 (MSRIPv6 ver 1.4).

– Experimental Gaming Platform - EGP

Online games have certain characteristics that are vastly different from standard real-time streaming applications. Additionally different types of online games have very different requirements towards the network. Real-time games require low latency communication, either directly between the computers of the participating player or between the game device and the connecting server.

Future development (e.g. Microsoft DirectPlay) will incorporate conversational multimedia communication directly into the game play. For example, a player might want to directly talk to other players in his team. This results in a set of communication streams that are connecting game devices or servers. Based on the usage, each of the communication streams will require different QoS and communication characteristics.

The emphasis of the EGP is to focus on the communication aspects of games and to use various IPv6 features to support the game play accordingly. The EGP will be a prototypical gaming platform for demonstrating IPv6 features using online games. The gaming platform will focus on the networking aspects only (not on graphics etc.). We target to have at least two simple games with different network requirements using the platform. The games may be open-source games that are modified to use the platform. The platform will be build by enhancing an already existing SONY internal multimedia platform (AMUSE).

As already mentioned, one of the aspects we want to investigate using the platform is the combination of the game play with the possibility to communicate with other players (audio/video). For setting up gaming and conversational sessions, we are targeting to use the Session Initialisation Protocol (SIP, RFC2543). We also want to investigate the usage of QoS mechanism (if available) for example different DiffServ classes for different types of traffic. Additionally we may use the multicasting infrastructure provided with IPv6. Optionally we may investigate the usage of IPSec to prevent cheating attempts which intercept the game traffic.

– XPilot

XPilot is a networked multiplayer game created at the University of Tromso. The game is based on small, user configurable spaceships that fly in a configurable two dimensional world, described by a Map. A map contains information about the shape of the world, the gravitational forces, ship stations, treasures, weapons and other extras as energy boosts, shields etcetera.

The XPilot software consists of three distinct parts;

1: The XPilot server

The XPilot server is a piece of software that hosts a game of XPilot. It is configured with a map, and can handle a configurable number of XPilot clients. Players can play in teams or individually, depending on the map. The server will receive information from each client, and send the correct, updated information back to the client at a given framerate. The server hence controls each frame on the client. This makes the client very simple, but the game becomes very touchy about network response times. The server can also receive and distribute other information, such as chat-lines sent by players. Some sound extensions also exist, although they are not a part of the actual XPilot distribution.

2: The XPilot client:

The XPilot client can connect to either a local XPilot server or a publicly available XPilot server on the Internet. It has some functionality to verify roundtrip times to servers (ping), and also Broadcast mechanisms to locate local servers. The client can configure the player's shipshape, as well as player information. The client will then send the player's commands, and the server will update the client graphics.

3: The XPilot meta servers

The eta servers are servers that gather information about public XPilot servers on the Internet. The clients can then contact the Meta servers in order to get hints as to where available servers are. A typical site running an XPilot server does not run a Meta server, only a few Meta servers exist, and they are hard coded into the XPilot clients.

The XPilot game is perfect to stress network latencies and is therefore believed to be an interesting test application for any IP based network.

– **MUD gaming environment**

This application is a text-based multi-player gaming environment, combat oriented. The server can be hosted on any host, e.g. on home network with ADSL. Main MUD model is client server, but “server” not restricted to non-NAT locations when IPv6 is used. IPv6 MUD server code is expected to run on Linux. Client is plain TELNET client.

1.3. E-business solutions

– **IBM Websphere Portal Technology**

6net portals will provide to their associated community of users a secure, single point of access to diverse informations and applications. Each community will be able to have its own personalized environment (for instance for the support of a national language). Thanks to the communication environment of 6net and the use of appropriate application level protocols, the informations and applications will be integrated and shared seamlessly and easily providing a potential wide collaborative environment. The portals will also handle informations on the users that will allow presenting a per-user customized view. Among applications, the portals will provide the secure environments to be associated with commerce applications.

The 6net portals will be developed with the use of existing IBM Websphere Portal Server technology associated with existing IBM Websphere Edge Server technology and IBM Websphere Commerce Suite technology.

Presentation Services

The purpose of the portal framework is to produce a customized, personalized home page for its users, where the home page contents are assembled (or “Aggregated”) from a variety of content and application data sources. Desktop browsers are the typical portal access point but other devices may be supported depending on the requirements of the users communities.

WebSphere Portal Server provides a pure Java portal engine. The portal engine’s main responsibility is to aggregate content from different sources, and to serve the assembled content to multiple devices. Each content area (or “portlet”) is developed and maintained as a discrete component, it is then faster and easier to develop the overall site. The presentation details of the portlets are decoupled from those of the overall page.

The central component in the portal engine is the portal servlet. It examines the URL and header fields of each request and invokes the appropriate handler.

The request is handled in two phases. In the first phase, portlets have an opportunity to send event messages to other portlets. (For example, portlets might send events in order to update data that will be rendered in the next phase).

In the second phase, the appropriate aggregation module for the user’s device renders multiple portlets in a single page. The aggregation modules accumulate information from each portlet, put standard decorations (e.g. a title bar, edit button, and enlarge button), around the portlet, position it on the page, and generate the overall page markup.

Access to portlets is controlled by checking access rights during page aggregation, page customization, and other access points such as viewing the portlet in its maximize state.

Portlets

Portlets are the visible components that users see on their portal pages. Portlets can be as simple as an e-mail inbox or as versatile as a sales forecast from an ERP application. Portlets are very similar to Java servlets, except that they only return a subset of the output page.

Portlets are the way that other applications can snap into the portal framework. They can be written in a variety of ways. The simplest portlets might use static HTML or WML mark-up, or perhaps Java

Server Pages syntax. Intermediate portlets could use Java beans or servlets, or perhaps XML/XSL transformations. More complex portlets involve writing custom Java code. For cases where custom coding is required, WebSphere Portal Server includes an open standard Java API, called the portlet invocation API. The API provides a stable, high performance, scalable interface for portlet writers. This API is independent of the portal engine to allow interoperability of portlets among future portal engines.

User and Group Management

6net Portals provides web pages that allow users to enroll at the portal and to self-manage their own preferences and account information.

Alternatively, universities or institutes can integrate the portal with existing user directories, and may choose to disable the self-enrollment pages.

WebSphere Portal Server allows connectivity between the portal and information in various user directories.

User-specific data, such as the user name, user ID, and password is stored in a lightweight directory access protocol (LDAP) directory. The Java Naming and Directory Interface (JNDI) enables read/write interoperability between WebSphere Portal Server and the LDAP directory.

Portal-specific data, such as home page settings and portlet settings, is stored in a relational database management system (RDBMS). WebSphere Portal Server supports IBM DB2 and Oracle.

WebSphere Portal Server provides a Java bean interface for accessing user information. The User bean acts as an interface to a stateless session EJB, which in turn, acts as a consolidation interface to multiple back-end EJB classes, each responsible for retrieving a portion of the user data.

End-user Page Customization

The portal engine's customizer component allows users to modify the content and layout of their portal pages. End users can define one or more home page tabs, and then decide how the portlets are arranged inside the portlet display area of each tab.

The customization is accomplished partly through administrative setup, which defines the default settings and access rights to portlets. Further customization is accomplished through explicit user actions to change the contents and layout of the portal home page.

IBM's WebSphere Personalization product is integrated into and included with WebSphere Portal Server, so that advanced levels of personalization can be achieved. For example, personalization can be based on business rules and user profile information, in addition to explicit user preferences. The WebSphere Personalization server goes beyond simple home page customization, and supports targeting information to specific users. It offers two advanced kinds of personalization techniques:

The rules engine:

Rules engine uses business logic to select content for the user. For example, a rule might display special discounts to gold customers, but only during the summer months.

The recommendation engine:

Recommendation engine uses collaborative filtering technology to select content based on common interests or behaviours. It observes click streams that can subsequently be examined for trends. This technique is often used in commerce portals for cross-selling products.

The rules engine and the recommendation engine share user profile and content repositories. In other words, the User bean class of the portal server is already enabled for use in WebSphere Personalization Rules.

Additionally, content can be stored in any data repository and is accessed through classes implementing the WebSphere Personalization Resource interface methods.

A portlet's JSP views can use WebSphere Personalization rules and recommendations in the same way that any JSP page does. This allows the content within the portlet to be personalized, based on the rules and recommend actions. Rule a recommendation can also be used in the layout JSP templates or in the page customizer JSP to provide more advanced personalization of the portal.

Web Services

A web service is an interface that describes a collection of network accessible operations. A web service is described using an XML description language, so that the service can be invoked without prior knowledge of the platform, language, or implementation design of the web service.

WebSphere Portal Server provides support for web services. Portlets are able to use web services to perform their processing, and portal administrators will be able to bind remote portlets as web services, making the remote portlets available in the portal's registry dynamically.

For example, 6net might have several different portals, such as an art student portal, a supplier portal and a student administration portal. Each of these portals may choose to publish some of its portlets as web services for access through other the portals.

Individual portlets can also bind to web services in delivering their functionality.

For example, a search portlet might query the user for a search string, then use a search web service to search the internet. Or, calendar portlet might act as a front end, providing views for a calendar web service.

– **GLOBUS 2.0**

The GLOBUS toolkit is an open source middleware suite that supports Grid computing.

Quoting from the www.globus.org web site, "The Globus Project is a community effort, led by Argonne National Laboratory and the University of Southern California's Information Sciences Institute".

Globus is developing the basic software infrastructure for computations that integrate geographically distributed computational and information resources. The Toolkit is first and foremost a "bag of services," a set of useful components that can be used either independently or together to develop useful grid applications and programming tools... This release includes new many features, including the Globus Project's Data Grid software, MDS-2, and GRAM 1.5.

This release is also the first Globus Toolkit release to use NCSA's Grid Packaging Technology (GPT), and the first to offer binary releases on popular platforms including Linux 2.x, Solaris 8, Compaq's Tru64, IRIX 5.1, and AIX 5.1."

– **Agent framework - SoFAR, SLITE**

The Southampton Framework for Agent Research (SoFAR) is a Java framework used primarily for RMI communication. A "light" version, aimed at use of multicast for service discovery, has recently been developed (SLITE).

The framework provides a registry so that agents can advertise their services and others can find them, and it supports several communication patterns including queries and a publish-subscribe model. The framework is being used to explore the application of software agents to multimedia systems.

The IPv6 advantage lies in the addressability of multiple SoFAR/SLITE devices, who can communicate directly, peer to peer.

– **Hypermedia link services**

Hypermedia link services are a key component of many of the multimedia applications developed in the research lab at University of Southampton.

A simple link server accepts a query from a client and returns a list of available links. The query could, for example, be a location in a temporal media stream. The key performance factor in link services is latency, particularly if the link services uses referrals or query routing or if there are synchronization requirements with temporal media.

As an output of 6WINIT, in order to experiment with a IPv6-enabled link service, two servers were ported:

- DLS1 is a specialised service with an HTTP interface. It is a standalone reference implementation in use by current research projects. DLS1 is a ‘context sensitive’ link service that makes use of information about the device making the query, as part of a pervasive computing infrastructure. Hence in addition to porting of the networking code, we are also introducing IPv6 addresses as part of the context handling mechanism.
- DLS2 is based on an LDAP directory service and provides a distributed implementation. DLS2, based on work in collaboration with BT, has recently been interfaced with the agent framework

– **FunnelWeb**

FunnelWeb is a system that runs on a node to provide an active services platform. Active services are loadable objects which provide particular application level functionality in the network. FunnelWeb is an implementation of an Application Level Active Networking (ALAN) active networking execution environment (EE). FunnelWeb was developed initially at UCL and has continued development at UTS, Australia under BT funding.

Specifically FunnelWeb provides an execution environment for java based active applications, known as proxylets. The FunnelWeb EE is termed the Execution Environment for Proxylets (EEP), which provides a java environment with a Remote Method Invocation (RMI) control interface for loading, running, modifying operation and stopping proxylets. The proxylets are java applications implementing the *Proxylet* base class, which exposes methods for initialising, starting, modification of operation, and stopping of the proxylet.

IPv6 functionality is possible using the Java JDK 1.4 which provides IPv6 functionality on Solaris8 and Linux.

