Accessing MBone Sessions over N-ISDN
(Merci project technical report, draft)

Jin Yang
Computer Science Department, UCL, UK
Email: j.yang@cs.ucl.ac.uk

August 21, 1997

1 Introduction

Two issues have to be addressed to provide conferencing connectivity to users over N-ISDN or other low speed circuits.

The first issue is concerned about the heterogeneity which is intrinsic to the Internet technology. Due to the mismatch of link speeds, the bandwidth available between different sender-receiver is different. In existing applications, each sender transmits signal at the same rate to all receivers. With heterogeneous receivers, the source would have to run at a rate that matches the most constrained receiver. When N-ISDN is used, the available bandwidth is limited to 64 Kbps (1 B-channel) or 128 Kbps (2 B-channel). Currently most MBone transmissions use a target rate of 128 Kbps for video and 64 Kbps for radio, precluding simple bridging of MBone sessions across 128 Kbps ISDN line. Thus some kind of rate control - reducing rate by transcoding or reducing frame-rate (for video) - becomes necessary.

Another concern is about network configuration. Multicast routing is not yet ubiquitous, and so there are sites that wish to participate in MBone sessions but cannot do so. This applies in particular to workstations situated at the end of ISDN lines running IP (usually PPP) over ISDN. Such workstations often do not wish to run full Multicast routing, or cannot do so if they are running any Microsoft operating system without a local Multicast router, and so an alternative solution is required.

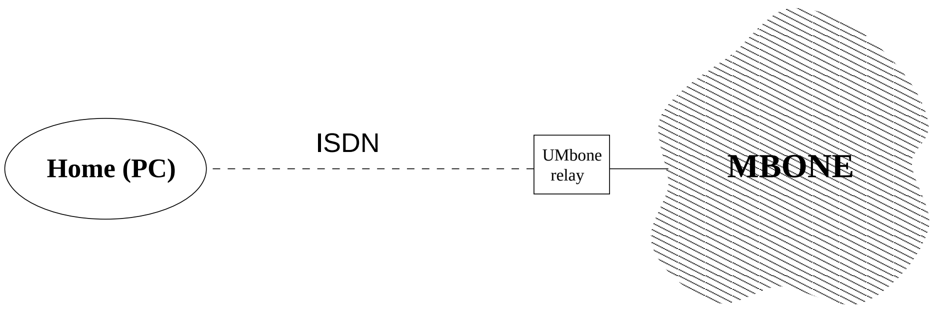

Figure 1: Accessing MBone session over N-ISDN line

One possible solution is based on the idea of UMbone (Unicast MBone) which was proposed by Crowcroft et al.[2] at UCL. It suggested a set of components that can be put together to form a relay between the multicast world and a single, multicast-deprived, end-system. Such a scenario is depicted in Figure 1.

Following the UMbone idea, we developed a set of tools which enable users accessing a MBone session (currently audio and video sessions). The functionalities and constructions of these tools, together with their interface and operation instructions, are described in Section 2. Some future work is mentioned in Section 5.

1.1 Related work

Application Level Video Gateway. One approach for rate adaption is to adjust the transmission rate from within the network. In this way, heterogeneity is "locally" accommodated by fine-tuning the transmission rate to exactly match the available capacity on each link in network. Amir et al. developed an application level video gateway[1], called *vgw*, which, being placed throughout the network, can transcode video streams into a lower bit rate coding, using either the same coding format with different parameters or a different coding scheme. the video gateway engine in this application software is adopted to carry out the video stream filtering work in our system. For our purpose of accessing MBone over ISDN, a separate control interface is developed to control the engine remotely.

Layered Coding. In this model, rather than distribute a single level of quality using a single network channel, the source distribute multiple levels of quality simultaneously across multiple network channels. In turn, each receiver individually tunes its reception rate by adjusting the number of layers that it receives. For more details see [4].

2 Major components

2.1 Network configuration

The network configuration used in the UCL-CS MERCI activity is illustrated in Figure 2.

