

WHITEBOARD-BASED TELECONFERENCING APPLICATION ON MOBILE DEVICES OVER EXISTING XMPP SERVERS

John Bennedict T. Lorenzo and Rowel O. Atienza

Electrical and Electronics Engineering Institute, University of the Philippines, Diliman, Quezon City, Philippines

Keywords: Remote Collaboration, Digital Whiteboard, VoIP, Mobile Application.

Abstract: The need for internet teleconferencing is fueled by businesses which do not have the time and resources to transport all the needed employees located around the world to work on a project or collaborate on an idea. Current solutions to internet teleconferencing include VoIP and video conferencing, e.g. Skype. VoIP permits a natural way of communication through voice but lacks visual information communication. On the other hand, video conferencing requires a lot of bandwidth. The proposed approach uses a shared drawing board, which all collaborators can see in realtime, together with VoIP to provide an intuitive and effective communication medium. Our approach requires less bandwidth than video conferencing and is able to communicate ideas such as ways of solving a circuit problem, collaborating on an interface prototype for a new project, or strategizing a basketball game. Our approach utilizes the touch screen interface of current tablets and smartphones as an intuitive way to draw. The application uses Google's existing XMPP servers, for three reasons: (1) most people already have accounts on them, (2) those accounts have existing contacts, and (3) offload server maintenance to another entity. The system obtained a passing SUS score and a lower than expected MOS.

1 INTRODUCTION

Collaboration over the internet is a convenient way to meet people from across the globe without having to wait for them to arrive and without spending for airfare. People often need to collaborate to produce meaningful work. The lecture hall of a university for example, collaborate the lecturer with the students to educate them. These lecture halls sometimes have a video projector that commonly projects a whiteboard onto a larger screen that all the students can see. The benefit obtained in that situation is sharing of information through graphs and writings. Therefore, even if the students and the lecturer are remotely located from each other, what is important is the communication that leads to sharing of information. When the students provide feedback to the lecturer via questions and queries, would it not be better to also allow them to draw on that whiteboard? Provided their insight and what they are going to draw there is meaningful. Having both students and lecturer being able to access the whiteboard makes the whiteboard shared, and if both parties are cooperative, collaboration between them begins. Our approach is inspired by great lectures made on

whiteboards or blackboards on lecture halls in a university.

Looking at the available technologies for remote collaboration, videoconferencing systems can bring a more realistic collaboration experience to users. However it requires enormous bandwidth, requires the cameras to be fixed on the collaborators' object of interest, and costs a lot. Videoconferencing also does not allow people to modify a shared object, such as a whiteboard. VoIP collaboration systems require less bandwidth than videoconferencing, however, they lack the ability to relay visual information. Meanwhile, pure text-based conferencing is similar to VoIP, both of which cannot easily relay visual information. However this kind of conferencing requires little bandwidth as it has been used since the 56k era. To be able to relay visual information while using as little bandwidth as possible, drawings can be represented as a series of line strokes, which can be transmitted as text containing its start and end points. Using this low bandwidth text transmission to relay visual information and at the same time enabling users to communicate naturally with their voice, using VoIP, one can build a system from tablets and smartphones

of today to produce a portable productivity and collaboration tool.

The number of smartphone owners worldwide has increased from 41 million users in 2009 to 80.5 million users in 2010 (Gartner, 2011). That figure does not yet include other mobile devices such as tablets, e.g. the iPad, and non-phone devices, e.g. iPod Touch. Due to this large and growing user base, mobile collaboration services provide tremendous new opportunities that are not possible on PCs (Li *et al.*, 2008). In addition, due to the fact that smartphones use a touch screen interface, the user has an intuitive way to draw on the device. Our proposed approach uses these advantages of the mobile device, specifically the iOS devices, both increasing user base and usability, as a leading edge over existing whiteboard collaborative systems.

There are existing applications like the proposed system in the Apple App Store. These are Netsketch (Gotow, 2011) and Whiteboard Pro (GreenGar Studios, 2011). These applications provide collaboration on a shared whiteboard. However, the applications only works on local network due to absence of a server for user presence. Having a collaboration tool work only on the local network prevents use cases of the system where collaborators are geographically challenged. The proposed system works outside of the local network by utilizing an existing presence and messaging server. In addition, the proposed system provides voice over IP, not found on other existing apps, that would possibly improve the perceived usability by users.