– **Transcoding Active Gateway – TAG**

The Transcoding Active Gateway (TAG) was developed to extend the functionality of an earlier tool, known as the UCL Transcoding Gateway (UTG). The implementation was based on the FunnelWeb [ALAN] Active Networking architecture. The key points to the new design were:

- Automatic configuration of a multicast session using the Secure Conference Store.
- Use of Active Networking for locating and positioning a reflection point
- re-multicasting of the reflected media streams on the client
- Modular approach to media relays, implemented as Java proxylets

As mentioned previously, TAG builds upon Funnel Web to provide its functionality. It was essential that components of the system were separated, to provide an easy upgrade when a new version of Funnel Web became available. The TAG client application is separated into two components that communicate using Remote Method Invocation (RMI):

- The Funnel Web EEP component of the client runs the Routing, Discovery and local Reflector proxylets. The Routing and Discovery proxylets are used by the client to identify its location in relation to other parts of the Active Network.
- The user interface component of the client is used to communicate both with the EEP component and with a remote EEP via the RMI interface. The server configuration section of the user-interface allows the user to query the local Routing proxylet for information regarding the current

EEPs available and the closest EEP in relation to the local host. Once an EEP has been selected the controls for starting, stopping and configuring media streams are enabled.

Each media reflector is a separate self-contained package, known as a *proxylet*, written in pure Java. An EEP downloads the proxylets from a central web server, allowing them to be easily upgraded. Proxylets contain additional information for execution on a Funnel Web EEP. The added metadata policy file limits control of the proxylet to a range of hosts, although this facility is not currently implemented. The Reflector proxylets process both the incoming unicast stream originating from the opposite end point and outgoing multicast streams originating from the multicast group/tools. In the case of video and audio, reflectors also process the additional RTCP stream. Simple rate limiting cuts off forwarding after a pre-set rate limit (in packets per second) has been reached. The rate can be altered from the client using the RMI connection to the server, which then in turn passes the new rate to the proxylet.

TAG has recently been updated to provide an additional mode that allows it to connect a client into a conference VPN and subsequently provide access to the VPN media streams. Using this mode, a client will automatically request to join the VPN once local multicast activity is detected. The join request takes the form of a registration of a local EEP to a Reflector Manager (RM). The registration triggers the RM to notify each node within the VPN to forward media streams back to the registered EEP. In addition the RM can be used to convey rate and access information to each remote node.

A number of extensions are required to the TAG to add functionality to support issues such as security and mobility. These extensions will be explored over the next few months. Some, concerned with the active networks aspects, will be carried out under ANDROID; some, which are required for the 6WINIT wireless environment, will be implemented under 6WINIT. These extensions are listed below:

- Media conversion (transcoding) and rate control in the Reflector Proxylet
- Quality of service improvements to the reflection
- Improvements to the discovery and location of Active Servers/EEPs
- EEP access control to prevent unauthorised user access.
- Services offered by an EEP
- Support for Foreign Agent functionality
- Secure communications and authentication of client
- XML messaging throughout design
- Policy control of Reflectors

- **TZI Stargate**

StarGate provides call signalling and media transcoding gateway functionality for connectivity between different kinds of endpoints interconnected through different types of networks (hence the name *Gate). This is expected to include in particular:

- Conversion between the three most important call signalling protocols (H.323, SIP, and ISDN) including media stream conversion if necessary;
- Actively accessing Mbone sessions from H.323 endpoints; and
- Inviting H.323 endpoints into Mbone sessions for audio and optional video communications.

StarGate was developed under the joint auspices of TELES and UB TZI. It also contains code from UCL's RAT tool.

The architecture of StarGate also allows for extension of the number of supported call signalling protocols. In addition, if feasible from the standardisation point of view (i.e. the necessary specification are complete and stable), security aspects will be incorporated into the StarGate implementation.

StarGate is conceptually built upon the same general Mbus architecture as AudioGate [AG], with largely different Mbus entities and different interactions between them, of course. AudioGate provides a dial-in point for users on any telephone network and enables them to participate in Mbone audio conferences.

All three of the aforementioned control protocol entities share a core set of Mbus messages to set up, tear down, and monitor progress of a call. In addition, each entity supports protocol-specific Mbus extensions that may not be (easily) mapped to other control protocols. The Mbus controller is expected to understand all these Mbus commands, route incoming messages, and optionally perform translation between different protocols.

Call control messages are intended for interaction with call control and invitation protocols such as H.323 and SIP. They are designed to constitute the union of the call control messaging needed by endpoints, gateways, proxies, multi-point controllers, and gatekeepers. This allows the use of the Message Bus to act as gluing mechanism to create any type of system from roughly the same building blocks. Mbus call control messages are based on a common basic message set defined in the following that will be supported by any kind of call control protocol entity. The basic message set may be augmented by protocol-specific extensions required for protocol specific interactions between a local controller and/or local applications on one side and the respective protocol engine on the other. While the basic Call Control commands have been worked through, they still need to be mapped to H.323, SIP, and ISDN-specific messages.

– **LightWeight Directory Access Protocol - OpenLDAP**

OpenLDAP is used in many middleware applications and it is important that it is available over IPv6. OpenLDAP Software (<http://www.openldap.org/>) is an open source implementation of the Lightweight Directory Access Protocol (v3), base protocol is specified in RFC 2251. OpenLDAP is a simplification of the X.500 DAP. We have ported OpenLDAP to IPv6 and this is in the official 2.x versions.

We want to get OpenLDAP tested on a wide variety of platforms, and make any necessary changes to make IPv6 work on those. Some changes have already been necessary, since IPv6 stack implementations do things in slightly different ways. We will also see if we can use OpenLDAP as a proxy to give IPv4 clients access to IPv6 LDAP servers and vice versa.

A number of 6Net applications depend on LDAP, and should obviously use it over IPv6. Part of the work will be to assist them with any LDAP IPv6 problems.

– **Public Key Infrastructure – PKI (University of Murcia)**

The purpose of a Public Key Infrastructure (PKI) is to define the mechanisms and elements needed to manage and enable the effective use of public key encryption technology on a medium or large scale.

The base components are a certification authority, one (or several) registration authorities, and a directory server. Some additional components, like smart cards, time stamping servers, OCSP servers, can be present depending on the services offered by a particular PKI implementation.

The Public Key Infrastructure of the University of Murcia is based on the design and implementation of IPv6-enabled X.509 certification services. It can be used by an organisation to provide its users with a range of public key mechanisms for securing their communications.

The end users of the PKI are able to carry out the majority of their PKI operations using a web browser, e.g. to request a certificate, renew it, revoke it, or look for another user's certificate. The PKI also allows the use of smart cards to store cryptographic information, enabling greater key mobility and also increasing the security of the system.

Users with web browsers can use their smart cards to store the cryptographic information by means of the Microsoft CSP (Cryptographic Service Provider) and the RSA PKCS#11 modules also developed by UMU.

– **Request Server**

This component is responsible for storing all certification, renewal and revocation requests generated by final users or other components of the PKI. These requests are stored in an internal database so that the CA can access to them when required.

It is important to recognise that there is no direct connection between this server and the CA. The CA always works in ‘off-line mode’ and never accepts incoming connections for security reasons.

– **Registration Authority**

This component is responsible for examining certification requests and then sending these to the CA for certification if they are successfully validated according to the systems Certification Policy. The PKI may have several Registration Authorities each with an administrator.

– **Certification Authority :**

This process is responsible for processing valid requests stored in the Request Server. In the case of a certification request, if the certificate can be issued according to the PKI internal policy and the RA validation process, it is signed, stored in the internal database and published in the LDAP server. Then the user is notified through a digitally signed email message. In the case of renewal requests, the certificate is updated in the internal database and in the LDAP server. For revocation requests, the certificate is marked as revoked in the internal database and is included in the next CRL published by the CA in the LDAP server.

– **Policy :**

One of the main features of the University of Murcia PKI is the use of Certification Policies designed according to standards from the IETF. All operations performed by a PKI component should be compatible with the subset of the Policy associated with the component or they will not be performed.

Certification Policies define the restrictions that an organisation can apply to the certification operations of its users. For example an organisation may require that every issued certificates make use of RSA keys, with a minimum length of 2048 bits, or that a particular certificate can only be renewed inside some specific time period.

– **Certificate Repository :**

The University of Murcia PKI supports an LDAP server, which acts as a public repository, where CA certificates, user certificates and CRLs can be accessed.

– **Smart Cards :**

Smart cards are used to store cryptographic information (private key, user certificate and CA certificate) for a user. They can be used in the Java components and in popular web browsers (through the cryptographic modules – RSA PKCS#11 and Microsoft Cryptographic Service Provider – developed by the University of Murcia).

1.4. Edge-services for IPV6

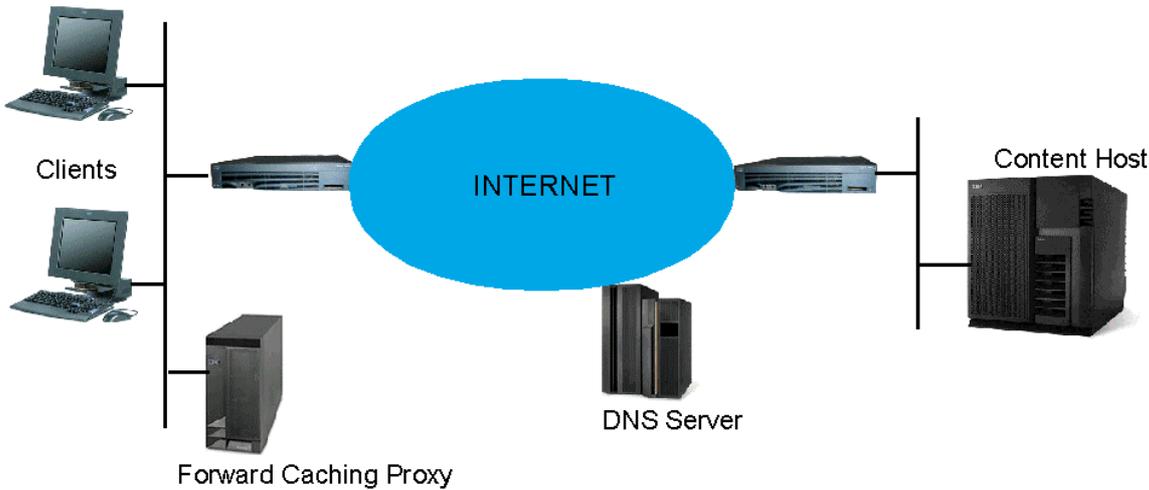
– **Edge Server Proxy**

IBM Edge Server is a powerful device providing a better service both to users who access information on the enterprise’s server and to internal users accessing to the internet. Such devices are close to the boundary between the enterprise’s network and the internet, that’s the reason for the name Edge Server.

Four systems are included in IBM Edge Server: Network Dispatcher, Application Service at the Edge, Content Distribution and Caching Proxy. The Caching Proxy intercepts a request from a client, retrieves the data from content host and sends it back to the client. Although HTTP(S) requests are often done, it can also deal with FTP and Gopher traffic. The Caching is done by storing cacheable content before sending it to the client, so next requests to the same content can be delivered more quickly and with saving network bandwidth.

Proxys can be used in two different ways: Forward Proxy when located on the client’s network, and Reverse Proxy when located on the server’s network.

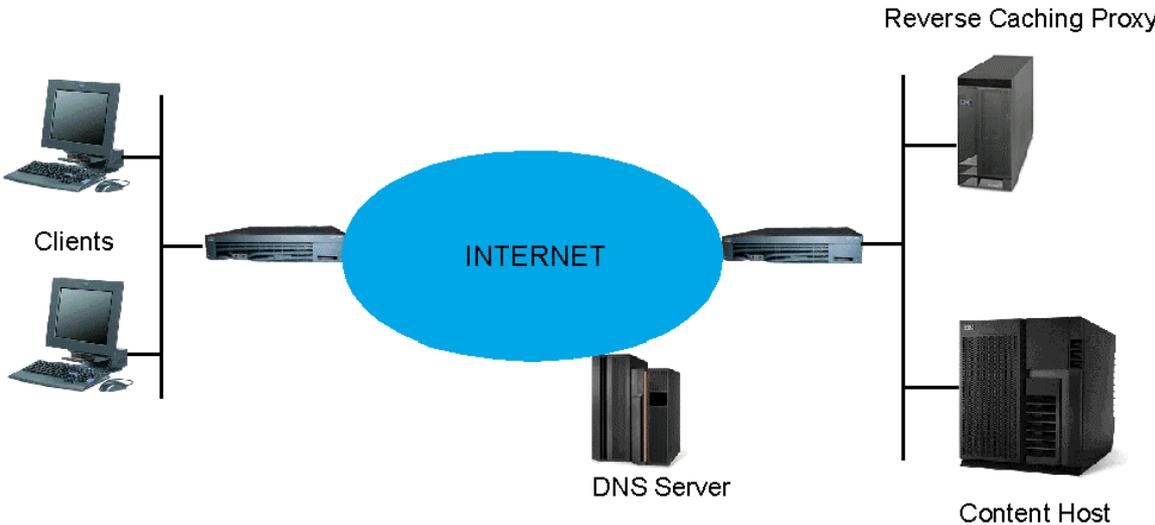
Forward caching proxy



Clients’s browsers are set to send their request to the Proxy. Then the Proxy performs a DNS request to get server’s IP address, and it sends a new request with its own IP address as source address. As it receives data from content host, it stores cacheable pages (static pages, slowly changing JavaServer pages, or fragments) and then sends it back to the client. Storing process allows to satisfy next requests to the same content.

