

ASYNCHRONOUS VIDEO TELEPHONY FOR THE DEAF

Zhenyu Ma and William D. Tucker

University of the Western Cape, Computer Science Department

Private Bag X17, Bellville 7535 South Africa

Phone: +27 21 959 3010 Fax: +27 21 959 1274 Email: {zma, btucker}@uwc.ac.za

Abstract—The South African Deaf community has very limited telephony options. They prefer to communicate in sign language, a visual medium. Real-time video over Internet Protocol is a promising option, but in reality, the quality is often not enough for the Deaf to be able to understand each other’s sign language. Furthermore, these applications were not design specifically for the Deaf. This paper introduces an asynchronous video chat system to provide better quality video at the expense of increased latency. It determined a codec/transmission protocol combination in the laboratory environment and tested it out with actual Deaf users. This paper will address the results based on comparison between different codecs, transmission protocol on asynchronous video communication for the Deaf.

SATNAC Classification: Innovation and Regulatory – Telecommunications Developments and Inventions

Keywords: asynchronous, Deaf telephony, Quality of Service, video over IP, Video Relay Service

I. INTRODUCTION

Due to a lack of hearing and often limited speaking abilities, the Deaf are effectively cut off from voice-based communication. The Deaf require third party mediation in order to communicate with a hearing person. Consider face-to-face communication where a bilingual mediator can translate between a given sign language and a spoken language. The same sort of “relay” can occur over a telephone with a co-located mediator. For example, a mediator would stand next to a Deaf person, using the phone on behalf of the Deaf user. In order to achieve some degree of independence for the Deaf user, a relay service can rather situate a relay operator in a call centre. Since the Deaf user cannot speak, communication with the relay operator must be via either text or sign language. Since sign language is visually expressive, the Deaf prefer a video interface in order to sign with the mediator[9].

Such a Video Relay Service (VRS) is quite common in the developed world, e.g. USA, UK and Australia. Text-based services are also available via ubiquitous Instant Messaging (IM) or the use of specialized text telephones (TTY). VRS and text relay services are often subsidised by telecommunications operators. In South Africa, there is no longer such a service – although there has been some local work in this area (described in Section II). Although several text communication options are available, even some with relay, the South African Deaf prefer to sign, and that requires video[9].

Standard Internet-based video over wired and wireless broadband may be enough for non-Deaf users, but is often of too poor quality for signing Deaf users – motion is blurry and facial expressions are not clear. Most Deaf people do not have dedicated bandwidth anyway. They might have access to GPRS and 3G with their cell phones, or shared broadband connectivity at best. Even so, Video over Internet Protocol (Video over IP) often lacks the Quality of Service (QoS) to provide intelligible sign language communication.

This paper introduces a different approach. Instead of using poor quality synchronous video between a Deaf person and a VRS operator/mediator, it employs an asynchronous option that offers better communication quality at the expense of slightly more latency. A store-and-forward approach to video ensures that sign language video is clear and understandable. This paper therefore presents a prototype for asynchronous video telephony for the Deaf. Section II describes local South African Deaf telephony environment. Section III delves into work related to VRS systems, video over IP in general and briefly explores synchronous video compression and codecs. Section IV presents goals for an asynchronous prototype. Section V details the prototype architecture and design. The prototype was tested both in the laboratory and with an actual Deaf community. Section VI describes the experimental design, data collection and analysis. The paper concludes in a final section and identifies future work that will utilise the asynchronous video prototype within a locally crafted VRS.

II. BACKGROUND

VRS is quite mature in the developed world, but this is not the case in South Africa. There was a brief VRS several years ago in the form of TISSA (see below), but the South African Deaf only really have textual telephony options at their disposal. At least there are several options.