There are existing work on using electronic whiteboard on mobile devices. Work by Ambikairajah *et al.* (2007) focuses on using tablet PCs for use with teaching DSP education. Demeure *et al.*, (2005) used PDAs with whiteboards as supplement to lectures and meetings. Liu *et al.* (2002) did an earlier attempt to use wireless technologies as a tool in a classroom environment. The system would also work for people located remotely from each other.

The application without its Voice over IP capabilities has also been released on the Apple App Store under the name DoodleTalk.

The next section contains details about how the system was implemented from building the digital whiteboard and voice over IP features, call signaling and NAT traversal. Then in the succeeding section the system is tested and results are presented.

2 DESIGN AND IMPLEMENTATION

To implement the application, several libraries and frameworks were used. A library that implements the RTP stack called oRTP (Morlat, n.d.) was used. A library that implements XMPP related functions called xmppframework was used.

The shared digital whiteboard implemented allows two or more people to share a digital whiteboard. A person's changes to the digital whiteboard can also be seen by others. This enables communication not limited to text.

To complement the digital whiteboard feature, voice over IP is implemented. Voice communication will potentially allow easier communication than with the shared digital whiteboard only. The expected experience on the final application will be similar to listening to a classroom lecture while looking at notes being made shown on a digital video projector.

The voice over IP feature is implemented by parts in the following sequence: audio related functions, peer to peer communication, call signaling, and NAT traversal but most details of the implementation will not be discussed here. The audio related functions provide access to the voice data captured by the microphone and also the buffers that the speakers use when playing audio. Peer to peer communication provides wrapping of audio data into RTP packets, and facilities to send and receive RTP packets. Call signaling provides automatic discovery of the peer's IP address and port and exchange those information with that peer to establish a working call. Finally, NAT traversal allows the voice over IP feature to work through most NAT topologies.

Call signaling is derived XEP-0166 (Ludwig *et al.*, 2009) and the Google Talk Call Signaling protocol document (Google, n.d.a).

NAT traversal was implemented by using STUN, which is defined in RFC 5389 (Rosenberg *et al.*, 2008). It uses Google's own STUN server that echoes the public address and port number where a packet originated. Even with this utility, the application will work only on some network topologies.

The packet sniffer used is Wireshark on Mac OSX. The network setup used when sniffing packets is as follows. A laptop running Mac OSX was setup to make an ad-hoc network that shares its ethernet connection. Then an iPod Touch is made to connect to the ad-hoc network. That way, packets going to and from the iPod Touch can be sniffed including

packets to and from the iPhone Simulator running on the laptop.

2.1 Digital Whiteboard

The protocol for implementing communication for the devices using the digital whiteboard is partially derived from Postscript commands.

There are four PostScript derived commands used in drawing: `lineto`, `moveto`, `color`, and `width`. The functions of these instructions are as follows. The instruction `'lineto x y'` is used to represent a line drawing from a previous saved point (x',y') to the coordinate (x,y) and then changes the saved point to (x,y) . The instruction `'moveto x y'` is used to represent a state change that moves the saved point to (x,y) . The instruction `'color c'` changes the state to use the color c for the succeeding drawing instructions. The instruction `'width w'` changes the state of drawing to draw lines of thickness w in the succeeding drawing instructions. Finger painting can be simulated using these four instructions and that is the reason for choosing just these four instructions from Postscript. These instructions are passed to the participants of the digital whiteboard using XMPP messages.

The protocol for group communication used in described in XEP-0045: Multi-user Chat of the XMPP Standards Foundation (Saint-Andre, 2008). The XEP-0045 standard describes the messages that are being sent to and from the XMPP server and the users. It describes messages for various commands used in the multi-user chat. The commands particularly used in the implementation of the multi-user digital whiteboard are creating or joining a room, sending a message to all participants in the room, and receiving chat from the room.

A screenshot of the digital whiteboard is shown in Figure 1.

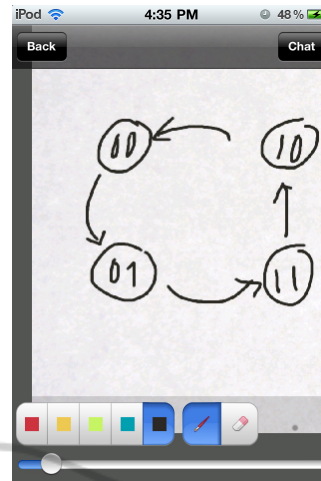


Figure 1: A screenshot of the application showing the shared digital whiteboard and controls.