Benefits brought by a such system are making clients less vulnerable by hiding them from internet, saving up global IP addresses, making responses faster and preserving bandwidth due to the caching.

Reverse caching proxy



In this configuration, the server’s URL known by internet clients is actually the Proxy’s one. When a client performs a request, the Proxy intercepts it and makes a new one towards the content host with its own IP address as source address. It retrieves the data and stores it in cache if possible, then sends it back to the remote client.

This solution makes the existence of a proxy absolutely transparent for the client. It also provides faster response to clients and it lightens content host processing. It can also load balance at the same time.

Other functionalities

The Cache Memory allows a wide range of settings to customise the caching, like URL and IP filters, caching duration, page pre-loads, etc.

While security is very important for each server on the web, Edge Server Proxy includes SSL protocol, key management, SOCKS function and cryptography hardware support.

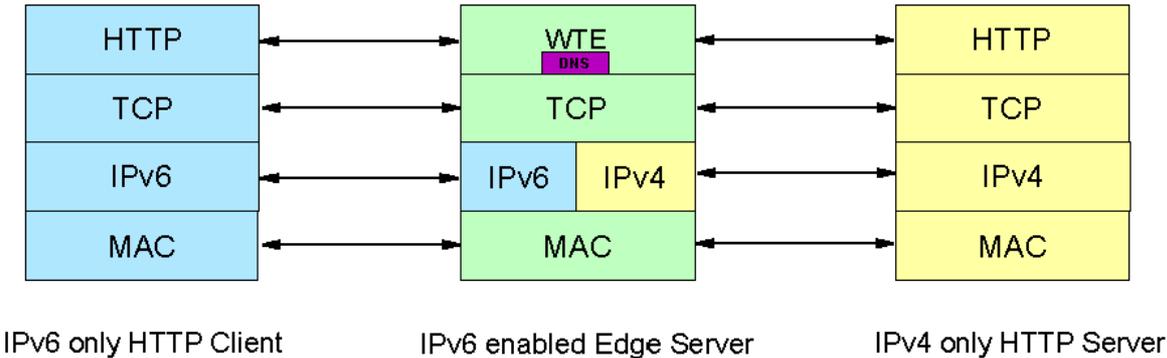
Edge Server Proxy manages a lot of logs to monitor its activity, and each one can be filtered to get only the information needed. It also provides SNMP support. Finally, it accepts some plug-ins for enhanced filtering, LDAP, ...

IPv6-enabled version

The current IPv6-enable prototype is based on Edge Server Proxy v1.1 but will be brought to version 5.0.

Current functions

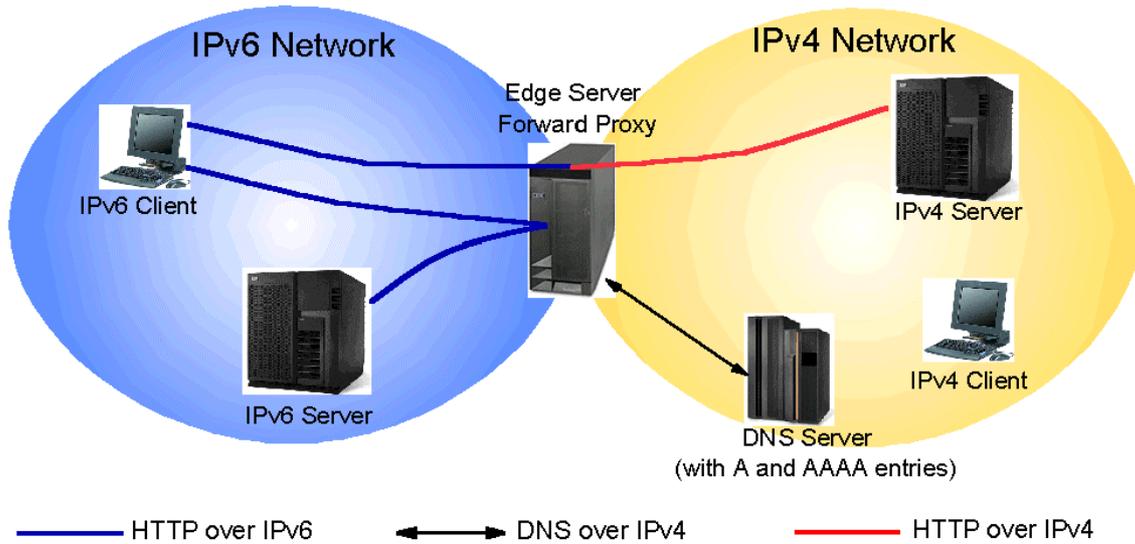
Web Traffic Express (Proxy’s name within the Websphere suite) works upon a dual stack for processing both IPv4 and IPv6 requests. The following picture shows protocol stack used in the case where an IPv6 only client asks for data on an IPv4 only server.



The scheme is quite the same with IPv4 client and IPv6 server. With two same IP version hosts, flows are common.

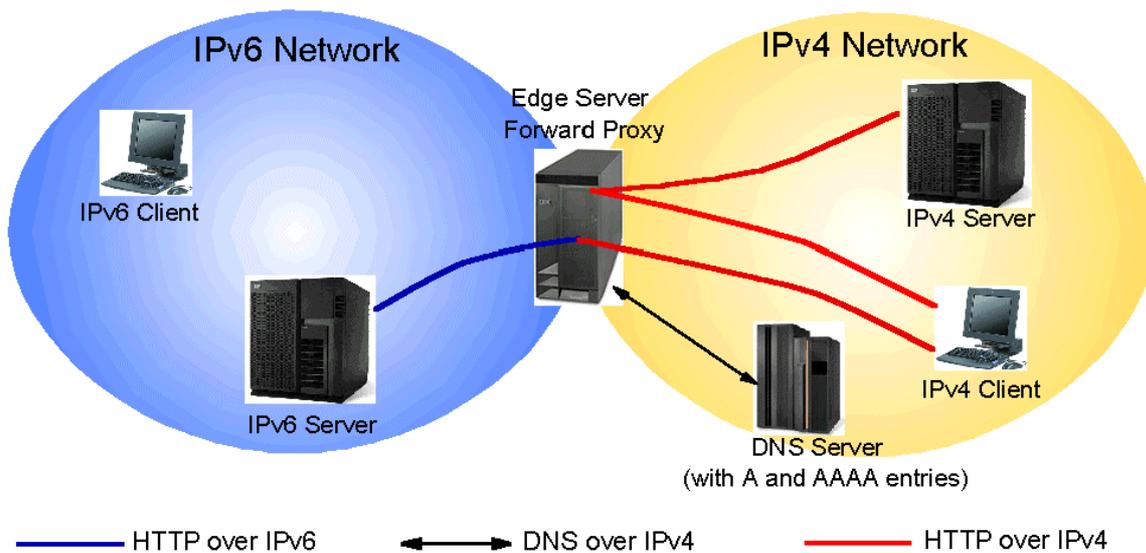
- The Edge Server Proxy can be listening, for example on usual TCP port 8080 for requests from IPv4 clients and on port 8090 for IPv6 ones. Obviously, that implies clients’s browsers have to be set to the right port too, depending on their IP version.
- Until today, IPv6-v4 Edge Server Proxy has been only tested as a forward proxy. Here are two schemes describing what has been done:

IPv6-only client accessing IPv4 and IPv6 servers transparently:



The DNS server is IPv4-only but allows IPv6 address entries (AAAA type). So, when The Proxy receives a request from the client it asks the DNS server to send IP address fitting the requested URL, either IPv4 or IPv6 address, then it performs its own request, and so on. From the client, the IP version of the servers is transparent while it asks the proxy to perform the request from the server's URL.

IPv4-only client accessing IPv4 and IPv6 servers transparently:



32603	Deliverable D5.1 – v2	
-------	-----------------------	--

- Other functionalities like reverse proxy are under development, but many functions from the original Edge Server Proxy are available (Caching, SSL, Logging, Access control ...).

Technical features

- The prototype is running on a Pentium 3 / 512 MB PC
- Software prerequisites are:
 - Linux Red Hat 7.0:* Even if the original version runs on many OS (Linux, AIX, Solaris, Windows), the prototype currently works on this version.
 - 2.4.2 kernel with IPv6 built:* Actually, Linux kernel has to be recompile to get IPv6 functionalities.
 - GSKit 5:* For SSL sessions.
 - Original IBM Edge Server Proxy v2.0:* It is used to support the prototype version.
 - IPv6-enabled IBM Edge Server Proxy:* Currently based on IBM proxy v1.1.

Proxy services are already widely spread. Main users are ISPs, Campuses, content distribution and e-business networks.

– Contents Delivery Networking - CDN

Content Distribution networks scale and accelerate content services by distribution content at the edge of the network and redirecting client request to the most appropriate edge server by means of a content routing process.

A CDN consists of a content distribution management function responsible for optimizing the distribution of content to the edge of the network, a content routing process to redirect client requests to the closest edge delivery node whereby closest is related to a metric based on RTT and content availability and the edge delivery node serving the content using different protocols like HTTP or RTSP.

6Net will study how the CDN infrastructure can be used to accelerate and scale other application services developed in Workpackage 5.

2. Summary sheets

Each application is summarized in a one page form according to the following template:

- Functional description, with emphasis on benefits brought by IPv6
- Name of application
- Prerequisites:
 - Client (hardware, software)
 - Server (hardware, software)
 - Network (usage of IPv6, bandwidth requirements, etc)
- Protocols used
- Development / porting effort, testing
- Deployment, monitoring (client, server, network, tools, resources...)
- User community
- Evaluation and future development

1. Video over IP - VIP

Activity: 5.1	Proposers: TELIN, IBM
Application name: Video over IP - VIP	
Overview Video environment associating advanced video-editing, search and retrieval from client's browser based applications using the breadth of v4/v6 capabilities.	
Functional description – Benefit brought by IPv6	
<ul style="list-style-type: none"> • A central repository provides a set of contents in several media formats: <ul style="list-style-type: none"> – Moving Pictures Experts Group (MPEG-1) – Video Files (.AVI) – Apple QuickTime Version 3,4,5 or all versions – MPEG-2 and MP3 – MPEG 7 • Searching capabilities are provided using XML/Mpeg7 • Editing is done thru on-line composition of play-lists • Embedded encoder support lets the application control the MPEG encoder for functions such as real-time IP Multicast and real-time IP Multicast with live recording of the same stream • Distribution is done thru <ul style="list-style-type: none"> – TCP for content serving thru firewalls – RTP/RTSP for classical media delivery – IP multicast for broadcasts – encoding is selected/adjusted based on client server path/policy available – fully enables video/audio delivery from low bit rate to business-quality MPEG1 and MPEG2 (up to 15 Mbps) streams 	
Prerequisites	
clients :	
hardware :	
intel pentium 3/4	
software :	
OS tbd	
browser tbd	
IBM videocharger client	
Microsoft MediaPlayer	
server :	
hardware :	
intel pentium 3/4	
software :	
win2000	
IBM videocharger	
Microsoft Media Server	
network :	
bandwidth must be high enough for streaming services up to 15 Mbps	
Protocols	
RTP, RTSP, TCP, IP multicast, RSVP (need videocharger on AIX)	
Development / porting effort, testing	
<ul style="list-style-type: none"> – Reuse applications that was designed in the Dutch Video-over-IP project – Adapt Videocharger V6 stack 	
Deployment, monitoring	
Central service would be delivered from Telin DataCenter thru Surfnet	
User community	
E.g. students in art/movie/design	

2. Video distribution - MPEG4IP

Activity: 5.1	Proposer: GRNET, TELIN
Application name: MPEG4IP	
Overview MPEG4IP live streaming & Darwin Streaming Server environment for wide video distribution	
Functional description – Benefit brought by IPv6 An open source encoding and streaming set of tools capable of serving multiple media formats. <ul style="list-style-type: none"> – MPEG4 live encoding and streaming – MP3/MPEG1/MPEG2/Apple QuickTime streaming – Unicast/multicast streaming – Audio/video streaming from low to high bit rates Benefit of using IPv6 flow labels can be investigated.	
Prerequisites clients : <ul style="list-style-type: none"> hardware : <ul style="list-style-type: none"> intel pentium II/III/IV software : <ul style="list-style-type: none"> Mpeg4player for Linux, Win, Mac, Solaris, FreeBSD QuickTime player for Win, Mac server : <ul style="list-style-type: none"> hardware : <ul style="list-style-type: none"> intel pentium III/IV software : <ul style="list-style-type: none"> Linux, Win, Mac, Solaris, FreeBSD Mpeg4live Mpeg4encode Darwin Streaming Server network :	
Protocols RTP/UDP, RTP/RTSP/TCP	
Development / porting effort, testing <ul style="list-style-type: none"> – Testing of IPV6 streams – Porting of client & server streaming tools. 	
Deployment, monitoring - Deployment and support for Greek Research Network community of 60 member institutions	
User community - Greek Research Network community.	