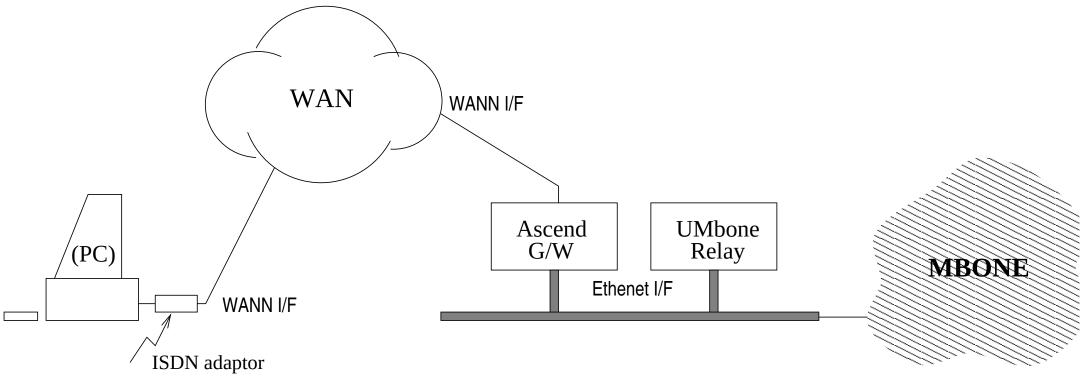

Figure 2: Network configuration

The Windows 95 Dial-Up Network subsystem allows remote/mobile users to designate a computer running Window 95 as a dial-up client. More than one connection can be defined to connect different remote access server. The current configuration for the dial-up network consists of the following components:

- ISDN adapter. A ISDN adapter from Teles, the TELES.BRI/PnP-Box, is used in our configuration. It is an external unit that connects a PC to the ISDN line through the parallel port. Teles provides a WAN-NDIS miniport driver for it which is used to support upper layer functionality. The Teles.WAN-NDIS miniport is an interface allowing use of the Windows 95 Dial-up Network and the various application such as Microsoft's ISDN Accelerator Pack.
- ISDN Accelerator Pack. This software from Microsoft supports multi-link which allows the bonding of two B channels for 128 Kbps performance. It works over the TELES WAN-NDIS miniport driver.
- PPP link to Ascend. The dial-up network is configured to dial the Ascend box which works as the remote access server. This allows Windows 95 PCs to connect automatically to the Ascend box which is running TCP/IP over PPP. The Ascend supports multi-link PPP, thus, together with the ISDN Accelerator Pack at the PC end, two B-channels can be bound providing a 128 Kp/s PPP link.

2.2 Operating Dial-up network

1. Start dial-up network.

   There are two methods to start dialling to the Ascend after the dial-up network is configured:

   (a) Windows 95 starts Dial-Up Network whenever an application attempts to access a network resource. With this property, invoking of media tools will start the dialling process automatically.

   (b) Alternatively, the Dial-up Network can be started explicitly by double-clicking the defined connection (named as `Ascend`) in the My Computer/Dial-Up Network window.

2. End and re-start a connection.

   To terminate a connection explicitly, double-click the "dial-up network icon" at the right end of the task bar and then click the menu `terminate` (Trennen in German).

   To re-establish a connection, the dial-up network need to be started either explicitly or implicitly as instructed above.

3. Monitor the link.

   Several things are helpful to monitor the link over ISDN line:

   - The ISDN adapter indicator: located at right side of the task bar, it contains two lights showing the current status of two B channels: **green** light indicates the disconnection (not in use) of the corresponding channel, while **red** light shows that the corresponding channel is being used. When two lights are in red, it means that both B channels are used.
   - The tools for tracing and displaying statistics of packets through the ISDN interface can be started by double-clicking the ISDN adapter indicator, and then a window will pop up containing a `tools` menu.

- The traffic over the dial-up network is also shown in the pop up window by double-clicking the "dial-up network icon"located at the right end of task bar.

