UCL Transcoding Gateway (UTG)

Server v1.2a

User Guide

Version 2 (DRAFT)

30 November 1998

Acknowledgements

Version 1 by Jin Yang (original)

Version 2 by Kristian Hasler (update)

Disclaimer

Copyright note

Contents

Figures

1 Overview

This document desribes the functions of the server components of the UCL Transcoding Gateway (UTG) which relays traffic from the Mbone to unicast connected clients. The UTG – Umbone relay - consists of five relay agents. Firstly there is the SDR relay to provide session information to a subscribed host, secondly the audio relay which provides audio connectivity and transcoding facilities, thirdly the video gateway which provides video connectivity with rate control and some transcoding facilities. Finally the other two relay agents, provide for shared text and whiteboarding, which are essentially 'blind' and just drop packets to acheive the desired data rate.

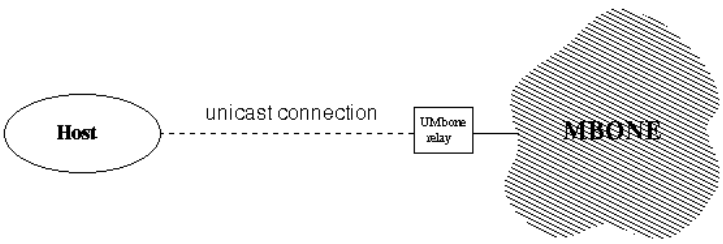

Figure 1: Schematic of client host to Mbone connection

2 Major components

2.1 Multicast traffic relay

To send/receive multicast traffic several relay agents have been developed. Upon request these agents will transform and relay the relevant packets between the multicast-capable server side and its clients, which are not multicast capable. The typical network configuration for applying the UMbone relay tools is illustrated in Figure 2.

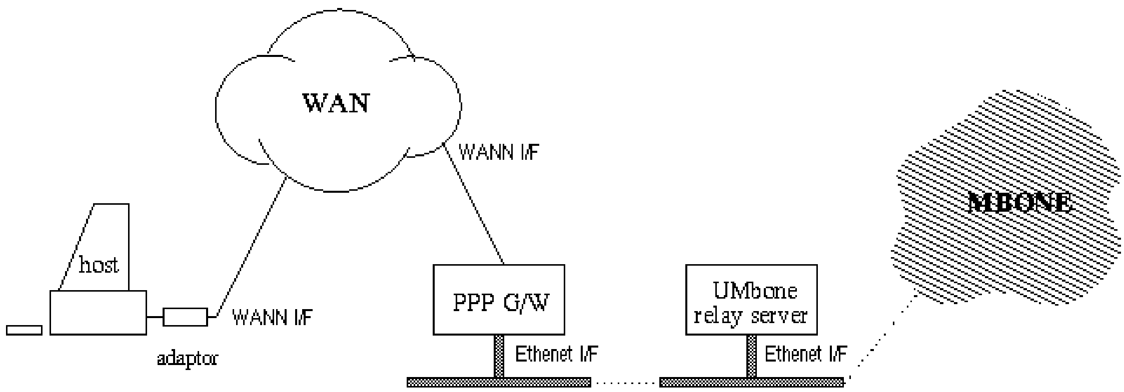

Figure 2: Network configuration

To join a multicast session, the unicast host needs to have a point-to-point IP connection to the Internet, e.g. via a PPP gateway. The relay agents runs on a host, called UMbone relay server, which is connected to Mbone. The relay server is not necessarily on the same local area network as the PPP gateway, although the performance will be improved if it is.

To make the Mbone tools, such as *SDR* [17], *VIC* [3], *RAT* [9] and *VAT* [8], available on the unicast host, the relay agents also need to perform other tasks e.g. transcoding. The overall data flow is illustrated in Figure 3. The functionality additional to the multicast traffic relay, of each relay agent is discussed in the following sections.