3. Real-time conferencing tools - GnomeMeeting

Activity: 5.1	Proposer: GRNET
Application name: Real-time conferencing tools	
Overview GnomeMeeting is an open source H323 application for Linux based on the OpenH323 platform	
Functional description – Benefit brought by IPv6 Real-time conferencing applications are becoming very popular but few are available for non-Windows OSs. GnomeMeeting is a Linux based H323 compatible application that claims interoperability with Windows based H323 clients, thus opening the possibility for a universal conferencing platform. Benefits from IPv6 support will be QoS features and security. Features include: <ul style="list-style-type: none"> - GUI interface for popular Linux window managers. - Gatekeeper and directory support. - Audio – only mode. - Automatic device detection 	
Prerequisites clients : <ul style="list-style-type: none"> hardware : <ul style="list-style-type: none"> intel pentium 3 Audio Card and optionally Video4Linux compatible Video Card software : <ul style="list-style-type: none"> Linux Video4Linux server : <ul style="list-style-type: none"> hardware : software : network :	
Protocols RTP/UDP H323, H261, H263, G711, GSM, LPC10	
Development / porting effort, testing <ul style="list-style-type: none"> - Testing - Porting 	
Deployment, monitoring <ul style="list-style-type: none"> - Deployment and support for Greek Research Network community of 60 member institutions 	
User community <ul style="list-style-type: none"> - tens of thousands of users 	

4. Music distribution - TUR on IPv6

Activity: 5.1	Proposer: UNINETT
Application name: TUR on IPv6 - music distribution	
<p>Overview</p> <p>We will do streaming of MP3 and possibly other formats. At present this is done 24x7 freely available to all the world over IPv4, see http://www.turmusic.no/ This site is named "Trondheim Underground Radio ("TUR for short). We will make this content available for IPv6. First unicast and later multicast. We expect (but can not guarantee) to distribute this on the same terms throughout the 6NET project period. The setup will be documented, and any software developed will be freely available, so that others can use the same solution</p>	
<p>Functional description – Benefit brought by IPv6</p> <p>Currently we offer 48 kbps and 128 kbps IPv4 unicast stream, using the shoutcast/icecast protocol (TCP). There are a number of available MP3 players to be used for decoding such streams, and some of them support both unicast and multicast. TUR has been experimenting with multicast using liveCaster (http://www.live.com/liveCaster/). We would like to base the primary service on software developed in the Icecast project (http://www.icecast.org/), which has been ported to IPv6. Some of the clients supporting IPv6 will be tested. Icecast has no native support for multicast, but we would like to support multicast streaming of TUR as well as unicast, both on IPv6 and on IPv4.</p>	
Prerequisites	
<p>Protocols</p> <p>Several streaming solutions (protocols, servers, and clients) will be tested, but at the moment unicast streaming with Icecast seems to be the most easily deployed solution with respect to the users.</p>	
<p>Development / porting effort, testing</p> <p>During the project period it would be interesting to include development of some statistical tools to present usage of the service.</p>	
<p>Deployment, monitoring</p> <p>UNINETT will provide necessary hardware in cooperation with TUR, and will be connected to UNINETT's IPv6 network.</p>	
<p>User community</p> <p>Anyone that wants to listen to music from Trondheim, and has a computer and internet connection with the necessary bandwidth.</p>	

5. Multimedia Conference Recorder - MMCR

Activity: 5.1	Proposer: UCL
Application name: Multicast Multimedia Conference Recorder - MMCR	
<p>Overview</p> <p>The Multicast Multimedia Conference Recorder is a system capable of recording and playing back multimedia data sent over the Multicast Backbone (Mbone). It's designed for recording and playing back multicast multimedia conferences over the Mbone. MMCR exists in two forms; The java application based system, and a FunnelWeb proxylet implementation which is currently in a beta form.</p>	
<p>Functional description – Benefit brought by IPv6</p> <p>All server components have access to the database archive to store/retrieve recordings and information about them. Much research has considered ways of providing efficient storage/access mechanisms for Video On Demand (VOD) systems, which require high bandwidth delivery. However, the simple disk model used here is adequate considering the current bandwidth limitations on the Mbone and a Redundant Array of Independent Disks (RAID) can be integrated in the system as an enhancement, if necessary.</p> <p>The Server: The server acts as the single point of contact for recording, browsing and playback. Most of the existing implementations have a similar architecture. They consist of independent components; a server manager, the player, the recorder and the browser. The server manager controls the whole service; it handles the establishment of connections with the clients. It has a separate, independent interface for each task and more interfaces can be added when required.</p> <p>The Recorder: To record the media streams the recorder need not be an active part of the conference; it 'listens' to the specified multicast groups and collects the data. Each stream is stored separately. In the case of RTP media, the RTCP messages transmitted are stored along with the data packets. Information about each recorded media and each source is saved in header files.</p> <p>The Browser: A listing of conferences a server has stored in its archive can be obtained through the browsing mechanism. A title keyword search facility is also available to help identify titles of interest. Further details about a particular conference can also be obtained to assist a user in deciding which conference to play back.</p> <p>The Player: This allows utilisation of all kinds of networks as users with bandwidth limitations may choose to play a subset of the available streams. The player schedules real-time packet transmission based on the timestamp in the index entry. RTP compliant media provide additional information in the RTP header that can be used for providing smoother playback. Other media packets are sent on the network based on their received timestamp.</p>	
Prerequisites	
Protocols: RTP.	
<p>Development / porting effort, testing</p> <p>MMCR is implemented as a client and server Java applications.</p> <ul style="list-style-type: none"> – MMCR requires no additional hardware – MMCR is available on request from: http://www-mice.cs.ucl.ac.uk/multimedia/software/mmcr/. 	
Deployment, monitoring	
User community	
<p>Evaluation and future development</p> <p>MMCR is currently fairly stable, and provides good performance.</p>	

6. Video Lan

Activity: 5.1	Proposer: UoS
Application name: VideoLAN	
Overview VideoLAN is an open source project that provides unicast and multicast a media streams from a variety of media sources. See: http://www.videolan.org/ for details.	
Functional description – Benefit brought by IPv6 Can source from a hard drive, a DVD player, a satellite TV card or an MPEG2 compression card. Can create streams with data rates of up to 6-9Mbit/s for DVD, less for MPEG-1. May be good multicast demonstrator. May be good QoS demonstrator.	
Prerequisites Built for Linux, but other OS's supported for clients. Client runs on Linux, Windows, Mac OS X, BeOS, *BSD, Solaris, iPaq, QNX (but IPv6 support may not be on all platforms at this time... further porting work may help here). Server Linux-only Need video source. Suggested Pentium 400 with 32MB RAM	
Protocols MPEG-1 and MPEG-2 streaming	
Development / porting effort, testing VideoLAN has been ported to IPv6 by Btexact for 6WINIT, and those mods have just this month (April) made it back into the main source tree. Further tests and modifications may be required. BTexact's code also includes a web-based source (movie) selection system for video on demand.	
Deployment, monitoring Server allocated in UoS, could stream to any partner.	
User community UoS has a satellite TV feed installed on the roof of its main building, and can thus readily stream any source of open-to-air UK television. It may be interesting to approach the BBC to see if we could stream News 24 (for example) as part of this research-oriented project. Could use VideoLAN to stream lectures or guest speakers.	

7. Multicast Radio

Activity: 5.1	Proposer: UoS
Application name: Multicast radio (Cradio, Icecast)	
Overview Cradio is an MP3 jukebox, home grown and already supporting IPv6. AN alternative package, icecast, is also available with IPv6 support. Here we propose Cradio, but we could run icecast also with similar MP3 source and user community.	
Functional description – Benefit brought by IPv6 Web-based MP3 track selection and queueing. Multicast operation, thus good test of multicast IPv6 network. Server can re-reference location of MP3 files via HTTP. Main modules: <ul style="list-style-type: none"> – Server Side Multicast IPv6 MP3 Provider – Web Based MP3 Selection Facility – Client Side Multicast IPv6 Cradio Player – Registration of Playlists Uses existing MP3 player, with mpg123 being favoured.	
Prerequisites Originally built for Linux. Also runs on FreeBSD. A Java client is available for IPv6 (jradio).	
Protocols Proprietary – doesn't currently use RTP. Playlist uses XML format, created automatically using the ID3Tag information within the MP3 files.	
Development / porting effort, testing Potential further effort to make tools standards-based and to modify for general audio (so we can use a live radio source).	
Deployment, monitoring Within UoS site, available to all.	
User community UoS has a local student radio station that we propose to use as the source of the Cradio traffic.	

8. Unified messaging system - 6UMS

Activity: 5.1	Proposer: UoS
Application name: Unified Messaging System	
Overview An IPv6-enabled unified messaging system (6UMS) allows peer-to-peer communication between users using a variety of media.	
Functional description – Benefit brought by IPv6 Includes messaging using text, audio, images and video. Includes <ul style="list-style-type: none"> • location awareness • user context and preferences • intrusiveness consideration UoS has SMS relay tools available. During project lifetime we expect to communicate with advanced GPRS and 3G devices. Primary focus is WLAN PDA devices. Major benefits lie in addressability and security.	
Prerequisites Still being specified – Linux and Solaris initially, Windows XP later	
Protocols Will include consideration from many IETF WGs, e.g. SIMPLE (and thus SIP) and IMPP.	
Development / porting effort, testing 6UMS is being developed by UoS in Euro6IX, but will be made available to 6NET. Existing tools will be re-used where appropriate.	
Deployment, monitoring Available to all partners, in 6NET and Euro6IX.	
User community In theory any UoS user running IPv6, and anyone at any 6NET/Euro6IX site.	

9. Kasenna Mediabase XMP streaming server

Activity: 5.1	Proposer: DTU
Application name: Kasenna Mediabase XMP	
Overview Kasenna MediaBase XMP is a system for management, distribution and streaming of video and audio assets encoded as MPEG-1, MPEG-2 and MPEG-4 video or MP3 audio.	
Functional description – Benefit brought by IPv6 The system supports various MPEG encoders that can act as sources for live streams, and are redistributed by MediaBase XMP as either multicast streams, unicast streams or both. Stored assets can be scheduled to multicast or made available on-demand. Each asset can have metadata associated with it, and it is possible to create multiformat assets and to define clips and sequences from the assets. MediaBase XMP also includes a content distribution module, for use with several servers, but it is not the intention to set up more than a single server.	
Prerequisites <ul style="list-style-type: none"> – Client (hardware, software) Hardware: PC, Mac, Set-top box Software: Kasenna Broadband Player (MPEG-1,2), Windows Media Player (MPEG-1,2), any ISMA compliant player, e.g. Philips WebCine, EnvivioTV (MPEG-4), Apple QuickTime Player ver. 6. – Server (hardware, software) Hardware: Dual Pentium 3/4 or Dual Athlon Server with fast disks (also available for SUN and SGI servers). Software: Red Hat Linux 7.2, Apache, OpenLDAP, Informix, MediaBase XMP (also available for IRIX and Solaris) – Network (usage of IPv6, bandwidth requirements, etc) Bandwidths from few hundred kbit/s up to several Mbit/s. 	
Protocols TCP, UDP, RTSP, RTP, IP multicast	
Development / porting effort, testing Migrate existing MediaBase Enterprise Edition (5.0) to MediaBase XMP and IPv6.	
Deployment, monitoring Streaming server with Kasenna MediaBase XMP will be deployed in the facilities of Forskningsnettet (the Danish Research Network).	
User community Researchers, teachers, students.	