2.3 Multimedia tools made available by relay

Currently the multimedia tools `sdr`, `vic`, `rat` and `vat` can be used over ISDN link. To receive session announcement and video/audio packets, corresponding relay servers are launched on the server host when required. The server not only passes the relevant packets between the multicast-capable server side and its clients side which is not connected to MBone, it also performs other tasks such as audio mixing and video frame rate control. The relay for these tools are briefly discussed in the following. The overall data flow is illustrated in Figure 3. The additional user interface which is necessary for using these tools is introduced in the next section.

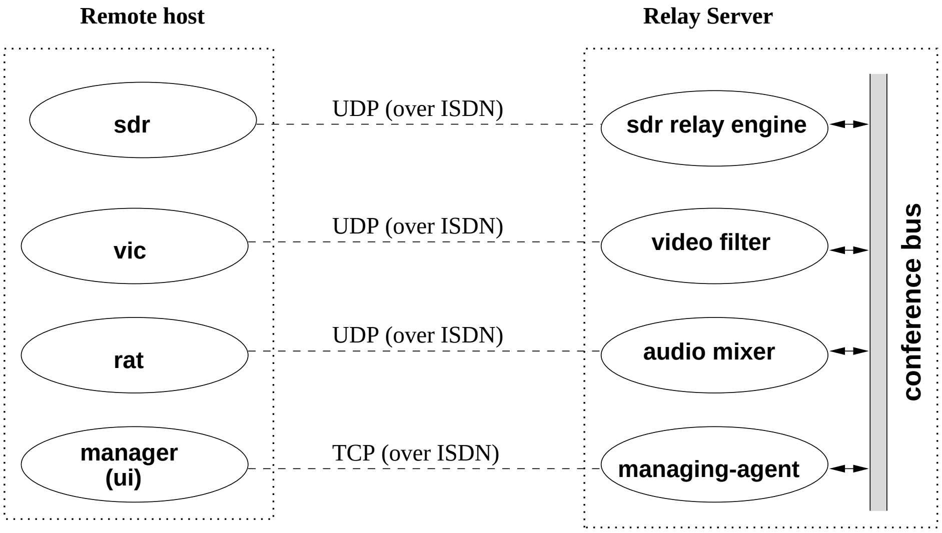

Figure 3: Data flow of media tools and their relay servers.

2.3.1 `SDR` via relay

The session directory tool `sdr` serves two kind of jobs in the current framework: firstly it listens to and display session announcements, and then it allows user to invoke corresponding media tools to join a session. To follow this setting without modifying `sdr` code, a few things have been provided to enable `sdr` on the PC side:

- a sdr relay server is developed which listens session announcement from well known multicast address and passes SAP packets to remote PC. The sdr relay server also passes cached session announcements to the PC so that it can receive the most up-to-date announcement quickly.
- On the PC side, the relayed SAP packets are sent to the well known multicast address for session announcement, which are then picked by the locally executed `sdr`.

- additional plugins are provided so that corresponding media tools can be invoked in special modes. With these plugins, the interfaces to start media tools are shown in Figure 4 and Figure 5.

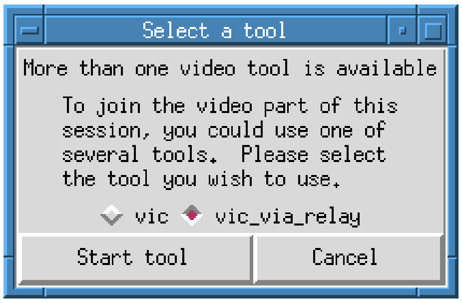

Figure 4: Interface to start `vic` via relay

Figure 5: Interface to start `rat` via relay

2.3.2 Video filter

When a user joins a video session from `sdr`, an invitation for video filtering is sent to the relay server before video tool `vic` is started locally. Upon receiving the request of audio relay, the server start a video filtering engine on a a multicast-capable host. the video filtering engine is adopted from the "video gateway engine" developed by Amir et al.[1] 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 allow user to choose video streams which are of interest to him/her. It also listens on the conference bus to receive control instructions from 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 6). 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.