The first is a dedicated text telephone (TTY) device built and distributed by Telkom called the Teldem. The advantages and disadvantages of the Teldem have been clearly documented [10]. Unfortunately, the Teldem has not experienced successful uptake. The logical successor is Instant Messaging. However, due to the extreme disadvantages of endemic poverty and illiteracy for many South African Deaf people, PC-based communication is not a readily available option. The most common text communication mechanism for the Deaf in South Africa, then, is SMS on a cell phone. For the Deaf, SMS is a still a relatively expensive option, but it enables them to communicate with Deaf and hearing people alike. Cheaper cellular options for text are available, such as MXit and Fring, but awareness of these options in the Deaf community is low (but growing). Thus far, however, there

has been very little work on video relay.

Such a service was piloted in South Africa several years ago. The Telephone Interpreting Service for South Africa (TISSA) was launched for all eleven official South African languages and also catered for sign language [3]. However, after a six-month pilot, TISSA was cancelled only to reappear in 2005, but without support for sign language [7].

There has been some other local work with respect to text-based relay for all three of the above-mentioned text-based options: Teldem [5], Instant Messaging [13][14], and even with SMS (an extension to [13] not yet reported in the literature). One of these is shown in Fig. 1.

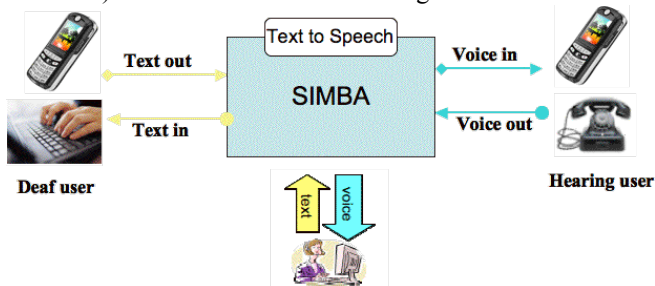


Fig. 1. SIMBA is a locally developed text relay service [13]. The Deaf user communicates with a hearing user with text via Instant Messaging or SMS. The relay service is partially automated. SIMBA converts outgoing text to speech with an open source Text to Speech (such as FreeTTS), and delivers audio to a Voice over IP (VoIP) soft phone or to some form of handset, via a PSTN/GSM gateway (not shown). A relay operator has a soft phone and texts the hearing user's speech to the Deaf user.

SIMBA trials have been experimented in Deaf Community of Cape Town (DCCT), a Deaf NGO in Newlands, Cape Town, for more than a year. Unfortunately, the Deaf seldom used the application even though they had explicitly asked for the SMS interface. After expending so much effort on semi-automated text relay, it was turned out to be that the Deaf were much more interested in communicating in their mother tongue, South African Sign Language (SASL). More importantly, they were much more interested in communicating with each other, i.e. Deaf to Deaf communication, than with relay to hearing users. So the research refocused its effort on video telephony for the Deaf.

III. RELATED WORK

The majority of the South African Deaf community has little exposure or access to Information and Communication Technology (ICT) [11]. This is not the case in much of the developed world where ICT access is often taken for granted, even by the Deaf. This section presents work related to video telephony for the Deaf in the following areas: VRS, video over IP, video messaging and information delivery, and finally, video codecs and compression techniques. The first two topics concern synchronous messaging. The third topic concerns asynchronous communication. The codecs and compression techniques are relevant to both forms of communication.

A. Video Relay Service (VRS)

An example of a VRS is the Hamilton relay VRS (www.relaycall.com/vrs). The system allows the Deaf and hearing to interact via an Internet enabled telephone, TV set

or videophone. The system requires either installation of some specialised software and/or expensive equipment. The Hamilton service separates the communication into two parts. One is synchronous video communication between a Deaf person and a relay operator. The other is synchronous voice communication between a relay operator and a hearing person. Overall, the communication between the Deaf and hearing users is considered asynchronous.

Another example is the Wireless Information Service for Deaf people On the Move project (WISDOM) [2]. This Swedish initiative meant to provide VRS for the mobile (cellular) market, but also works on a PC with video over IP and VoIP. WISDOM pursued an ambitious research agenda including: real time conversation service, sign language video relay service, distance sign language and automatic sign language recognition. Not all of these objectives were attainable.