10. FreeAMP

Activity 5.1	Proposer: GARR
Application name: FreeAmp	
<p>Overview At the moment there is a lack of high quality client applications on Win32 platform. FreeAmp is a free MP3 player that supports both unicast and multicast MP3 streams and it is available on Win32 and Linux.</p>	
<p>Functional description - Benefit brought by IPv6 This software is currently maintained by an open source group (http://www.freeamp.org). Our porting aims at making this software IPv6-compatible in order to be able to play both IPv4 and IPv6 streams. The focus is on Win32 platform, where the number of IPv6-compliant tools is quite limited. This software can be used as a client for a certain number of other 6Net partners which are offering MP3 streaming (UNINETT) The code will be freely released on the web. Furthermore, we plan to activate both a unicast and a multicast MP3 source in our network, which will be available to all 6Net partners. This source will be used to test our software.</p>	
<p>Prerequisites Client: a Windows 2000 / XP / Linux machine with the IPv6 stack installed on it Server: any server which sends MP3 data streams Network: the requirements do not depend on this software.</p>	
<p>Protocols used HTTP for unicast streaming, RTP for multicast streaming. The network must have multicast support in order to receive multicast traffic.</p>	
Development / porting effort, testing	
Deployment, monitoring (client, server, network, tools, resources...)	
<p>User community All the people who want to play with MP3 audio streams.</p>	

11. OpenH323

Activity: 5.1	Proposer: CTI
Application name: Video Distributor	
Overview OpenH323 platform is the result of OpenH323 project, which started in September 1998 by Equivalence Pty Ltd, a private company based in Australia (http://www.openh323.org) and has as target the development of an open source H.323 protocol stack	
Functional description – Benefit brought by IPv6 OpenH323 platform is an open source platform, which contains both clients and server, which can be used for H.323 videoconference. The clients that are available include: a command line H.323 videoconference application and a GUI H.323 videoconference application. The servers that are available include: a H.323 MCU (Multipoint Control Unit), a H.323 Gatekeeper and a H.323 to PSTN Gateway. The porting of OpenH323 platform in IPv6 network environments will result the first IPv6 compatible H.323 implementation. The porting of H.323 to IPv6 will enable the use of H.323 over native IPv6. In addition the H.323 will benefit from the advantages that IPv6 can offer to real time applications (like QoS support, security support, etc). Moreover, an IPv6 compatible H.323 implementation will encourage the use of H.323 videoconference to other platform (except the personal computer and set-top boxes) like PDA and mobile devices.	
Prerequisites Tested on Linux and Win32 platforms and compiled on Solaris, FreeBSD, and BeOS. Hardware required: a sound card for audio communication and camera for visual communication.	
Protocols – H.323 for videoconference – G.711, GSM MS-GSM and LPC-10 for audio encoding – G.723.1, G.728 and G.729 for audio encoding (with the use of appropriate hardware) – H.261 for video encoding; H.235 Annex D support for Gatekeeper access	
Development / porting effort, testing The current version of OpenH323 platform does not support IPv6 networks. As the main developers of OpenH323 platform mentioned: “The ways the system was designed was to make the support of different networks as simple as possible”. See http://crl.research.compaq.com/projects/mercury/ . As we have described, OpenH323 platform consists of a number of modules (endpoints, MCU, gatekeeper, etc). Based on the available resources some (one or more) of the OpenH323 modules will be ported to IPv6 networks. The ported OpenH323 modules will be tested among the participants of WP5 in order to validate their proper operation.	
Deployment, monitoring Based on the OpenH323 modules (endpoints, MCU, gatekeeper, etc) which are going to be ported in IPv6 networks, different deployment scenarios can be used like: point to point connection between two H.323 endpoints or multipoint connection among a number of H.323 endpoints with the use of H.323 MCU. The network demands of H.323 videoconference in terms of bandwidth are the following: – For endpoints (H.323 clients): In most of the cases 768Kbps is enough – For MCUs (H.323 server): In most of the cases n*768Kbps is enough.	
User community H.323 videoconference has already a big enough user community. For example the members of GRNet - VNOC (Virtual Network Operating Centre) in Greece, are using H.323 videoconference in order to arrange scheduled virtual meetings	

12. Robust Audio Tool - RAT

Activity: 5.1	Proposer: UCL
Application name: Robust Audio Tool (RAT)	
Overview The Robust Audio Tool (RAT) is an open-source audio conferencing and streaming application that allows users to participate in audio conferences over the Internet.	
Functional description – Benefit brought by IPv6 RAT requires no special features for point-to-point communication, just a network connection and a soundcard. For multiparty conferencing RAT uses IP multicast and therefore all participants must reside on a multicast-capable network. RAT is based on IETF standards, using Realtime Transport Protocol (RTP) [RFC1889] above UDP/IP as its transport protocol, and conforming to the RTP profile for audio and videoconference with minimal control. RAT features a range of different rate and quality codecs, receiver based loss concealment to mask packet losses, and sender based channel coding in the form of redundant audio transmission [RFC 2198]. RAT consists of two entities which are connected via the mbus: <ul style="list-style-type: none"> • The media engine – Which provides the bulk of the functionality, including all audio coding systems. • The User Interface – Which provides the GUI for user control over the media engine. 	
Prerequisites	
Protocols RTP, MBUS	
Development / porting effort, testing <ul style="list-style-type: none"> • RAT is implemented in C and TCL/TK [TCLTK] • Hardware required is a full-duplex soundcard, though RAT can be used for monitoring purposes without the presence of a soundcard. • Binaries are available for the following platforms: Linux, Windows 95/98/NT/2000, Solaris, and FreeBSD. Source code is available for compilation for other platforms: http://www-mice.cs.ucl.ac.uk/multimedia/software/rat <p>Installation is via a platform specific setup program</p>	
Deployment, monitoring	
User community	
Evaluation and future development <ul style="list-style-type: none"> • RAT is currently fairly stable, and provides good performance. • Further work is required on transcoding and support for interleaved formats 	

13. Video Conference Tool - VIC

Activity: 5.1	Proposer: UCL
Application name: Video Conference Tool (VIC)	
Overview VIC is an open-source video conferencing and streaming application that allows users to participate in video conferences over the Internet.	
Functional description – Benefit brought by IPv6 VIC requires no special features for receiving video from a session. To send video to a session a video capture device is required, which supports the platform specific capture libraries which include; Video4linux [v4l], Video for Windows, and Sunvideo. VIC is based on IETF standards, using RTP above UDP/IP as its transport protocol, and conforming to the RTP profile for audio and videoconference with minimal control. VIC features a range of different codecs (H.261, H.263, JPEG, H263, H263+, PVH, RAW (YUV), NV, cellb), which allow for the choice of quality and bandwidth employed. Vic provides support for layered video streams using the PVH codec. It now uses the UCL common library for Mbus operations, and cryptographic algorithms. Support for IPv6 from UCLA has been added. It also features application level symmetric encryption for private conferencing. VIC supports IPv6 operation for multicast and unicast use.	
Prerequisites	
Protocols RTP, MBUS	
Development / porting effort, testing VIC is implemented as a single threaded asynchronous application in C/C++ and Tcl/TK <ul style="list-style-type: none"> • VIC requires no additional hardware for receiving video from a session, though a capture card or USB camera is required for sending video. • Binaries are available for the following platforms: Linux, Windows 95/98/NT/2000, Solaris, and FreeBSD. Source code is available for compilation for other platforms: http://www-mice.cs.ucl.ac.uk/multimedia/software/vic Installation is via a platform specific setup program.	
Deployment, monitoring	
User community	
Evaluation and future development <ul style="list-style-type: none"> • VIC is currently fairly stable, and provides good performance. • Further work is required on use of direct video display and integration of more codecs. 	

14. Network Text Editor - NTE

Activity: 5.1	Proposer: UCL
Application name: Network Text Editor (NTE)	
Overview NTE is an open-source shared text editor. The collaborative text editing can be between two participants directly, or between a group of participants on a common multicast group.	
Functional description – Benefit brought by IPv6 NTE tries hard to ensure that the user does not get confused by unexpected events caused by other users - it always tells users who did what if it can. However, it cannot do the impossible, and sometimes network conditions may mean that a change arrives somewhat delayed. If this happens, NTE will reach a consistent result, but this may not be what any individual user expected. Thus we recommend using NTE as part of a multimedia conference in which it is a support tool, rather than as the only channel of communication. NTE supports IPv6 operation for multicast and unicast use..	
Prerequisites	
Protocols No standardised protocols are utilised.	
Development / porting effort, testing NTE is implemented as a single threaded asynchronous application in C and TCL/TK <ul style="list-style-type: none"> • NTE requires no additional hardware • Binaries are available for the following platforms: Linux, Windows 95/98/NT/2000, Solaris, and FreeBSD. Source code is available for compilation for other platforms: http://www-mice.cs.ucl.ac.uk/multimedia/software/nte Installation is via a platform specific setup program.	
Deployment, monitoring	
User community	
Evaluation and future development NTE is currently fairly stable, and provides good performance. Further work is required on transcoding and support for interleaved formats	

15. Whiteboard - WBD

Activity: 5.1	Proposer: UCL
Application name: Whiteboard (WBD)	
Overview WBD is an open-source shared whiteboard compatible with the LBL whiteboard, WB. The collaborative whiteboard activities can be between two participants directly, or between a group of participants on a common multicast group.	
Functional description – Benefit brought by IPv6 WBD provides a shared canvas that may be edited by a number of users at the same time. WBD provides facilities for drawing various shapes, and text, in a variety of different colours. External postscript files may also be imported into WBD for collaborative annotation. WBD utilises the Ghostscript engine for processing of postscript input. The drawing primitives, which represent the whiteboard state, are distributed between participants using an early version of Scalable Reliable Multicast (SRM) [SRM] protocol from LBNL. WBD supports IPv6 operation for multicast and unicast use.	
Prerequisites	
Protocols No standardised protocols are utilised.	
Development / porting effort, testing WBD is implemented as a single threaded asynchronous application in C and TCL/TK <ul style="list-style-type: none"> • WBD requires no additional hardware • Binaries are available for the following platforms: Linux, Windows 95/98/NT/2000, Solaris, and FreeBSD. Source code is available for compilation for other platforms: http://www-mice.cs.ucl.ac.uk/multimedia/software/wbd • Installation is via a platform specific setup program. 	
Deployment, monitoring	
User community	
Evaluation and future development <ul style="list-style-type: none"> • WBD is currently fairly stable, and provides good performance. 	

16. High-quality Audio Tools - HAT

Activity: 5.1	Proposer: UCL
Application name: HAT	
Overview	
The High Quality Audio Tool (HAT) provides for sending and receiving MP3 audio over Realtime Transport protocol (RTP) on IPv6.	
Functional description – Benefit brought by IPv6	
HAT uses the MP3 encoder, LAME to encode the MP3, which is taken packetised and sent out on RTP. For playback HAT retrieves the MP3 payload from the RTP packets and uses mpg123 to decode the MP3 stream.	
IPv6-enabled: Currently hat works on MSR IPv6 stack (We tested it on Microsoft Windows2000 with MSR IPv6 suite installed). In the near future (hopefully the end of Dec. 2001), another version which works on Microsoft Tech Preview IPv6 will be released.	
Multicast/unicast support: We support two types of communication: n-to-n (multicast) and 1-to-1 (unicast). When you're in a network with IPv6 multicast capability, you can exchange data via IPv6 multicast simply by typing in a multicast address in the address field of hat UI or joining an advertised hat session through sdr (To add hat as a plug-in to sdr , please refer to FAQ). Even if in a multicast-incapable network, you can communicate with one who has an IPv6 address by assigning your counterpart's unicast address to the address field.	
Bandwidth-adjustable: You can choose the targeted bandwidth among 32, 80, 128 kbps. VBR (variable bit rate) / CBR (constant bit rate) option is also provided.	
Traffic monitoring: Hat presents some traffic monitoring data per participant basis: the total number of bytes received, the proportion of lost packets, and the proportion of disordered packets.	
Prerequisites	
A machine running Windows 2000 with the Microsoft research IPv6 stack installed. Windows compatible audio card.	
Protocols	
RTP.	
Development / porting effort, testing	
It was designed for use with IPv6. HAT is available for binary download from: http://mmlab.snu.ac.kr/~hat	
Deployment, monitoring	
User community	
HAT has been tested from Korea to a number of sites including UCL.	

17. ISABEL CSCW

Activity: 5.1	Proposer: External Contributions via UCL
Application name: The Polytechnic University of Madrid ISABEL CSCW application	
Overview The ISABEL CSCW application is a group communication tool for the Internet, based on advanced videoconferencing features. Isabel allows efficient organisation of working procedures over the Internet in large enterprises or groups.	
Functional description – Benefit brought by IPv6 The architectural model of ISABEL is a set of media components, which are controlled by a management agent. The management agent implements the management policy of a given service. The media components manage the media flows and the media presentation at the connected sites. The media components are audio, video, shared workspace, VNC windows OS interconnection. The data traffic generated by ISABEL is generated by many independent and variable rate sources. The overall traffic send to the network by an ISABEL workstation is the aggregation off all the individual sources. A special network interface agent has been added to ISABEL, which aggregates the traffic coming from the various sources and which tries to adapt the traffic shape and rate to the requirements of the network service available. This network agent is called the ISABEL Flow Server. The roles and functions are very similar to the roles and functions performed by the multicast server needed by ISABEL and the flow server has been extended to support multicast server functions. Therefore the flow server is a key element, which has three fundamental roles in ISABEL: 1) Creation of gateways for interconnecting heterogeneous networks. 2) Adaptation of the multimedia flows to the quality of service provided by the network. 3) Creation of a multicast server which connects a large number of endpoints.	
Prerequisites	
Protocols: RTP.	
Development / porting effort, testing ISABEL is implemented in C and TCL/TK, and the latest version supporting operation over IPv6. ISABEL requires video and audio hardware for participation in conferences. ISABEL is available in binary form for linux from: http://isabel.dit.upm.es	
Deployment, monitoring	
User community	
Evaluation and future development ISABEL has been used in number of events and is being commercialised through: http://www.agoratechnologies.com . It is under active development within other research projects.	