2.2 Voice over Internet Protocol (VoIP)

The voice over internet protocol feature of this project was implemented to be compatible with Google Talk's implementation. That means the application developed can be used to initiate or receive calls to or from a Google Talk user. That is aside from other users using the same application developed in this project. This made it easy to test the functionality of the VoIP features. The codec chosen is iLBC in 30ms mode.

A visual diagram of how audio is processed in the application is seen on Figure 2. The audio data from the microphone is supplied to the application from the remote i/o unit's input element. In one of the render callbacks, the audio data from the microphone is collected and copied into a buffer that eventually gets sent to the remote peer/s. On the other hand the audio data received from the remote peers get processed into a designated render callback for that peer.

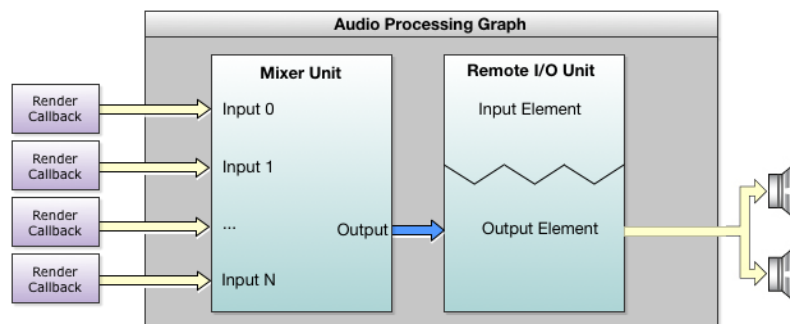


Figure 2: A graph showing how the audio units are connected. Render callbacks are triggered when they need audio data. The mixer unit combines audio from its inputs into a single stream of audio on its output. The remote i/o unit's input and output elements represent the buffers for the iOS device's microphone and speakers respectively.

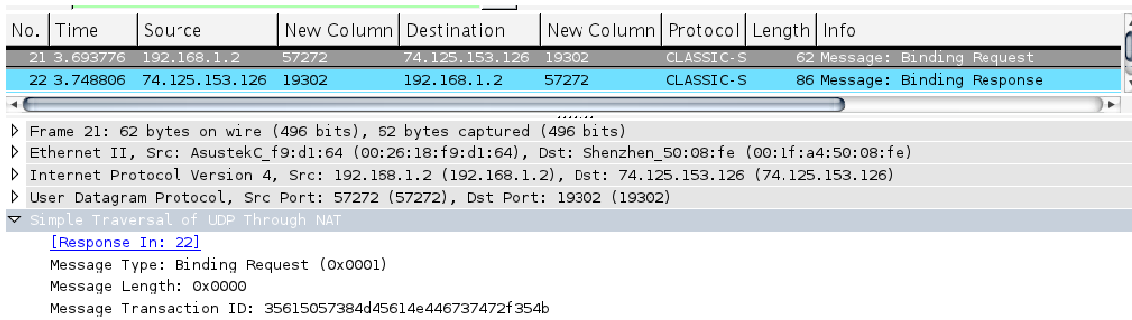


Figure 3: Wireshark capturing STUN binding request from Google Talk to a STUN server at 74.125.153.126.