B. Synchronous Video over IP

There are a host of available synchronous video tools meant for non-Deaf users. There are numerous dedicated high end ISDN and broadband H.323 and SIP-based video systems. The cost of these systems precludes budget experimentation, so only “free” Internet tools are taken into consideration, such as Ekiga, Skype or CamFrog running with low-end web cams like the Logitech series. Such systems are often free and can support synchronous communication with both P2P and multicast. Most are forced to put a performance limit on the number of simultaneous video users due to bandwidth overhead. The quality of video is consequently constrained.

There are, however, examples of synchronous video systems designed specifically for Deaf users. Mobile ASL is such a system intended to run, similar to WISDOM, on cell phones [1]. The Deaf use a rich combination of visual communication that includes hand gestures, facial expressions and body and eye movements. So much visual detail is difficult to achieve with a low-end device such as a web cam or cell phone.

C. Video messaging and information delivery

Low-end devices are more conducive to asynchronous video communication. This section briefly mentions several tools that employ video media to either send a message or deliver information in a manner that a Deaf person could use.

The EyeJot web-based system innovatively incorporates email functionality to video over IP (www.eyejot.com). The product borrows from email systems, replacing text content with video messages. In this way, Eyejot provides asynchronous video over IP for anyone, Deaf or not. Because it is web-based, Eyejot is client-free, easy to operate, and supports a limited form of offline “chatting”. In some ways, then, Eyejot is not a dedicated chat tool, but more like video messaging via email.

Many Deaf people around the world use SMS. However, in developing countries, particularly in South Africa, the relatively high cost of the SMS service prevents the Deaf from using it frequently. The video version of SMS, the Multimedia Messaging Service (MMS), is just as expensive and requires more contemporary hardware to access GPRS and 3G networks, and to create/view video. Yet MMS is a viable asynchronous communication option for the Deaf.

Unfortunately, South African service providers do not subsidise call/data usage for Deaf users as in other countries. Our experience with DCCT tells us that few Deaf South Africans use MMS because they simply cannot afford to do so.

On a completely different front, the TESSA project (not to be confused with TISSA [3]) is asynchronous in that it offers one-way information delivery on request with a virtual signing system [12]. TESSA is part of the larger ViSiCAST project. TESSA is an information service that aids a clerk in a post office to transact with a Deaf person. TESSA translates the clerk's speech to sign language via speech recognition, phrase lookup, phrase assembly and then displays sign language to a Deaf person with a virtual signing avatar. The overall ViSiCAST project aims to embed virtual signing into a Deaf person's daily life, from multimedia and Internet to face-to-face transactions and broadcast.

D. Video codec and compression techniques

With so many video telephony options available, there has been much work to design more efficient codecs and to provide more video compression standards. A given codec employs a standard to compress a video file by removing some unnecessary information for storage and transmission, and to decompress for playback. Some common examples are Quicktime, RealVideo, FlashVideo and DivX. Fortunately, many versions of these are freely available, if not open source. FFmpeg is a free software solution that can record, convert and stream video. It is available for multiple programming languages and can be compiled on most operating systems. For example, there is a version in Java for the Java Media Framework (JMF) called JFFmpeg. The promising point about FFmpeg is that it comprises a libavcodec library that is the leading audio/video codec library and supports many newly produced codecs.

IV. ASYNCHRONOUS PROTOTYPE GOALS

With so much support to develop video telephony applications, it is no surprise that there are many "free" options available on the Internet. However, these freely available synchronous video over IP systems are not designed nor engineered specifically for Deaf end users. Some tools may demonstrate reasonable performance for Deaf users with P2P communication over a LAN. For example, Skype has very clear LAN performance with only two users, and is also easy to operate with little training required. However, once participants are not co-located on a LAN, or the number of participants grows, performance and quality drop dramatically.

Most video communication applications are based on Real-time Transport Protocol (RTP). Since RTP is based on User Datagram Protocol (UDP), some quality is often sacrificed for speed via packet loss. Moreover, even today's best video encoders cannot produce the quality video needed for sign language communication in real time. Therefore, a system with asynchronous video relay over IP via a peer-to-peer (P2P) connection was designed. The basic idea is illustrated in Fig. 2.