18. Digital Video Transport System - DVTS

Activity: 5.1	Proposer: UCL
Application name: DVTS	
<p>Overview DVTS (Digital Video Transport System) is an application for sending and receiving DV (Digital Video) streams using the Internet. IEEE1394 (Firewire) cables are used for connecting DV devices. However, the length of a single IEEE1394 cables can not be longer than 4.5 meters. Using DVTS, DV data can be sent anywhere using the Internet.</p>	
<p>Functional description – Benefit brought by IPv6 The DV data is sent over the Internet using RTP (Real-Time Transportation Protocol). The isochronous stream packet of DV is encapsulated in RTP/UDP/IP, with audio and video in the one stream. The design and implementation is adaptable on the Internet regarding jitter and packet loss. DVTS also has ability to adapt to variety of network bandwidths. At the highest quality of communication, the system consumes over 35Mbps as network bandwidth, however the system can also adaptively change the bandwidth according to the end to end network conditions.</p>	
<p>Prerequisites Currently DVTS works on FreeBSD, Mac OS X, Linux, and Win2K/XP. An IEEE1394 (Firewire) interface on the machine is required, with a suitable DV source connected.</p>	
<p>Protocols RTP</p>	
<p>Development / porting effort, testing The source and binaries for DVTS on various platforms are available from: http://www.sfc.wide.ad.jp/DVTS</p>	
<p>Deployment, monitoring</p>	
<p>User community DVTS has been tested from Japan to a number of sites including UCL and USA.</p>	

19. Quadapt

Activity: 5.1	Proposer: TELIN
Application name: Quadapt	
Overview A heterogeneous mobile streaming platform that uses Mobile IP to offer streaming services to mobile clients. Research issue is the solution for mobility handoff. Two alternatives are investigated and compared against each other: a network level solution (Mobile IP) and an application level solution (VIC with extensions ie. SIP).	
Functional description – Benefit brought by IPv6 Mobile IP on IPv6 is used for the network layer solution	
Prerequisites	
- Client	
Hardware:	
Intel Pentium III/IV PC	
Software:	
MS Windows 2000	
Microsoft Research IPv6 stack	
Microsoft Research Mobile IP stack	
VIC videoconferencing tool	
(eventually Darwin/Quicktime streaming client)	
- Server	
Hardware:	
Intel Pentium III/IV PC	
Software:	
MS Windows 2000	
Microsoft Research IPv6 stack	
Microsoft Research Mobile IP stack	
Linux MIPL stack	
VIC video conferencing tool	
(eventually Darwin streaming server)	
- Network:	
Mobile IP must be supported; bandwidth must be high enough for streaming services up to 5 Mbit/s.	
Protocols	
Mobile IP, RTP, RTSP	
Development / porting effort, testing	
VIC must run on the IPv6 stack. Perhaps some additional porting is needed to run it on the Microsoft Technology Preview IPv6 stack for Windows 2000.	
Testing is done in a local laboratory.	
Deployment, monitoring	
User community	

20. Bonephone – SIP client

Activity: 5.1	Proposer: FhG
Application name: Bonephone	
Overview Bonephone is a SIP based internet phone that acts as a SIP user agent.	
Functional description – Benefit brought by IPv6 <ul style="list-style-type: none"> ▪ Bonephone is capable of sending and receiving SIP messages and reacting to them. ▪ It provides the user with a GUI to enable him to start, answer or terminate calls as well as maintain a phonebook with SIP addresses of possible callees and allows the user to establish and maintain parallel calls whereas it displays the status of the different calls. ▪ It integrates the audio engine needed to interact with the audio device of the system and allows the user to generate packetized audio and display incoming audio streams. 	
Prerequisites Operating system: <ul style="list-style-type: none"> ▪ Linux Programming language: <ul style="list-style-type: none"> ▪ C, C++ Required software: <ul style="list-style-type: none"> ▪ SUN jdk 1.4 (beta) ▪ MBUS⁵ package ▪ RAT⁶ tool ▪ NIST-SIP⁷ and SIP-UA ▪ Bonephone application containing the call control, media control and GUI software Network: No special requirements	
Protocols Session Initiation Protocol (SIP)	
Development / porting effort, testing Bonephone SIP UA (Phonebook, GUI) Bonephone software porting to IPv6 is required Development and testing is done at Fraunhofer FOKUS premises	
Deployment, monitoring Nearly all Linux based systems including PDA	
User community All	

⁵ <http://www.mbus.org/>

⁶ <http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/>

⁷ <http://www.antd.nist.gov/proj/iptel/>

21. RTP Quality Matrix - RQM

Activity: 5.1	Proposer: UCL
Application name: RTP Quality Matrix (RQM)	
Overview The Real-time Transport Protocol, RTP, provides quality of service feedback with reception reports sent alongside the media stream. If the media is sent via IP multicast it is possible for a third party to snoop on these reception reports, displaying reception quality for all members of a group. The RQM application performs such snooping.	
Functional description – Benefit brought by IPv6 When running RQM displays a matrix, in a window, with participant details on the left, and a number of cells to the right of these. Each row of cells denotes the packet loss rate observed for data sent from the participant indicated at the left of that row (point to a cell and a popup will appear giving the names of the source and destination of the traffic represented by that cell). The colours of the cell start as green (no loss) and fade to red (20% loss). A white cell indicates that no information is available. A light blue cell indicates that the receiver is not receiving media data from a particular sender (at present light blue is only used when an empty reception report is received, indicating that a receiver can hear no-one). Clicking on a cell will initiate an mtrace between the indicated participants, if you have mtrace installed and available in your path.	
Prerequisites	
Protocols: RTP.	
Development / porting effort, testing RQM is implemented as a single threaded asynchronous application in C and TCL/TK <ul style="list-style-type: none"> • RQM requires no additional hardware • Binaries and source code are available for the following platforms: Linux, Windows 95/98/NT/2000, Solaris, and FreeBSD: http://www-mice.cs.ucl.ac.uk/multimedia/software/rqm • Installation is via a platform specific setup program. 	
Deployment, monitoring	
User community	
Evaluation and future development <ul style="list-style-type: none"> • RQM is currently fairly stable, and provides good performance. 	

22. Multicast streaming tools - MUST

Activity: 5.1	Proposer: GRNET
Application name: Multicast streaming tools - MUST	
Overview Web-interface for simplified MBONE access using unicast or multicast connections	
Functional description – Benefit brought by IPv6 The toolkit is composed of two parts: <ul style="list-style-type: none"> - multicast enabled listener application that monitors MBONE announcements and stores session info - a cgi script for serving MBONE announcements over a web page to multicast or unicast enabled clients 	
Prerequisites clients : <ul style="list-style-type: none"> hardware : <ul style="list-style-type: none"> intel pentium 3 software : <ul style="list-style-type: none"> web browser & RTSP capable player (Linux,Win,Mac,Solaris,BSD) server : <ul style="list-style-type: none"> hardware : <ul style="list-style-type: none"> intel pentium 3 software : <ul style="list-style-type: none"> Linux Apache Darwin Streaming Server network :	
Protocols RTP/UDP, RTP/RTSP/TCP, HTTP, SDP	
Development / porting effort, testing <ul style="list-style-type: none"> – Development – Porting 	
Deployment, monitoring - Deployment and support for Greek Research Network community of 60 member institutions	
User community - tens of thousands of users	

23. Session Directory Tool - SDR

Activity: 5.1	Proposer: UCL
Application name: Session Directory Tool (SDR)	
Overview SDR is a session directory tool designed to allow the advertisement and joining of multicast conferences on the Mbone. It was originally modelled on <i>sd</i> written by Van Jacobson at LBNL, but implements a later version of the session description protocol than <i>sd</i> .	
Functional description – Benefit brought by IPv6 When SDR is running it lists all the announced sessions that are currently scheduled on the Mbone. SDR listens on the standard SAP announcement multicast address for SAP packets and displays the SDP sessions in the main window. SDR allows the user to join the sessions, where the relevant tools are automatically started up on the correct addresses/ports. For secure sessions SDR can be used to generate symmetric and asymmetric keys. Smart Cards can be used to encrypt personal information to be used for authentication and encryption. SDR also allows the user to make <i>quick calls</i> or multimedia “phone calls” to other SDR users, using a version of the SIP protocol.	
Prerequisites	
Protocols: SAP, SDP, SIP.	
Development / porting effort, testing SDR is implemented as a single threaded asynchronous application in C and TCL/TK <ul style="list-style-type: none"> • SDR requires no additional hardware • Binaries and source code are available for the following platforms: Linux, Windows 95/98/NT/2000, Solaris, and FreeBSD: http://www-mice.cs.ucl.ac.uk/multimedia/software/sdr • Installation is via a platform specific setup program. 	
Deployment, monitoring	
User community	
Evaluation and future development <ul style="list-style-type: none"> • SDR is currently fairly stable, and provides good performance. 	

24. Secure Conference Store - SCS

Activity: 5.1	Proposer: UCL
Application name: Secure Conference Store (SCS)	
Overview The UCL Secure Conference Store is a web-based system for secured creation, storage and access to conference information.	
Functional description – Benefit brought by IPv6 The system is mainly designed to manage secure conferences, but may be used to manage any conference. The server also maintains a cache of Session Announcement Protocol (SAP) [RFC2974] announcements, created by other announcement tools such as SDR, and allows public sessions to be created. Any user can view such public announcements and join the relevant sessions. The browser-server dialogue is protected by HTTP-S over IPv4 and IPv6. It can be used either with a simple user chosen "username" and "password" or with a user certificate stored in the browser. Mechanisms have been provided for managing groups of users. Each potential conference participant must register with the conference store by using a certificate. This certificate is verified in the normal way. The conference organiser can then set up an authorised group of participants by simple entries on the list of registered participants; the group managers can add/remove users, create/delete sessions and change keys. Group managers can also grant management capabilities to other group members. Only these authorised members of the group can access the web store containing the private session announcements pertaining to a particular group. Users have access only to the session information for groups to which they belong and to those pertaining to the public sessions; however, they may request to join a group that is not secret (i.e. one that they can see on the server). The server maintains a database of conference sessions, created using user-supplied details plus a random set of multicast addresses, ports and encryption keys. When setting up sessions, the server selects multicast addresses at random from a "GLOP" range [RFC2770]; these addresses are unique across all active sessions defined. Random, even numbered ports are selected, which are unique for each of the media defined for the session. The session encryption keys are random alphabetic strings - they can be regenerated at any time. The server automatically removes a session after the expiry time. When a user joins a session, a server script generates an HTML page referencing the SPAR Java applet and encodes the session details as Session Description Protocol (SDP) [RFC2327] data. The applet is run by the browser and parses the SDP to execute the media tools on the host with the correct addresses/ports/keys required for the session. The server can run on both IPv4 and IPv6 networks and manage sessions for either.	
Prerequisites	
Protocols SDP, SAP, SSL.	
Development / porting effort, testing SCS is implemented as a number of Perl scripts running on an IPv6 capable secured Apache web server. SCS requires no additional hardware. The secure conference store is accessible at: IPv6: http://www-secure.ip6.cs.ucl.ac.uk , IPv4: http://www-secure.cs.ucl.ac.uk .	
Deployment, monitoring	
User community	
Evaluation and future development SCS is currently fairly stable, and provides good performance.	