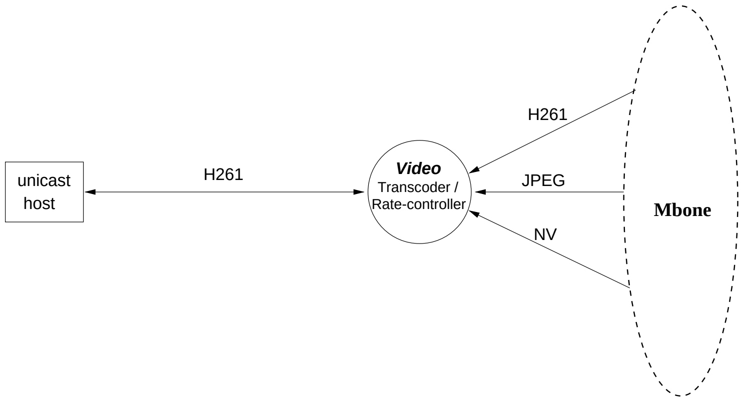

Figure 6: Video transcoder/filter.

2.3.3 Audio mixer

When a user joins a audio session from `sdr`, as in the case for video tool, an invitation for audio relay is sent to the relay server before the audio tool `rat` or `vic` is started locally. Upon receiving the request of audio relay, the server start a audio mixer which is developed by Colin. It works as an RTP transcoder/mixer in which RTP packets received from a multicast group are transcoded into a format which can be user specified, multiple sources are mixed together, and the result is transmitted to the other group/site (see Figure 7)..

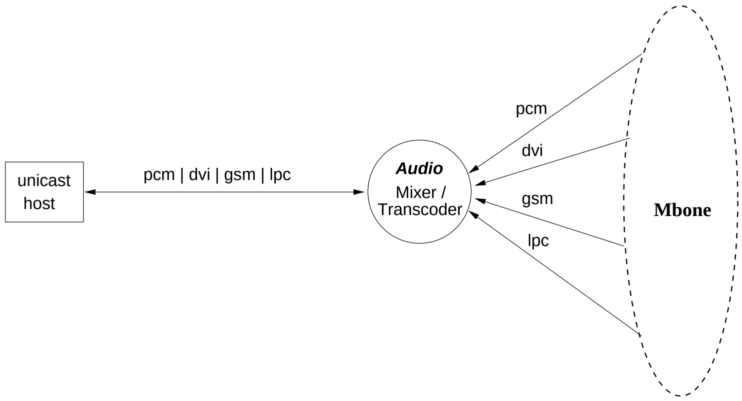

Figure 7: Audio mixer/transcoder.

3 Remote monitoring and control of UMbone relay

The relay agents (audio mixer and video filter) need to be controlled by users behind ISDN link. 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 management point of view, the status of these relay agents, e.g. their users, multicast sessions addresses and the PPP link status, would be also required for managers. Thus we constructed the agents in such a way that the interface of each agent is divided into two parts: operational interface and management interface (see Figure 8). The operational interface allow agents 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. The major user interfaces are described in the following sections.

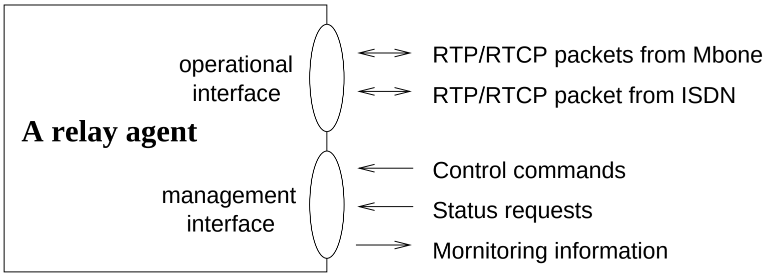

Figure 8: The two-parts interface for a relay-agent

3.1 The top-level user interface

When a user starts `vic_via_relay` or `rat_via_relay` from `sdr` on the PC, a user interface for audio and video relay control will be presented which is show in Figure 9. This interface can also be started stand alone without using `sdr`.