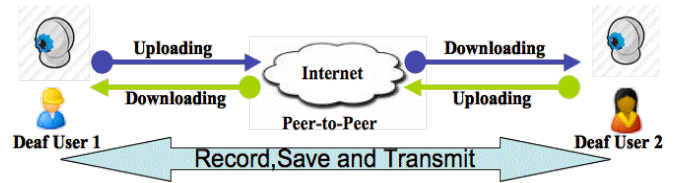


Fig. 2. Asynchronous video relay employs store and forward transmission to bridge the communication between two signing users. Asynchronous transmission improves video quality at the expense of increased latency. The users communicate in sign language using video. The system provides a simple peer-to-peer (P2P) architecture.

The scenario for this strategy is that a user requests a connection with another user with a simple client/server service. The server initialises the communication channel for both sides. Then, a Deaf user signs in front of the web camera. A web camera captures a sign language video with an appropriate codec or compression standard. The frame rate is set to enable superior video quality. Then the video is saved on the local machine. The user decides when to send it to the other user by a transfer protocol. The server does not participate in the communication at all except for setting up the communication connection for the P2P clients.

Considering that many South Africa Deaf people are functionally illiterate and/or undereducated, the human computer interface must be easy to understand and convenient to operate. To achieve this, two separate windows will be provided, one each for video-in and video-out, respectively (see Fig. 3). The video-in screen displays the video frames or video clips transferred in whilst the video-out screen is used to capture the video recorded from the web camera. Similarly, several simple buttons are provided, including a replay option.

Apart from basic login and registration interfaces, the video chat interface is the main interface to an end user. This interface actually hides much underlying functionality from the user, such as recording techniques, video compression, transmission, decompression, and playback. Notification interfaces are provided for each action. For example, if a user has received a new video, the application notifies the user of the availability of a new video (see Fig. 4).

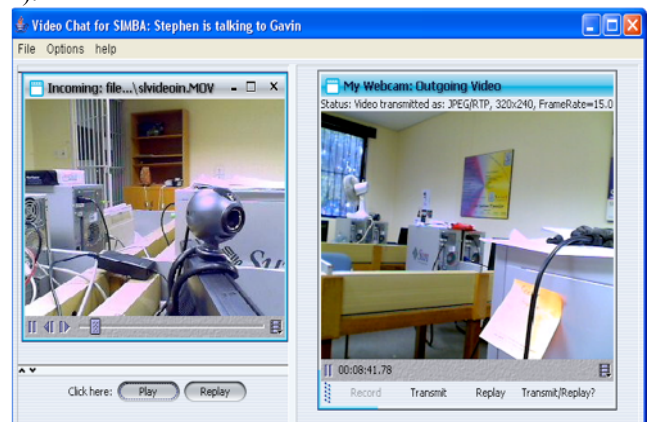


Fig. 3. The video chat user interface has a video-in screen (left) to play or replay the received incoming video. The video-out screen (right) is used to capture video from a web camera, and then transfers the video to another user's client interface.

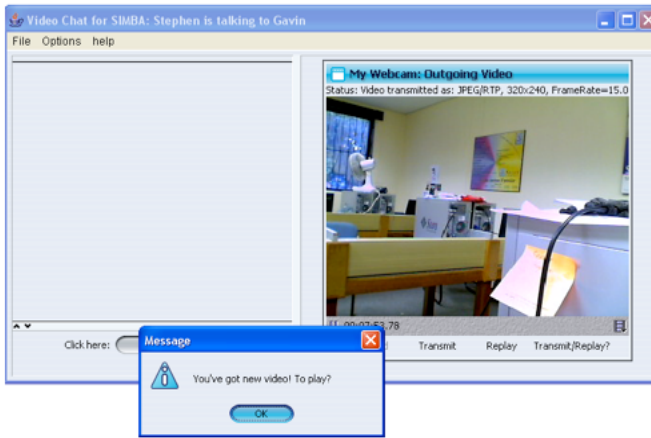


Fig. 4. When a new sign language message is available for viewing, the system notifies the user with a popup dialog.