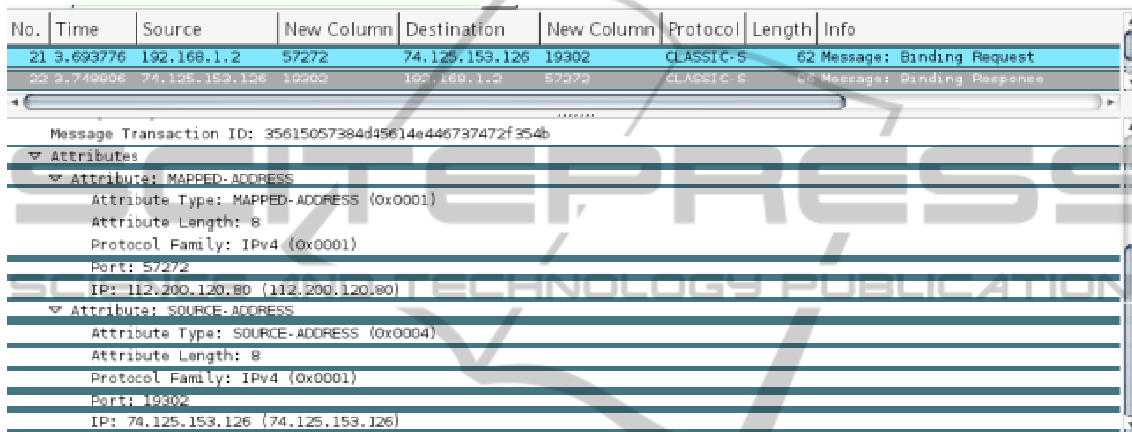


Figure 4: Wireshark capturing STUN binding response to Google Talk from a STUN server at 74.125.153.126.

2.3 Call Signaling

The protocol for call signaling used in the application is implemented to be compatible to Google Talk's. Therefore we derived the protocol from Google Call Signaling (Google, n.d.a) with some modifications. The modifications were derived by first following the call signaling protocol specification and inspecting the response of Google Talk's web client. The call signaling protocol by Google does not completely describe the protocol in use by Google Talk's voice chat call signaling.

2.4 NAT Traversal

STUN was implemented for NAT traversal, which according to Google (Google n.d.b) is enough to enable VoIP to work most network topologies without the need for a relay server. By inspecting the packets sent by

Google Talk while generating its candidates, it is found out that the binding request sent has no attributes and is just a plain STUN binding request addressed to some IP address assumed to be Google's STUN server, as seen in Figure 3. The

binding response for the request is a STUN packet containing mapped-address and source-address attributes, as seen in Figure 4. From the packet captures obtained, the ip address of Google's STUN server is dynamically obtained. Google's STUN server is used to discover the public ip:port pair for the application's VoIP socket.

3 RESULTS AND ANALYSIS

Two user studies were done to test the application. The first one is using Mean Opinion Score (MOS) test that aims to measure the satisfaction of users on the audio and conversational quality of the application. Testers for MOS test were gathered from a university. There were 6 testers for MOS test, 5 of them are male and 1 of them is female. They were asked to form pairs and test and application by pair. Pair by pair, they were oriented on the testing procedure. First, they are give two devices both running the application. Those devices running the application already has the VoIP call set-up so that testing is focused on the quality of audio. The two

Table 1: SUS questionnaire mean ratings.

Question	Mean Rating
I think that I would like to use this system frequently.	3.69
I found the system unnecessarily complex.	2.31
I thought the system was easy to use.	3.63
I think that I would need the support of a technical person to be able to use this system.	2.00
I found the various functions in this system were well integrated.	3.63
I thought there was too much inconsistency in this system.	1.88
I would imagine that most people would learn to use this system very quickly.	4.00
I found the system very cumbersome to us.	2.75
I felt very confident using the system.	3.94
I needed to learn a lot of things before I could get going with this system.	2.00

devices were connected to a common access point through WiFi, therefore the test is for voice chat on the local network. There are many other access points running 802.11g/b/n in the test environment aside from the access point used. The access point used for testing is a Ruckus Wireless ZoneFlex. Then one person in a pair is asked to read out to his/her buddy the five phrases recommended by (ITU-T, 1998) for MOS test.

The MOS of the system, as rated by the testers is 2.67. Compared to the MOS of iLBC codec in (Bhatia, 2006), iLBC has a MOS of 3.9, the application's MOS is lower.

Another test performed is for measuring usability of the whole application. It includes using both the shared whiteboard and voice chat. The test was performed by 16 people, one pair at a time. The testers were all students at a university with an average age of 20.69 years old and standard deviation of 0.82 years old. Of the 16 testers, 3 are female and 13 are male. They were asked to perform task completion and then fill up a questionnaire at the end of the task. The questionnaire items were derived from (Brooke, 2007). The task they performed was about building a floor plan, similar to the task performed by users in (Kunz *et al.*, 2010). In a pair of testers, each is given a unique set of requirements. They have to draw a floor plan that satisfies both of their requirements. The results of the test is shown in Table 1. The system obtained a SUS score with a mean of 69.84 and standard deviation of 14.21.