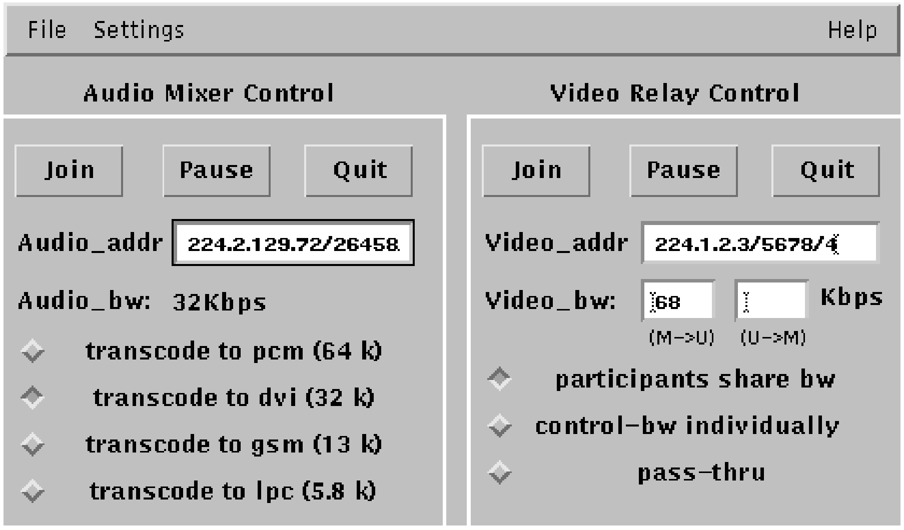

Figure 9: Interface to start `rat` via relay

The top-level user interface for the gateway is shown in Figure 9. It is divided into two parts, one is for controlling the audio mixer and the other one is for video filter. The following control items are common to both parts:

- **Join** will initialise audio mixer or video filter on remote server and start a local media tool accordingly. Note that the tool, `vic`, `rat` or `vat`, is running in a point-to-point mode (PC to remote server).

- **Pause** will cause the audio mixer or video filter to stop forwarding packets. Another press of this button will resume it. *(not implemented yet.)*

- **Quit** will cause quitting from current audio or video session. After that the user can join another audio or video session.

- **Session address and ttl** to appear in `Audio_addr` or `Video_addr` field. This field specify the multicast group address (an ip address with a port) in which users wish to join. User also can specify the `ttl` value in this field which will influence how far the packets sourcing from the unicast host will reachinto Mbone. For example:

  `224.2.172.238/51482/127`

  will set `ttl=127`. The default ttl value is `64`. There are several ways to get this field filled:

  1. when user join a session from sdr interface, the corresponding session address and required ttl will be filled in automatically.
  2. the address can be specified through command-line arguments.
  3. the default session addresses can be set in the Configuration menu (see below) so that they will appear in the address fields each time the interface is up.
  4. it can be filled in by typing into the address at any time.

The lower sections of the two parts are used to control the bandwidth allocation and relay behaviour, i.e. transcoding of audio mixer and filtering of the video filter. They are introduced Section 3.2 and 3.3.