25. SDP Parser Applet - SPAR

Activity: 5.1	Proposer: UCL
Application name: SDP Parser Applet - SPAR	
<p>Overview</p> <p>New multicast capable multimedia applications, such as RealPlayer and Murrtech Pro, can accept files containing session descriptions (in SDP format) to join a multicast conference. This allows SDP files to be served from a web page and a conference joined, using a registered SDP MIME type associated with the application. Alternatively popular Mbone applications, such as VIC, RAT, WB and NTE rely on the Session Directory Tool (SDR) to convert the SDP into command line parameters and thus cannot take advantage of joining conferences via a web page. The solution offered here is to accept the SDP from the web browser, convert it to command line parameters at the client side and then to start up the tools with the relevant parameters. This could be implemented in several possible ways:</p> <ul style="list-style-type: none"> • a platform dependent binary running on the client machine with a MIME type for the SDP • a plug-in installed in the browser • a Java applet with execution privileges <p>The Java applet was implemented because it requires no installation or configuration by the user and is platform independent.</p>	
<p>Functional description – Benefit brought by IPv6</p> <p>The Java applet parses the SDP, extracts essential parameters and then starts the tools with the parameters on the client machine. The SDP content is embedded within the HTML as a parameter to the applet. The field terminator used in the SDP is replaced with a 'browser friendly' alternative, as SDP's CR/LF field terminator is removed by the browser. An added advantage of using HTTP to communicate the SDP content between client and server is that by using a secure web server, the SDP content will also be secure. For obvious security reasons standard Java applets do not have permissions to access local resources and thus cannot execute software on the client machine. To overcome this, both Netscape's Communicator and Microsoft's Internet Explorer 4 allow applets to be digitally signed with a private key associated to a RSA object-signing certificate. If the user accepts the certificate, therefore trusting the applet, then the browser allows the applet access permissions outside the Java security sandbox. Communicator and Internet Explorer implement different methods and technologies for digitally signing and distributing objects:</p> <p>Communicator requires Java applets to be signed using Netscape's Netscape Object Signing software.</p> <p>Internet Explorer requires Java applets to be signed using Microsoft's Authenticode software and packaged using a CAB file structure.</p> <p>Alternatively, browser's that do not support signed applets but do have a 'plug-in' architecture can use SUN's Java Plugin to view the applet. The Plugin requires the applet to be signed and packaged using Netscape's Object Signing software but it doesn't implement Netscape's Capabilities API. Since the API is not supported by the Plugin, any certificate accepted by the user, grants universal access to local system resources. SPAR supports IPv6 operation.</p>	
Prerequisites	
Protocols SAP.	
<p>Development / porting effort, testing</p> <p>SPAR is implemented as a Java Applet. SPAR requires no additional hardware.</p> <p>SPAR is implemented as part of the Secure Conference Store:</p> <p>IPv6: http://www-secure.ip6.cs.ucl.ac.uk IPv4: http://www-secure.cs.ucl.ac.uk.</p>	
Deployment, monitoring	
User community	
<p>Evaluation and future development:</p> <p>SPAR is currently stable, and provides good performance.</p>	

26. Quake

Activity: 5.2	Proposer: Sony
Application name: Quake	
Overview A First Person Shooter action game from id Software, whose game engine has been made public available.	
Functional description – Benefit brought by IPv6 Quake's story revolves around the "slipgate," a device the US army has developed to instantaneously transport people anywhere. Of course, an evil fellow who calls himself Quake gets hold of a slipgate and is sending his troops to wreak havoc on Earth. The US army launches "Operation Counterstrike," of which the player is part of, to seek out Quake and annihilate him. The player must hop into a slipgate and head for Quake's home world in a desperate attempt to save the planet. Of course, this is not as easy as it sounds... To solve this task the game player has to steer his character through a virtual world full of dangers and secrets and to fight his way to find and destroy Quake. When playing in multiplayer mode, the different players steer their characters through the same virtual world. They can play in teams or against each other.	
Prerequisites A PC running FreeBSD 3.4/KAME, Solaris 8 for SPARC, Windows 2000 and NT 4 using MSRIIPv6 ver 1.4. At least 32 MB of RAM and an Internet connection. Users must have the .wad files from the original game (buy Quake I).	
Protocols Game specific	
Development / porting effort, testing Setup of game server and clients. Testing of the game play using different setups. Testing Ipv4-Ipv6 interworking (game client in IPv4 network and server using IPv6 or vice versa) using a NAT/PT approach.	
Deployment, monitoring Setup of an own game server.	
User community Has to be identified.	

27. Experimental Gaming Platform - EGP

Activity: 5.2	Proposer: Sony
Application name: EGP (Experimental Gaming Platform)	
Overview Development of a prototypical gaming platform for demonstrating IPv6 features using online games.	
Functional description – Benefit brought by IPv6 The gaming platform will focus on the networking aspects (not on graphics etc.). We target to have at least two simple games with different network requirements using the platform. The games may be open-source games that are modified to use the platform. Some of the aspects we want to investigate: <ul style="list-style-type: none"> • using SIP for setting up gaming sessions • QoS (e.g. using different Diffserv classes) • usage of multicast • enhancing the game play with the possibility to communicate with other players (audio/video) 	
Prerequisites Java 1.4 (on a IPv6 platform supported by Java)	
Protocols game specific, http, rtp/rtcp, rtsp, sip	
Development / porting effort, testing Development of the platform (as a basis we use an already existing internal multimedia platform named AMUSE). Porting or developing of games for the platform. Testing of the games in different setups.	
Deployment, monitoring Internal testing. Tests with other WP5 partners.	
User community The EGP will be used for research purposes. We are not targeting to use this platform in a “field trial” or having a larger user community for it.	

28. XPilot

Activity: 5.2	Proposer: Invenia
Application name: XPilot	
Overview Network based game developed at the University of Tromso. The server will be natively reachable from both IPv4 and IPv6 hosts. Local network discovery will be done gracefully through IPv6 multicast.	
Functional description – Benefit brought by IPv6 Our expectations are to extend our knowledge about multicasting and real-time traffic in IPv6 networks. Also, our Artic Beans project (http://abean.cs.uit.no) is based on modern networking (IPv6), which require first hand knowledge about IPv6 based middleware. The server will be natively reachable from both IPv4 and IPv6 hosts. Local network discovery will be done gracefully through IPv6 multicast.	
Prerequisites Both server and client will be tested on NetBSD and Linux..	
Protocols XPilot uses UDP for communication	
Development / porting effort, testing Initially we plan to port the game "XPilot" to IPv6. We will then attempt to extend or redesign the game by using multicast, or by adding IP telephone communication. If there is an opportunity, we would also like to test the game on a Playstation running Linux. In addition, we are not part of 5.3, but we have extensive middleware knowledge. The XPilot server and client will be ported to support both IPv4 and IPv6. Broadcast local server discovery algorithm will be replaced by IPv6 multicast. We also want to test XPilot, as an interactive application with need for low latency, in a mobile environment.	
Deployment, monitoring A server will be available at Invenia Innovation AS, and at any interested 6NET partners. Clients can connect from anywhere.	
User community Mainly students.	

29. MUD gaming environment

Activity: 5.2	Proposer: UoS
Application name: MUD (lpmud)	
Overview A text-based multi-player gaming environment, combat oriented.	
Functional description – Benefit brought by IPv6 Server can be hosted on any host, e.g. on home network with ADSL. Main MUD model is client server, but “server” not restricted to non-NAT locations when IPv6 is used.	
Prerequisites Expect to run IPv6 MUD server code on Linux. Client is plain telnet client.	
Protocols All runs over telnet.	
Development / porting effort, testing Can investigate enhanced MUD services where server can open connection directly back to player’s host, given IPv6 addressability (some game developers have found ways to “work around” NATs, but we believe good value-add services may be possible through IPv6). UoS has a student-run MUD. We will offer free MUD hosting for the students on condition the MUD is IPv6-enabled. The student group (Cslib) can specify additional features to be developed. We expect to use (and port) lpmud, as a good example of an object oriented MUD scripting language. The lpmud code could open connections to other ports/services, e.g. retrieve information via HTTP (and possibly SOAP).	
Deployment, monitoring Local to UoS, can connect from anywhere.	
User community Mainly students. Server could be designed to take IPv4 and IPv6 connections to the same server.	

30. IBM Websphere Portal Technology

Activity: 5.3	Proposer: IBM
Application name: Portal	
Overview A portal infrastructure that will provide a single access point for users with the capability to enable e-commerce facilities in the 6net academic and research environment	
Functional description – Benefit brought by IPv6 The portal application will use web service over Ipv6 to create a more open collaborative environment	
Prerequisites Hardware : IBM xServers Software: Linux distribution Websphere Portal Server Websphere Edge Server Websphere Commerce Suite	
Protocols HTTP,XML,WSDL,SOAP	
Development / porting effort, testing A portal customized structure is to be created for each interested community of users	
Deployment, monitoring	
User community	

31. GLOBUS 2.0

Activity: 5.3	Proposer: IBM
Application name: GLOBUS 2.0	
Overview Open sources Grid middleware. Exact component inventory TBD.	
Functional description – Benefit brought by IPv6 Offers Grid Security Infrastructure, Grid Resource Manager and related APIs to IPv6 client applications. Ideally, these will be able to invoke existing Grid resources in the IPv4 world.	
Prerequisites IPv6 stack on Linux.	
Protocols Globus runs over SSL/TCP and supplies GridFTP. GLOBUS 3.0 will also use SOAP over HTTP.	
Development / porting effort, testing ?	
Deployment, monitoring TBD	
User community Potentially, “big science” users (physics, bioinformatics) already involved in existing projects such as the European Data Grid and willing to try IPv6 client access.	
Evaluation and future development Possible later migration to GLOBUS 3.0/ OGSA.	

32. Agent Framework – SoFAR, SLITE

Activity: 5.3	Proposer: UoS
Application name: Agent Framework (SoFAR, SLITE)	
Overview	
<p>The Southampton Framework for Agent Research (SoFAR) is a Java framework used primarily for RMI communication. A “light” version, aimed at use of multicast for service discovery, has recently been developed (SLITE).</p>	
Functional description – Benefit brought by IPv6	
<p>The framework provides a registry so that agents can advertise their services and others can find them, and it supports several communication patterns including queries and a publish-subscribe model. The framework is being used to explore the application of software agents to multimedia systems.</p> <p>The IPv6 advantage lies in the addressability of multiple SoFAR/SLITE devices, that can communicate directly, peer to peer.</p>	
Prerequisites	
Linux or Solaris, Linux preferred, running at least Sun JDK 1.4.	
Protocols	
An example SoFAR application has been developed to broker and control RTP streams (including some functionality also seen in SDR). Interaction with other protocols is expected in other SoFAR or SLITE-based applications.	
Development / porting effort, testing	
<p>Currently testing Sun JDK 1.4.1 under Sun’s CAP program.</p> <p>No advanced Java IPv6 API yet. This should be pressed for in the lifetime of 6NET.</p> <p>A functional clone of rtpttrans has been ported to IPv6 to enable RTP stream relaying – we should investigate porting the full version of rtpttrans.</p>	
Deployment, monitoring	
<p>Within UoS.</p> <p>Framework available to all partners.</p>	
User community	
<p>UoS has a strong agent-based computing community in our department’s IAM group. Existing and future SoFAR applications will be tested under IPv6, and the main SoFAR code made IPv6-aware. (Note this raises the question of which protocol stack is used for dual-stack hosts. Java includes a preference method).</p>	

33. Hypermedia Link Services

Activity: 5.3	Proposer: UoS
Application name: Hypermedia Link Services	
Overview	
Hypermedia link services are a key component of many of the multimedia applications developed in the research lab at Southampton.	
Functional description – Benefit brought by IPv6	
A simple link server accepts a query from a client and returns a list of available links. The query could, for example, be a location in a temporal media stream. The key performance factor in link services is latency, particularly if the link services uses referrals or query routing or if there are synchronization requirements with temporal media.	
As an output of 6WINIT, in order to experiment with a IPv6-enabled link service, two servers were ported:	
<ul style="list-style-type: none"> • DLS1 is a specialised service with an HTTP interface. It is a standalone reference implementation in use by current research projects. • DLS2 is based on an LDAP directory service and provides a distributed implementation. 	
DLS1 is a ‘context sensitive’ link service that makes use of information about the device making the query, as part of a pervasive computing infrastructure. Hence in addition to porting of the networking code, we are also introducing IPv6 addresses as part of the context handling mechanism.	
DLS2, based on work in collaboration with BT, has recently been interfaced with the agent framework	
Prerequisites	
Demonstrators run on Linux. OpenLDAP is available for IPv6. Apache (web server) is available for IPv6.	
Protocols	
UoS’ DLS (distributed link service). LDAP HTTP	
Development / porting effort, testing	
Hypermedia link services are an interesting area for IPv6, given link servers can be distributed, and temporal media may be involved (as per UoS’ work on HyStream with BT). We expect to continue to develop IPv6-enabled link services. DLS2 uses Fundamental Open Hypermedia Model (FOHM). We have MIDI streaming tools available for IPv6 (client and server).	
Deployment, monitoring	
Local to UoS, but accessible anywhere.	
User community	
UoS’ IAM group makes heavy use of hypermedia link services and the DLS architecture. Making these IPv6-ready will enable IPv6 to pervade into other research projects.	