V. VIDEO CHAT PROTOTYPE DESIGN

The video chat application is intended to be seamlessly compatible with the existing SIP-based SIMBA service. Since SIMBA was developed with Java APIs for Integrated Networks (JAIN) from National Institute of Standards and Technology (NIST) [13], the new video client is also meant to be completely open source and implemented in Java with Java Media Framework (JMF). SIMBA provides clients based on SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE), a SIP proxy and a SIP gateway [13]. The video chat application builds on the foundation of current services and adds a file transmission service and a file queuing service for asynchronous video over IP. In reality, however, the video chat prototype was first developed as a standalone P2P system for unit testing.

A. Constraints and assumptions

The SIMBA server was developed on the Windows operating system platform. However, the main programming language was Java so that the system could be moved across operating system platforms. JMF is still a work in progress with respect to video processing, esp. the limited number of supported compression standards and consequently can be difficult to integrate with video capture hardware and their drivers.

B. Tools used

A Java-based Unified Modelling Language (UML) program was used, such as Star UML or Poseidon UML, to diagram the design (shown below). Apache is used on the server side so that client interaction can be easily serviced. NetBeans 5 integrated with Java JDK1.5 acts as the programming platform.

C. Architecture

The overall architecture of key modules is shown in Fig. 5. The client side starts with the video control module, and is meant for either two Deaf users, or a Deaf user and a relay operator.

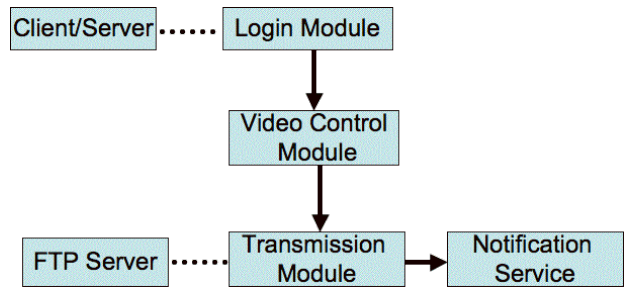


Fig. 5. The video chat system components and their relationships are shown here. The server only handles user login and call setup. The P2P clients handle the video compression and transmission, and the interface to the end users.

The login control module enables connection of two parties with a simple client/server rendezvous service. More detail on the video control is shown in see Fig. 6. The key functions are to:

- 1) Record video: allows a Deaf user to record what s/he signs in front of a web camera
- 2) Save video: video is saved to a file to be sent out later
- 3) Open file: opens a received video file
- 4) Play file: allows the user to view a video in the player
- 5) Upload: send a video
- 6) Download: receive a video

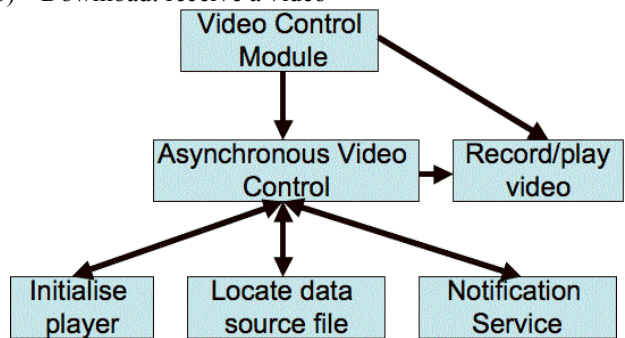


Fig. 6. The video control module is responsible for all functionality related to handling the asynchronous video: player initialisation, source file handling, data format, recording and playback (including replay).

The transmission control module (see Fig. 7) employs File Transfer Protocol (FTP) to exchange video files between P2P clients. Each client acts simultaneously as an FTP client or server, depending on whether it is receiving or sending a sign language video file, respectively.