3.1.1 The configuration menu

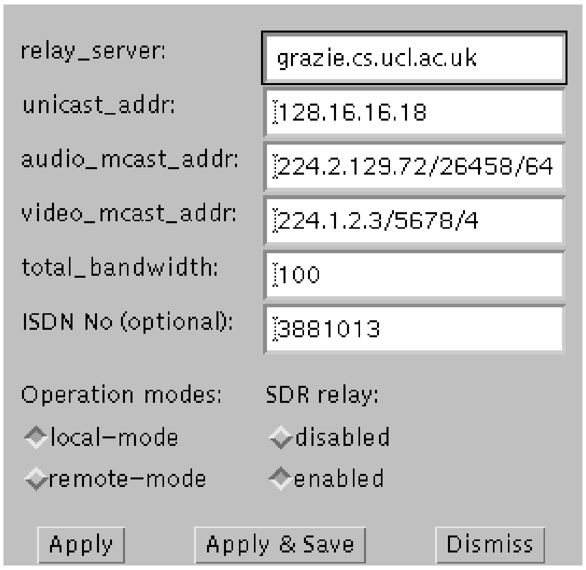

Figure 10: The configuration menu

The configuration menu, as shown in Figure 10, can be reached via top bar menu Settings. The fields in this menu are explained below:

- `Relay_server` specifies the host name (or dotted ip address) of the relay server.
- `Unicast_addr` specifies the host name (or dotted ip address) of the unicast site, which is the destination of relay from Mbone. This is usually the local host address running the user interface. Note that for host with muiltiple network interfaces, e.g. one for Ethenet, another one for ISDN, this field has to be filled in.
- Audio_mcast_addr and `Video_mcast_addr` are used to specify the default multicast session addresses for audio and video respectively. `Total_bandwidth` indicates the total bandwidth available for the audio *and* video sessions in the direction of Mbone $\rightarrow$ unicast relay. The transmitting rate for the gateway server to feed video stream packets from unicast host to Mbone is set in the top-level interface (Video_bw filed (`U` $\rightarrow$ `M`)) and is to be explained later.
- ISDN number is a optional to provide, for the purpose of management only.
- Operation modes. There are two modes: `local` or `remote`. Local mode means the multimedia tools are running locally at the same host as the interface. Remote mode enable to request/control relay service on behalf of another (unicast) host which can be completely unware of the relay behind, as illustrated in Figure 11. In this case the field `Unicast_addr` is the address for the remote host.
- SDR relay checkbox allows to enable or disable the sdr relay service.

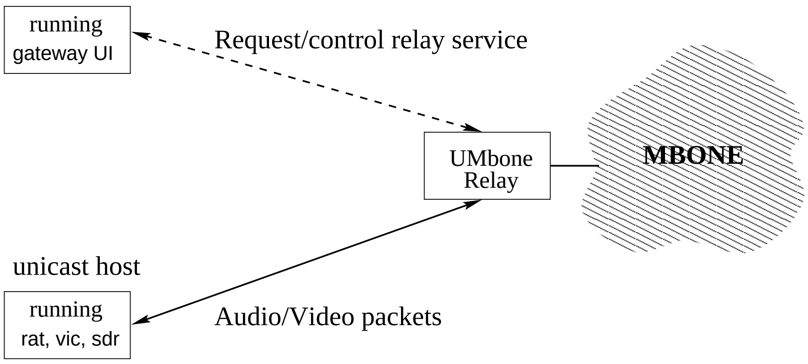

Figure 11: Request and control relay for another unicast host.

3.2 Audio transcoding control

Currently four encoding/decoding methods are supported by the audio mixer/transcoder. Each encoding transmits audio stream in different bit rate. Approximately the bit rates required are: PCM - 64 Kbps, DVI - 32 Kbps, `sf GSM` - 13.2 Kbps and LPC - 5.8 Kbps. User choose one from these four encoding methods and the corresponding bandwidth for audio session is displayed. The default encoding is DVI.

The relay is bi-directional and in each direction there are two separate stages (see Figure 12. Users have control over codings for the following three stages:

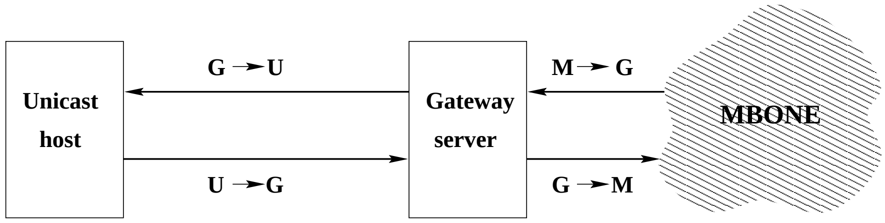

Figure 12: Bi-directional relay.