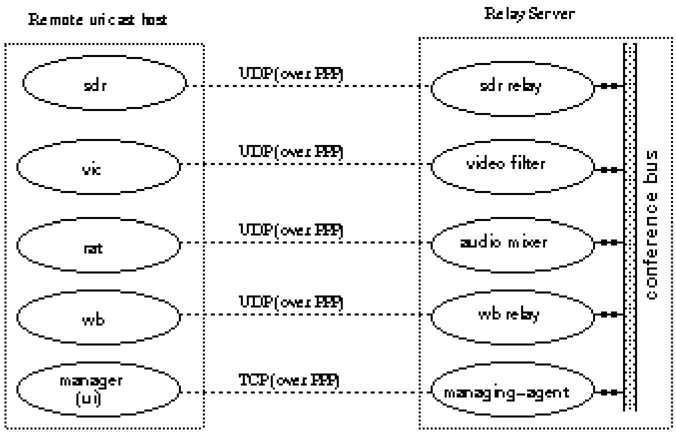

Figure 3: Data flow of media tools and their relay servers.

2.2 SDR via relay

The session directory tool *SDR* serves two functions in the current framework: first it listens to and displays session announcements; second it allows user to invoke corresponding media tools to join a session. To follow this setting without modifying *SDR* code, the following components have been added.

A SDR relay agent has been developed which listens to session and passes those announcements to a remote unicast host. The SDR relay agent also passes cached session announcements to the unicast client on startup so that it can receive the most up-to-date announcement quickly.

Additional plug-ins are provided on the unicast side so that corresponding media tools can be invoked in special modes. With these plug-ins, the interfaces to start media tools are shown in Figure 4.

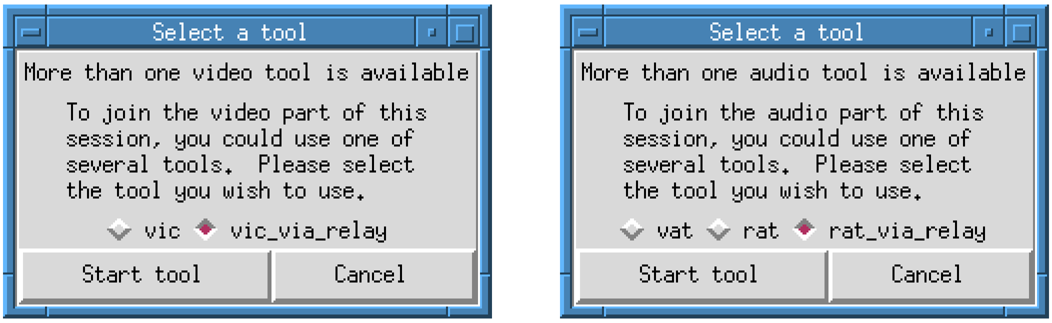

Figure 4: Interface to start tools via relay.

2.3 Video transcoder and rate-adapter

When a user joins a video session from SDR, a request for video filtering is sent to the relay server before video tool VIC is started locally. Upon receiving this request, the relay server starts a video filtering engine on a multicast-capable host. The video filtering engine is adopted from the “video gateway engine” developed by Amir et al.[3] and it performs two major tasks:

- Collecting session participants' information and accepting instructions from interface. The engine listens to a multicast session which is of interest to its client and cashes all senders' information, their CNAMEs and SSRCs, stream bit rate etc. On demand it provides latest senders' information to the user interface that query it. This info allows the user to choose video streams which are of interest. It also listens on the conference bus to receive control instructions from the clients.

- Manipulating and forwarding RTP and RTCP packets between Mbone session and unicast site. For RTP packets, two operations are applicable: passing through or filtering by reducing frame-rate or transcoding (JPEG to H261, NV to H261) (see Figure 5). In both cases, the modified or unmodified RTP packet is immediately forwarded to the other part. For RTCP packets, sender and receiver reports are modified before forwarding to reflect the distribution quality as seen by the engine.

For multiple unicast clients, an easy solution to serve them simultaneously is to run multiple copies of the video engine. This is inefficient if the clients are joining the same video session with similar transcoding requirements, in which case the transcoding needs to be performed only once and then adapting the frame-rate separately for each client.