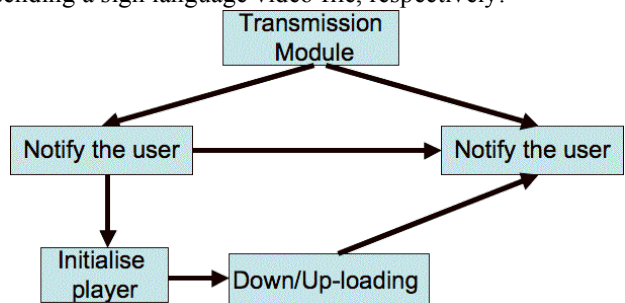


Fig. 7. The transmission control module is concerned with file transfer management via FTP, and also with user notification of file transfer activity.

VI. TESTING, DATA COLLECTION AND ANALYSIS

Once the basic video chat prototype was implemented, the research sought an optimal solution with respect to system

performance. An evaluation process was accomplished through several steps. First, the test examined different video formats and codecs, such as JPEG, MPEG, H.261 and H.263. Next, several different transfer protocols: FTP, TFTP and SFTP were explored. The result determined an optimal combination of video format/codec and transfer protocol, and then took that combination to real users at a local Deaf community. With those Deaf users, this research also explored the differences between asynchronous and synchronous video over IP by administering simple survey questionnaires.

A. Compression tests

File compression algorithms vary widely, but all of them are concerned with ways to reduce the space a file occupies. Video chat requires a method that can rapidly compress and decompress a compact file [8]. This test examines file size and perceived quality. It is not necessary measuring compression/decompression times. The test recorded the same videos in different codecs and compared their file sizes and their quality. Perceptively, RGB, H.261 and H.263 were not very clear and it was hard to recognize the content of video files due to segmented and/or interrupted frames. MPEG, one of the better video compression standards, was not tested because JMF did not yet support MPEG. On the other hand, JPEG demonstrated decent quality albeit with larger file sizes. Overall, the aim of this research was looking for acceptable compression and quality. Therefore, JPEG was chosen for the time being.

B. Transmission tests

Fast file transmission speed is crucial to achieve lower latency for users. Obviously, speed is dependent on the size of the file to be transferred, and the network throughput as well. Due to the limitation of facility provided in the lab, the fixed wire-line bandwidth was 10 Mbps to carry out the transmission tests, FTP, SFTP (secure FTP) and TFTP (trivial FTP) on different file sizes. A Deaf person was encouraged to create variable length sign language messages by giving them a series of more and more complex topics to discuss. There were no interest in the conversation topics but those caused a graduated series of longer sign language messages to get different file sizes. It is worth mentioning that these reference sign language messages were obtained via a conversion using the asynchronous video chat application. The timing and file sizes were recorded for analysis. During subsequent testing, the Deaf user experienced protocol-dependent latency. The overall comparison is shown in Fig. 8.

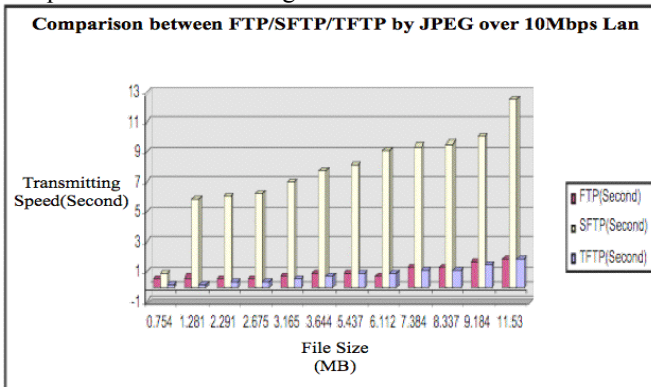


Fig. 8. Time consumption comparison among FTP, STP and TFTP based on specific codec and bandwidth.

SFTP is based on FTP and uses SSH. SFTP is capable of resuming an interrupted transfer, and can directly list and remove remote files. Because of the security, SFTP's performance, however, is not ideal. The simplicity of TFTP revealed itself in faster transfer times. Unlike FTP and SFTP, TFTP is based on UDP, a connectionless protocol that is much faster than those based on TCP. However TFTP is so simple that it omits some desired functionality such as error handling, directory listing, and permission certification, etc. Video chat needs some of this functionality, so the FTP was chosen for file transfer.