- **G** → **U**: the gateway relay packets from mbone to unicast destination. The coding method for this process is chosen in the user interface. The default one is DVI.
- **U** → **G**: packets from unicast host reach the gateway. The coding is, as default, same as the one used for **M** → **U** coding. If the session is launched via sdr, then the required coding is used. User can change this via rat interface.
- **G** → **M**: packets sourced from unicast host are feed in the mbone. The default coding is DVI, or set to the one required by sdr session information if launched via sdr.

3.3 Video Filtering control

Similar to the control over audio mixer/transcoder, users have control over the bit rates for the following stages:

- **G** → **U**: the video streams are relayed from mbone to the unicast destination. The bandwidth allocated for these video streams is specified in the `Video_bw (M → U)` field. There are several possible ways to use this bandwidth among the video streams, or users can ask the gateway to pass through all video streams without rate reduction. These control are discussed in Section 3.3.1.
- **U** → **G**: an user *must* control the transmittiong rate via the vic interface if he/she want to transmit video strteam to the gateway (and then to mbone) over a low-speed unicast connection.
- **G** → **M**: when the user's video stream is feed to mbone by the gateway, user can control the bit rate from gateway to mbone. This is useful when feeding high bandwidth video stream into mbone. In other cases, user can ask the gateway to simply pass the packets to mbone without rate reduction. If users want to control the transmit rate for the **G** → **M** relay then they need to set the field of `Video_bw (U → M)`. Otherwise just leave this field blank.

3.3.1 Video gateway control

.

For G → U direction of relay, there are three modes for rate control of video streams (listed in the video part of the top-level user interface):

1. All participants share the bandwidth allocated. This is the default mode in which the bandwidth is evenly allocated to each video stream in the current session. For example

if the bandwidth for video is 60 Kbps and there are three participants transmitting video streams, each stream will be filtered so that its bit rate will not exceed 20 Kbps.

2. Control rate individually. This setting means that the bit rate for each stream is to be set separately. To set bit rate for a specific stream, a further user interface will be presented which is described in below.

3. Pass through . Choosing **pass-thru** means no any rate control measure is to be taken. All video streams from the mbone are forwarded to the unicast site by the gateway without modification.

To control the bite rate for each video stream, a further user interface is provided which is shown in Figure 13.

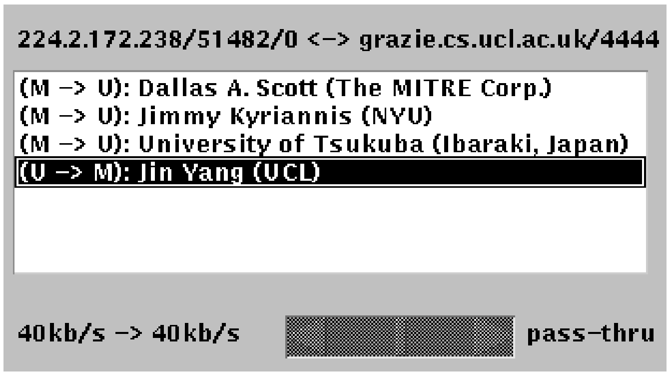

Figure 13: GUI for remote control of video filter.

The main section of this interface is a window showing all active senders' information in the current session. All senders' names (their CNAMEs) and their relay directions are listed in the window. To check the bit rate of a stream, double-click the sender's name and the corresponding bit rate is printed beneath the window.

When using the mode of "control rate individually", users can allocate bandwidth for a specific stream. Double click the sender in the senders window and set its bandwidth using the scrollbar under the main window. The allocated bandwidth for each stream is checked against the total bandwidth allocated for the video session.