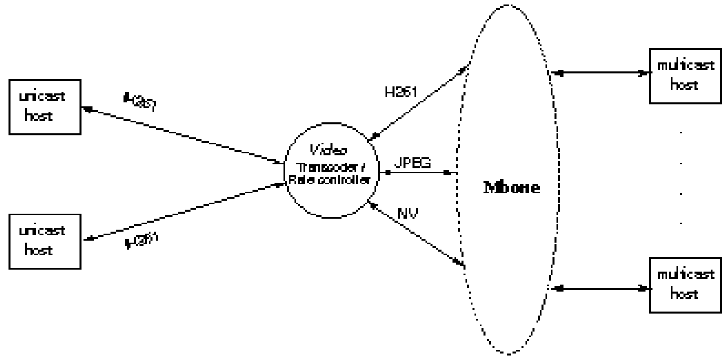

Figure 5: Video transcoder and rate-adapter.

2.4 Audio Transcoder/Mixer

When a user joins an audio session from SDR, as in the case for video tool, a request for audio relay is sent to the relay server before the audio tool is started locally. Upon receiving this request, the server starts an audio transcoder/mixer which has been developed as part of the UCL Robust-Audio Tool (RAT version 3.0.22) [21] under the MERCI project [10]. It performs two functions: it mixes the audio from a number of participants on the Mbone to produce a single output stream, and optionally transcodes that stream to a format which is more bandwidth-efficient. The transcoder operates between a number of formats, including PCM, DVI, GSM and LPC [11,12] and can convert from any one of these formats to any other. In addition, packet-loss repair is performed on the input stream, including the decoding of redundant audio [22] and receiver based repair such as packet repetition, etc. This operation is illustrated in Figure 6.

Similar to the video case, multiple copies of the transcoder/mixer can be run to serve multiple clients simultaneously.

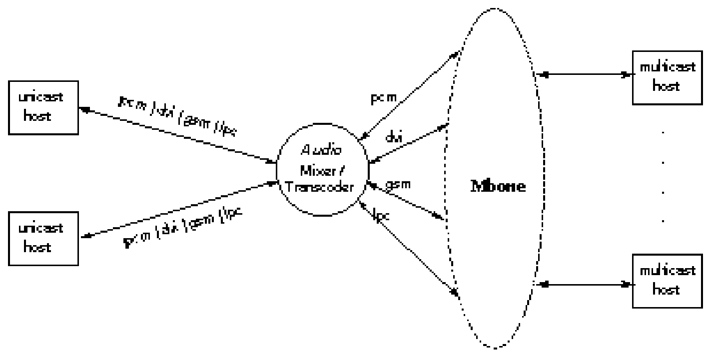

Figure 6: Audio Transcoder/Mixer Operation

2.5 Remote monitoring and control of UMbone relay

The relay agents (audio transcoder/mixer and video filter) need to be controlled by users on the unicast site. To serve this purpose, some information, e.g. the participants of the video session including their names and transmitting rate, need to be collected/monitored continuously. From the management viewpoint, the status of the relay agents is also required. We therefore constructed the relay agents in such a way that the interface of each agent is divided into two parts: an operational interface and a TCP management interface (see Figure 7). The operational interface allows agents to perform the task of relaying RTP/RTCP packets and transcoding or filtering, while the management interface allows control/monitoring task be carried out. GUIs are provided so that users can control the relay agents via their management interfaces.

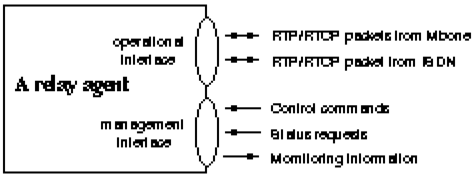

Figure 7: The two-parts interface for a relay-agent.

3 System Requirements

- Solaris2.X with multicast kernel, 4Mb of disc space and 32Mb of memory (minimum).

  (multicast kernal available from: http://playground.sun.com/pub/multicast/Solaris_mc35.2.5.1.tar.Z)

- Connection to the Mbone.