C. User tests

Then video over IP tests was conducted with Deaf users in both asynchronous and synchronous modes. First the test designers created a scaled questionnaire sheet for each participant on video quality, interface and delay. In asynchronous testing, the optimal combination of JPEG and FTP was used throughout the test. For synchronous tests, the Deaf users were asked to use JMStudio, MSN, and Skype for real-time video conferencing. In both modes, the test encouraged the Deaf users to discuss several topics ranging from shorter to longer conversations.

The questionnaire results showed that the Deaf users did not like JMStudio and MSN at all. They struggled to keep the conversation going due to the poor quality of video. Skype did a better job over the LAN but was not satisfactory via the over the Internet (via 512kbps ADSL). Also, the Skype interface was not easy for the Deaf to use since it was not developed specifically for the Deaf. It was necessary to conduct a special training class on Skype so that they would use it internally in their offices.

In comparison, the users liked the increased quality of video in asynchronous tests, but they felt the delay was too long. Herein lay the tradeoffs for asynchronous video. Asynchronous video provides better quality than synchronous options, but introduces more delay from cumulative time spent to record, compress, transmit, decompress and playback.

VII. CONCLUSION

Video telephony for the Deaf is a challenging alternative to conventional text telephony that dominates remote communication for the Deaf. Synchronous video over IP enabled the Deaf to communicate with each other in their own language. Moreover, it also enables the Deaf to communicate with hearing people via a sign language-oriented relay operator. However, the quality of synchronous video remains an issue.

Asynchronous video over IP offers an alternative approach to video telephony for the Deaf with proper codec/transmission combination. With experimental tests, It offers improved video quality with regardless of bandwidth interference, but several other issues need to be considered, such as reducing the overall delay or learning how to deal with it at the user interface level, and optimising codecs.

VIII. FUTURE WORK

From the testing, informative feedback posed lots of issues to be improved. Future work will concentrate on the following:

Codec optimisation. The compression algorithm could be

optimised for sign language by placing more emphasis on the Region of Interest (ROI) rather than the whole frame[8]. With the help of a conventional quantitative comparison method, such as Peak Signal to Noise Ratio (PSNR) and Picture Quality Rating (PQR)[5], it is easier to measure video quality directly from the ROI and not the entire video scene.

SIMBA integration. The SIMBA server could be equipped with the ability to switch amongst synchronous video, asynchronous video and text communication due to the availability of bandwidth (see Fig. 9).

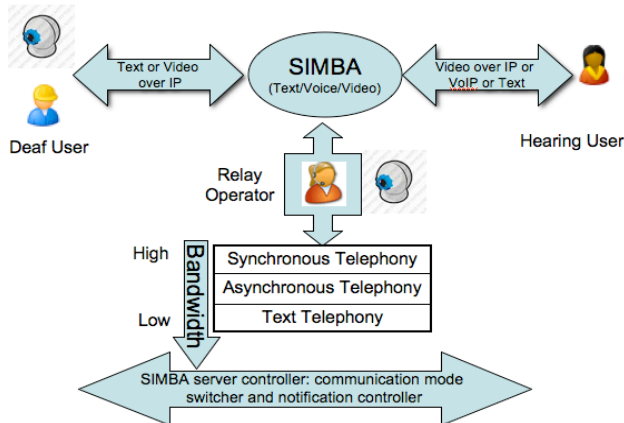


Fig. 9. Video integration onto SIMBA with bandwidth-awareness switching. Video chat integration into SIMBA could help tackle bandwidth issues by adjusting communication modalities in order to provide acceptable and intelligent telephony for the Deaf.

User notification. In order to catch a Deaf user's attention, the user interface must be able to draw the attention of a Deaf user without sound. For example, one of the Deaf participants suggested vibrating his cell phone via Bluetooth. Another challenge arises from the long conversation delays: the users would like to know what is happening on the other side. They desire awareness of file transmission, incoming video or sign language translation.