34. Funnel Web

Activity: 5.3	Proposer: UCL
Application name: FunnelWeb	
Overview FunnelWeb is a system that runs on a node to provide an active services platform. FunnelWeb is an implementation of an Application Level Active Networking (ALAN) [ALAN] active networking execution environment (EE).	
Functional description – Benefit brought by IPv6 Specifically FunnelWeb provides an execution environment for java based active applications, known as proxylets. The FunnelWeb EE is termed the Execution Environment for Proxylets (EEP), which provides a java environment with a Remote Method Invocation (RMI) control interface for loading, running, modifying operation and stopping proxylets.	
Prerequisites	
Protocols RMI.	
Development / porting effort, testing FunnelWeb is implemented as a Java application. <ul style="list-style-type: none"> • FunnelWeb requires no additional hardware • FunnelWeb is available on request within the project 	
Deployment, monitoring	
User community	
Evaluation and future development The FunnelWeb is being extended for use with XML based events and policies. Security functionality is being extended to fully utilise certification services and transport level security.	

35. Transcoding Active Gateway - TAG

Activity: 5.3	Proposer: UCL
Application name: Transcoding Active Gateway -TAG	
Overview The Transcoding Active Gateway (TAG) was developed to extend the functionality of an earlier tool, known as the UCL Transcoding Gateway (UTG). The implementation was based on the FunnelWeb [ALAN] Active Networking architecture.	
Functional description – Benefit brought by IPv6 TAG builds upon Funnel Web to provide its functionality. The TAG client application is separated into two components that communicate using Remote Method Invocation (RMI): the Funnel Web EEP component of the client runs the Routing, Discovery and local Reflector proxylets and the Routing and Discovery proxylets are used by the client to identify its location in relation to other parts of the Active Network. The user interface component of the client is used to communicate both with the EEP component and with a remote EEP via the RMI interface. The server configuration section of the user-interface allows the user to query the local Routing proxylet for information regarding the current EEP available and the closest EEP in relation to the local host. Once an EEP has been selected the controls for starting, stopping and configuring media streams are enabled. TAG has recently been updated to provide an additional mode that allows it to connect a client into a conference VPN and subsequently provide access to the VPN media streams. Using this mode, a client will automatically request to join the VPN once local multicast activity is detected. The join request takes the form of a registration of a local EEP to a Reflector Manager (RM). The registration triggers the RM to notify each node within the VPN to forward media streams back to the registered EEP. In addition the RM can be used to convey rate and access information to each remote node.	
Prerequisites	
Protocols RTP, XML	
Development / porting effort, testing TAG is implemented in Java. JDK1.4 is required if TAG is used in an IPv6 environment. <ul style="list-style-type: none"> • TAG requires no additional hardware • The initial release is available for download to 6WINIT partners: ftp://6winit@cs.ucl.ac.uk/6winit/ftp/incoming/wp6/tag-1.1.zip • The installation of TAG is documented on the web site: http://www-mice.cs.ucl.ac.uk/multimedia/software/tag/ 	
Deployment, monitoring	
User community	
Evaluation and future development A number of extensions are required to the TAG to add functionality to support issues such as security and mobility. These extensions will be explored over the next few months. Some, concerned with the active networks aspects, will be carried out under ANDROID; some, which are required for the 6WINIT wireless environment, will be implemented under 6WINIT. These extensions are listed below: <ul style="list-style-type: none"> • Media conversion (transcoding) and rate control in the Reflector Proxylet • Quality of service improvements to the reflection • Improvements to the discovery and location of Active Servers/EEPs • EEP access control to prevent unauthorised user access. • Services offered by an EEP • Support for Foreign Agent functionality • Secure communications and authentication of client • XML messaging throughout design • Policy control of Reflectors 	

36. TZI Stargate

Activity: 5.3	Proposer: External Contribution via UCL
Application name: TZI StarGate	
Overview StarGate provides call signalling and media transcoding gateway functionality for connectivity between different kinds of endpoints interconnected through different types of networks (hence the name *Gate)..	
Functional description – Benefit brought by IPv6 StarGate is conceptually built upon the same general Mbus architecture as AudioGate [AG], with largely different Mbus entities and different interactions between them, of course. AudioGate provides a dial-in point for users on any telephone network and enables them to participate in Mbone audio conferences. All three of the aforementioned control protocol entities share a core set of Mbus messages to set up, tear down, and monitor progress of a call. In addition, each entity supports protocol-specific Mbus extensions that may not be (easily) mapped to other control protocols. The Mbus controller is expected to understand all these Mbus commands, route incoming messages, and optionally perform translation between different protocols. Call control messages are intended for interaction with call control and invitation protocols such as H.323 and SIP. They are designed to constitute the union of the call control messaging needed by endpoints, gateways, proxies, multi-point controllers, and gatekeepers. This allows the use of the Message Bus to act as gluing mechanism to create any type of system from roughly the same building blocks. Mbus call control messages are based on a common basic message set defined in the following that will be supported by any kind of call control protocol entity. The basic message set may be augmented by protocol-specific extensions required for protocol specific interactions between a local controller and/or local applications on one side and the respective protocol engine on the other. While the basic Call Control commands have been worked through, they still need to be mapped to H.323, SIP, and ISDN-specific messages.	
Prerequisites	
Protocols: H.323, SIP, ISDN, RTP, MBUS	
Development / porting effort, testing StarGate currently does not support IPv6 but porting work is being considered. Stargate is implemented in C/C++ and java on Linux StarGate requires a Linux PC, and an ISDN basic rate interface board. The ISDN-card must be supported by the HiSax-driver, because it is the only driver which supports <i>audio via ISDN</i> . AudioGate was developed using a TELES S0 16.3 ISDN board. The ISDN HiSax driver is required to be version 3.3 or above (from ftp://ftp.suse.com/pub/isdn4linux). The kernel needs to be configured to support <i>audio via ISDN</i> and <i>IP: multicast</i> .	
Deployment, monitoring	
User community	
Evaluation and future development A possible future extension for Mbus command is a set for the Real-Time Streaming Protocol (RTSP) extensions.	

37. LightWeight Directory Access Protocol - OpenLDAP

Activity: 5.3	Proposer: UNINETT
Application name: OpenLDAP (Lightweight Directory Access Protocol)	
Overview OpenLDAP is used in many middleware applications and it's important that it is available over IPv6. OpenLDAP Software (http://www.openldap.org/) is an open source implementation of the Lightweight Directory Access Protocol (v3), base protocol is specified in RFC 2251.	
Functional description – Benefit brought by IPv6 OpenLDAP is a simplification of the X.500 DAP.	
Prerequisites	
Protocols LDAP (v3), RFC 2251 etc.	
Development / porting effort, testing We have ported OpenLDAP to IPv6 and this is in the official 2.x versions. We want to get OpenLDAP tested on a wide variety of platforms, and make any necessary changes to make IPv6 work on those. Some changes have already been necessary, since IPv6 stack implementations do things in slightly different ways. We will also see if we can use OpenLDAP as a proxy to give IPv4 clients access to IPv6 LDAP servers and vice versa. A number of applications in A5.3 depend on LDAP, and should obviously use it over IPv6. Part of the work will be to assist them with any LDAP IPv6 problems.	
Deployment, monitoring OpenLDAP is widely deployed and since binaries by default support IPv6, there are already many installations that support IPv6. This includes UNINETT's own LDAP server, and we expect many 6NET participants to also install it.	
User community There are probably tens of thousands of OpenLDAP users, and OpenLDAP will by default support IPv6. As operating systems and networks are IPv6 enabled, the number of people using OpenLDAP over IPv6 will grow.	

38. Public key infrastructure - PKI

Activity: 5.3	Proposer: External Contribution via UCL
Application name: The University of Murcia Public Key Infrastructure	
<p>Overview</p> <p>The purpose of a Public Key Infrastructure (PKI) is to define the mechanisms and elements needed to manage and enable the effective use of public key encryption technology on a medium or large scale. The base components are a certification authority, one (or several) registration authorities, and a directory server. Some additional components, like smart cards, time stamping servers, OCSP servers, can be present depending on the services offered by a particular PKI implementation.</p>	
<p>Functional description – Benefit brought by IPv6</p> <p>The Public Key Infrastructure of the University of Murcia is based on the design and implementation of IPv6-enabled X.509 certification services. It can be used by an organisation to provide its users with a range of public key mechanisms for securing their communications.</p> <ul style="list-style-type: none"> • It allows certificates to be requested, renewed and revoked for every entity (end user or process) of an organisation. • It allows the use of an LDAP directory to store the users and Certificate Authority (CA) certificates and Certificate Revocation Lists (CRL). • Final users can carry out a variety of certification operations from their own web browser or through the Registration Authority (RA). • Users can use smart cards to store cryptographic information (private key, certificate and CA's certificate). This allows mobility and increases the security of the system. • It supports the definition of a Certification Policy that will establish the restrictions inside an organisation. This policy is defined by the administrator and is applied in every PKI component (registration authority, certification authority, request server, etc.). • It is completely developed in Java, allowing the use of any operating system to run an implementation of the PKI. • It is based on those drafts and standards specified by the IETF inside its PKIX working group. • It supports the Simple Certificate Enrolment Protocol (SCEP), enabling router certificate requests. • It supports the Online Certificate Status Protocol (OCSP). • Time Stamping is implemented in the system. • The end user interface for the system is IPv6 enabled, e.g. the LDAP server and web server, so final users can access the system using this network protocol. <p>Work is underway by the University of Murcia to convert internal communication to use IPv6.</p>	
Prerequisites	
Protocols SCEP, OCSP, X.509, PKCS#11	
<p>Development / porting effort, testing</p> <p>The PKI is implemented as a number of components written in C and java, including the OpenLDAP server, Postgress SQL server, Apache with Jserv and SSL running a number of java servlet processes on Linux.</p> <p>The PKI requires no additional hardware</p> <p>The PKI is available on request within the project from: http://www.um.es/atica/pki</p>	
Deployment, monitoring	
User community	
<p>Evaluation and future development</p> <p>The PKI is being run at number of sites including UCL. It is being extended for use with attribute certificates.</p>	

39. Edge Server Proxy

Activity: 5.4	Proposer: IBM
Application name: Edge Server Caching Proxy	
Overview Forward or reverse caching proxy intercepts page requests from a client, retrieves with its own IP address the requested information from content-hosting machines, and delivers that content back to the client. The proxy stores cacheable content to satisfy the following requests to the same content, which improves response time and network resources.	
Functional description – Benefit brought by IPv6 - Forward caching proxy: it is placed at the edge of the client’s network. It intercepts the client’s request, asks a DNS server to resolve the web server’s IP address and performs the request to it. As it receives the content, the proxy delivers it to the client. This system allows the client to be anonymous by mapping his address with the proxy’s one. It also preserves external bandwidth when cached pages are requested. - Reverse caching proxy: it is placed at the edge of the server’s network. It intercepts a request from internet, asks the server for the corresponding content, then delivers it to the internet client. This is transparent for the client. The server is hidden by the proxy, providing anonymity and reducing server access due to the caching. The IPv6-enabled Edge Server Proxy receives IPv4 and IPv6 requests, as well as reaching IPv4 and IPv6 servers.	
Prerequisites Hardware: Intel Pentium 3 / 4 512 MB RAM Software: Linux Red Hat 7.0 2.4.2 kernel with IPv6 built GSKit 5 (for SSL sessions) Original IBM proxy v 2.0 IPv6 IBM proxy (currently based on IBM proxy v 1.1)	
Protocols IPv4, IPv6, HTTP, DNS	
Development / porting effort, testing A proxy prototype based on IBM Proxy v1.1 has been basically tested in IPv4-IPv6 environments. It currently doesn’t perform reverse proxy yet. Added functionalities are under development, and a version based on IBM Proxy v5.0 will be released. We also wish to add some edge services like security, load balancing, or QoS.	
Deployment, monitoring The proxy architecture should be installed in UK and Netherlands PoPs.	
User community Proxy services are already widely spread. Main users are ISPs, Campuses, content distribution and e-business networks.	

40. Contents Delivery Networking - CDN

Activity: 5.4	Proposer: Cisco, IBM
Application name: CDN (Content Distribution Network)	
Overview Content Distribution networks scale and accelerate content services by distribution content at the edge of the network and redirecting client request to the most appropriate edge server by means of a content routing process	
Functional description – Benefit brought by IPv6 A CDN consists of a content distribution management function responsible for optimizing the distribution of content to the edge of the network, a content routing process to redirect client requests to the closest edge delivery node whereby closest is related to a metric based on RTT and content availability and the edge delivery node serving the content using different protocols like HTTP or RTSP. The CDN infrastructure can be used to accelerate and scale other application services developed in WP5	
Prerequisites Cisco CDN solution, IBM origin server	
Protocols HTTP, HTTPS, RTP/RTCP, RTSP	
Development / porting effort, testing Depending on product availability the CDN might be deployed in a v4 or v6 domain. Details still need to be worked out	
Deployment, monitoring	
User community All 6net user accessing static content and streamed video	