4 Installation

The package can be downloaded from:

http://www-mice.cs.ucl.ac.uk/multimedia/software/

First unzip and untar the file: utg-server-1.2a-solaris.tar.gz:

```
gunzip utg-server-1.2a-solaris.tar.gz

tar xfv utg-server-1.2a-solaris.tar
```

Seven files will be extracted into the directory utg-1.2-server:

- server-installation - this file.
- audio_relay - the audio relay agent
- video_relay - the video relay agent
- blind_relay - the whitebaord and text-editor relay agent
- sdr_relay - the sdr relay agent
- vgw - the video transcoder/rate-adapter engine
- rat - the audio mixer/transcoder engine

5 Running the server

Use the start script provided or (eg using sh):

```
cd /path-name/utg-1.2-server
PATH=$PATH:.
export PATH
audio_relay&
video_relay&
blind_relay 6016&
blind_relay 6018&
sdr_relay&
```

These agents will write run-time messages using syslog(), so if you want to see the messages you will need to edit your syslog.conf file. e.g.

```
user.debug     /var/adm/messages
```

6 Control messages

The messages used in the ISDN G/W belong to the session dependent messages, and they loosely fall into the Tool Messages, Activity Messages, Participant Messages and Media Control Messages. For easy understanding they are grouped into the following sets:

6.1 Tool Initialisation Messages:

The following messages are used to invoke remote tools:

- audio-relay-gui/pid all/all INVITE audio-mixer
- video-relay-gui/pid all/all INVITE video-filter
- sdr-client/pid all/all INVITE sdr-relay-agent

The following messages are response from invoked tools:

- audio-mixer/pid all/all ALIVE
- video/filter/pid all/all ALIVE
- sdr-relay-agent/pid all/all ALIVE

The following messages are used to initialise the relay tools:

- audio-relay-gui/pid audio-mixer/pid SET-ADDR-A multicast-addr/port/ttl
- audio-relay-gui/pid audio-mixer/pid SET-ADDR-B unicast-addr/port/ttl
- video-relay-gui/pid video-filter/pid SET-ADDR-A multicast-addr/port/ttl
- video-relay-gui/pid video-filter/pid SET-ADDR-B unicast-addr/port/ttl
- sdr-client/pid sdr-relay-agent/pid SET-ADDR-B unicast-addr/port/ttl

6.2 Tool Control Messages:

To control the behaviour of relay tools:

- audio-relay-gui/pid audio-mixer/pid SET-ENCODER [dvi|lpc|gsm|pcm]
- video-relay-gui/pid video-filter/pid SET-RATE-CONTROL [0|1]
- video-relay-gui/pid video-filter/pid SET-TOTAL-BW bandwidth
- audio-relay-gui/pid audio-mixer/pid PAUSE

- audio-relay-gui/pid audio-mixer/pid QUIT
- video-relay-gui/pid video-filter/pid PAUSE
- video-relay-gui/pid video-filter/pid QUIT

6.3 Participant Message:

To report a new participant detected by the audio mixer or video-filter, or update info for a specific participant:

- audio-mixer/pid audio-relay-gui/pid ACTIVATE participant-id
- video-filter/pid video-relay-gui/pid ACTIVATE participant-id
- audio-mixer/pid audio-relay-gui/pid SRCUPDATE participant-id name source/encoder new-encoder
- video-filter/pid video-relay-gui/pid SRCUPDATE participant-id name source/encoder rate-control-mode original-bps new-bps original-f/s new-f/s participant-id

The following messages are used to enquiry info or set control for each participant in the video session:

- video-relay-gui/pid video-filter/pid SRCUPDATE participant-id
- video-relay-gui/pid video-filter/pid SET-MAXBPS participant-id max-bps
- audio-mixer/pid all/all ACTIVE-SPEAKER participant-id

7 Known Bugs

- The video-relay server process sometimes hangs/loops (probably if the client does not close down correctly) - the server process that is looping (running continuously) needs to be killed before new connections will work.
- When the client has not set the host address field the server sends traffic to address 0.0.0.0 which is taken as a local broadcast address and effects the network badly!