3.4 Local conference bus

The local conference bus was proposed by McCanne et al.[5] and was used by the LBL media tools to provide a mechanism for coordination among separate tools. It was also used in *vgw*. The idea is further deployed in SCP[3]. The conference bus is used in *vgw* for communication between vgw engine and its interface. The same structure is followed in our system: the video filtering engine and audio relay engine interact with their control interface through conference bus (see Figure 14). Currently the conference bus is used only for message exchange between audio/video engines and their control interfaces. It can be further deployed to coordinate different engines (e.g. audio and video engines) or control multiple copies of same engine (e.g. 3 video engines for 3 clients).

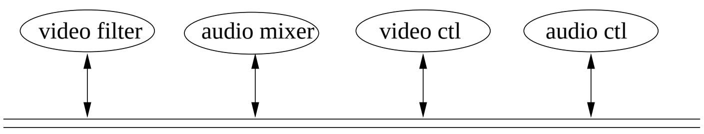

Figure 14: Using conference bus for local coordination.

4 Some performance measures

Some tests have been carried out to test the workload on both client machines (PC) and server (Sun Ultra-SPARC). The test environment is as below:

- *client machine 1:*
  HP Vectra XA Desktop, CPU: Pentium 200, 32M RAM
  audio card: Creative Lab Sound Blaster 16 Plug and Play,
  video capture card: Hauppauge WinCast/TV

- *client machine 2:*
  HP Ominibook Laptop, CPU: Pentium 133, RAM ??
  audio card: (built-in)
  video capture card: QuickCam (more details ?)

- *the server machine:*
  Ultra-Sparc I or II (64M memory)

- *Mbone sessions to join*
  audio session: one stream, DVI coding.
  video session: two video streams, each 100-150Kbps, 1-10f/s.

Workloads on client machines:

Operations	CPU usage(%) - P200	CPU usage(%) - P133
non (no A/V operation)	1-2	2-3
receiving 2 video streams	4-8	27-30
receiving 2 video streams receiving 1 audio stream	7-10	60-70
receiving 2 video streams transmitting own video	17-23 (75-80 Kbps)	85-95 (15-20 Kbps)
receiving 2 video streams receiving 1 audio stream transmitting own video	20-25 (75-80 Kbps)	95-100 (15-20 Kbps)

Workloads on the server machines:

In this test, eacg client join the audio session (receiving only, DVI coding), and the same video session (transmitting and receiving two mbone video streams plus video streams from all other clients. Each video stream 25-100Kbps, 1-5f/s).

It is observed that 1-2 seconds delay is introduced by the audio mixer, but there is no measurment about the delay caused by the video transcoding server.

Num of clients	CPU usage (Ultra-I)	CPU usage (Ultra-II)
0	1.2	3.1
1	3.8	8.8
2	13.4	17.6
3	15.2	23.8
4	24.7	40.8
5	41.6	58.0
6	69.1	86.9
7	84.7	89.0

5 Future Work

Shared whiteboard tool on PC. Currently there is no whiteboard tools available on Win95. `Teledraw` is a candidate being developed by RUS. Once it is available it would be possible to be used over ISDN in our settings.

6 Acknowledgement

We would like to acknowledge the help from the followings:

E. Amir (UC Berkeley) et al. for allowing use and modification of their *vgw* code.

J. Andrews and N. Pugh (UCL) for their help in setting up the link to the Ascend unit using the Teles box.

J. Crowcroft for his message about the UMbone idea.

The Merci group people, including partners from Teles, for their technical advice/help.

References

[1] E. Amir, S. MaCanne, and H. Zhang. An application-level video gateway. In *Proceedings of ACM Multimedia '95*, 1995.

[2] J. Crowcroft, I. Wakeman, M. Handley, and S. Clayman. *Internetworking Multimedia*. UCL Press, 1996.

[3] M. Handley. Session coordination protocol. Technical report, UCL, 1996.

[4] S. MaCanne. *Scalable Compression and transmission of Internet Multicast Video*. PhD thesis, University of California, Berkeley, December 1996.

[5] S. MaCanne and V. Jacobson. *vic*: a flexible framework for packet video. In *Proceedings of ACM Multimedia '95*, 1995.