ACKNOWLEDGMENTS

The authors thank the staff and members of Deaf Community of Cape Town (DCCT) for their participation in the project. We also thank Telkom, Cisco and THRIP for financial support via the Telkom Centre of Excellence (CoE) programme.

REFERENCES

[1] A. Cavender, R. E. Ladner, and E. A. Riskin, "MobileASL: intelligibility of sign language video as constrained by mobile phone technology," in *8th international ACM SIGACCESS conference on Computers and accessibility*, Portland, Oregon, USA, 2006, pp. 71-78.

[2] B. Bauer and K. F. Kraiss, "Towards a 3rd Generation mobile Telecommunication for Deaf people," in *10th Aachen Symposium on Signal Theory*, Aachen, Germany, 2001, pp. 101-106.

[3] B. S. Ngubane, "For the Launch of the Pilot Project: Telephone Interpreting Services for South Africa (TISSA)." vol. 2006 Katlehong Police Station: Minister of Arts, Culture, Science and Technology, 2002, p.

announcement of the failed TISSA service.

[4] C. L. James & K. M. Reischel(2001) Text input for mobile devices: comparing model prediction to actual performance. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 365-371, 2001.

[5] D. A. Rosenthal, "Analyses of selected variables effecting video streamed over IP," *International Journal of Network Management*, vol. 14(3), 19 March 2004, John Wiley & Sons, Inc, pp. 193-211.

[6] J. Penton, W. D. Tucker, and M. Glaser, "Telgo323: An H.323 Bridge for Deaf Telephony," in *South African Telecommunications Networks & Applications Conference*, Drakensberg, South Africa, 2002, pp. 309-313.

[7] K. van der Merwe, "Want access to government services? With TISSA you can!." vol. 2006: SAPS, 2005.

[8] L. J. Muir & I. E. G. Richardson, "Video telephony for the Deaf: Analysis and Development of an optimised Video Compression Product," in *ACM Multimedia Conference*, Juan-les-Pins, France, 2002, pp. 650-652.

[9] M. A. Hersh, and W. D. Tucker, "Ethics and Monodisciplinarity: Positivism, Informed Consent and Informed Participation," in *Proc. 16th IFAC World Congress*, Prague, Czech Republic, 2005.

[10] M. Glaser, "A Field Trial and Evaluation of Telkom's Teldem Terminal in a Deaf Community in the Western Cape," in *South African Telecommunication Networks and Applications Conference*, Stellenbosch, South Africa, 2000, pp. (CD-ROM publication).

[11] M. Glaser, P. Young, & L. Porteous, "An Information and Communication Technology (ICT) Needs Analysis and Training Programme for Deaf People," in *20th International Congress on the Education of the Deaf*, Maastricht, Netherlands, 2005, p. 231.

[12] Stephen. Cox, Michael Lincoln, Judy Tryggvason, Melanie Nakisa, Mar Wells, Marcus Tutt and Sanja Abbott, "TESSA, a system to aid communication with deaf people," *Proc. Fifth International ACM conference on Assistive technologies Assets '02*, Edinburgh, Scotland.

[13] T. Sun, & W. D. Tucker, "A SoftBridge with Carrier Grade Reliability Using JAIN SLEE," in *South African Telecommunications Networks & Applications Conference*, Stellenbosch, South Africa, 2004, pp 251-252.

[14] W. D. Tucker, M. Glaser, and J. Lewis, "SoftBridge in Action: The First Deaf Telephony Pilot," in *South African Telecommunications Networks & Applications Conference*, George, South Africa, 2003, pp. 293-294.

AUTHORS

Zhenyu Ma is a Masters student with the Broadband Applications Network Group (BANG) in the Department of Computer Science at the University of the Western Cape. He is currently working on video relay service over IP for the Deaf.

William D. Tucker is a senior lecturer of Computer Science at UWC. He leads BANG research. His PhD on communication abstractions is near completion at the University of Cape Town.