- Sometimes the server sends multicast traffic from src address 127.0.0.1 (rather than its IP address) - 127.0.0.1 is a "fake" loopback address and should never go out on the net.

- There is some limitation with audio and video when 2 or more clients use the UTG - the clients cannot see each other.

8 Acknowledgements

We would like to acknowledge the help from the following:

- Elan Amir *et al*. (UC Berkeley) for allowing use and modification of their *vgw* code
- John Andrews and Nigel Pugh (UCL) for their help with the local configuration and installation of the gateway components here at UCL
- Jon Crowcroft for his message about the UMbone idea
- The partners of the MERCI project, for their technical advice and assistance in testing these tools.

The authors acknowledge the financial support of the European Commission MERCI project (Telematics for Research #1007) and the UK EPSRC project RAT (GR/K72780).

9 References

Jon Crowcroft, Ian Wakeman, Mark Handley, and Stuart Clayman, *Internetworking Multimedia*, UCL Press,1996.

Elan Amir and Steve MaCanne and Hui Zhang, *An Application-level Video Gateway*, Proceedings of ACM Multimedia95, 1995,

Steve MaCanne and Van Jacobson, *vic: a flexible framework for packet video*, Proceedings of ACM Multimedia95, 1995,

Steve McCanne, *Scalable Compression and transmission of Internet Multicast Video*, PhD thesis, University of California, Berkeley , December 1996.

Mark Handley, Ian Wakeman and Jon Crowcroft, *The Conference Control Channel Protocol (CCCP): A scalable base for building conference control applications*, Proceedings of ACM SIGCOMM 95, Cambridge, MA, 1995.

Peter Parnes, Kare Synnes and Dick Schefstrom, *mTunnel: a multicast tunneling system with a user based Quality-of-Service model*. Proceedings of European Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services, Darmstadt, Germany, 1997.

The Mbone information web, URL:http://www.Mbone.com/

Jacobson V. *VAT manual pages*, Lawrence Berkeley Laboratory (LBL) February 1992.

Vicky Hardman, Angela Sasse, Mark Handley and Anna Watson*, Reliable Audio for Use over the Internet*, in Proceedings of INET'95, June 1995, Honolulu, Hawaii.

More information about the MERCI project can be found at http://www-mice.cs.ucl.ac.uk/merci/

Papamichalis P.E. *Practical Approaches to speech Coding,* Publ. Prentice-Hall 1987

Gold B. *Digital Speech Networks' Proceedings of the IEEE,* Vol. 65, No. 12, December 1977.

Ascend Products Overview, URL:http://www.ascend.com.au/products/

Teles Corp, URL:http://www.teles.com/

Steve Dusse, Paul Hoffman et al, *S/MIME Message Specification*, Work in progress, October 1997.

Peter Kirstein, Goli Montasser-Kohsari and Edmund Whelan*, Specification of Security in SAP Using Public Key Algorithms*, IETF MMUSIC working group, Work in progress, 28 October 1997 .

Mark Handley, *Session Directory Tool*, README file, University College London (UCL), 1996. http://north.east.isi.edu/~mjh/sdr/

MATES project, *mDesk: A multicast collaboration tool,* Centre for Distance-spanning Technology, Luleå University of Technology, Sweden. http://mates.cdt.luth.se/software/mDesk/

J. Highfield, *WBD: A WB compatible whiteboard*, Loughborough University, 1997.

Steve Deering, *Multicast Routing in a Datagram Internetwork,* PhD Thesis, Stanford University, December 1991.

The UCL Robust-Audio Tool. http://www-mice.cs.ucl.ac.uk/mice/rat/

C. Perkins, I. Kouvelas, O. Hodson, V. Hardman, M. Handley, J.-C. Bolot, A. Vega-Garcia and S. Fosse-Parisis, *RTP Payload for Redundant Audio Data*, IETF Audio/Video Transport working group, RFC 2198, September 1997.