4 CONCLUSIONS AND RECOMMENDATIONS

From the results obtained by performing MOS, SUS,

and benchmarking tests, the application can be said to be usable but still needs to improve its MOS and SUS score. It is advised that more tests be performed with a larger sample size to make a more accurate measurement of the application's audio quality and usability from user's perspective. It is also recommended to perform the test on a more diverse set of users, because the tests performed are concentrated on getting feedback on people around the age of 20.

REFERENCES

Ambikairajah, E., Epps, J., Sheng, M., Celler, B., 2007. Tablet PC and Electronic Whiteboard Use in Signal Processing Education [DSP Education]. In *Signal Processing Magazine, IEEE*, vol.24, no.1, pp.130-133, Jan. 2007

Bhatia, M., Davidson, J., Kalidindi, S., Mukherjee, S. and Peters, J., 2006. VoIP: An In-Depth Analysis. October, 2011, from <http://www.ciscopress.com/articles/article.asp?p=606583&seqNum=4>

Brooke, J., 2007. SUS - A quick and dirty usability scale. Retrieved October, 2011, from <http://hell.meiert.org/core/pdf/sus.pdf>

Demeure, I., Faure, C., Lecolinet, E., Moissinac, J.-C., Pook, S., 2005. Mobile computing to facilitate interaction in lectures and meetings. In *Distributed Frameworks for Multimedia Applications*, 2005. DFMA '05. First International Conference on, vol., no., pp. 359- 366, 6-9 Feb. 2005

Gartner, 2010. Gartner Says Worldwide Mobile Phone Sales Grew 35 Percent in Third Quarter 2010; Smartphone Sales Increased 96 Percent. Retrieved January, 2011, from <http://www.gartner.com/it/page.jsp?id=1466313>

Google (n.d.a). Google talk call signaling. Retrieved January, 2011, from http://code.google.com/apis/talk/call_signaling.html

Google (n.d.b). Important Concepts - Google Talk for

- Developers. Retrieved January, 2011, from http://code.google.com/apis/talk/libjingle/important_concepts.html#connections
- Gotow, B., 2010. Netsketch collaborative drawing application for the iphone and ipod touch. Retrived January, 2011, from <http://itunes.apple.com/us/app/netsketch-collaborative-drawing/id284939612?mt=8>
- GreenGar Studios, 2010. Whiteboard pro: Collaborative drawing for the iphone and ipod touch. January, 2011, from <http://itunes.apple.com/us/app/whiteboard-pro-collaborative/id321986541?mt=8>
- ITU-T Study Group 12, 1998. ITU-T Rec. P.800 (08/96) Methods for subjective determination of transmission quality. Retrieved October, 2011, from <http://www.itu.int/rec/T-REC-P.800-199608-I/en>
- Kunz, A., Nescher, T., and Kuchler, M., 2010 . Collaboard: A novel interactive electronic whiteboard for remote collaboration with people on content. In *Cyberworlds (CW), 2010 International Conference on*, oct. 2010, pp. 430 –437.
- Li, D. & Chandra, U., 2008. Building web-based collaboration services on mobile phones. In *Collaborative Technologies and Systems*, 2008. CTS 2008. *International Symposium on*, may 2008, pp. 295 –304.
- Liu, T.-C. Wang, H.-Y., Liang, J.-K., Chan, T.-W., Yang, J.-C., 2002. Applying wireless technologies to build a highly interactive learning environment. In *Wireless and Mobile Technologies in Education*, 2002. Proceedings. *IEEE International Workshop on* , vol., no., pp. 63- 70, 2002
- Ludwig, S., Beda, J., Saint-Andre, P., & McQueen, R., 2009. XEP-0166: Jingle. Retrieved January, 2011, from <http://xmpp.org/extensions/xep-0166.pdf>
- Morlat, S., (n.d.). oRTP, a Real-time Transport Protocol (RTP,RFC3550) library. Retrieved October, 2011, from <http://www.linphone.org/eng/documentation/dev/ortp.html>
- Rosenberg, J. et al, 2008. RFC 5389 - Session Traversal Utilities for NAT(STUN). Retrieved January, 2011, from <http://tools.ietf.org/html/rfc5389>
- Saint-Andre, P., 2008. XEP-0045: Multi-user chat. Retrieved January, 2011, from <http://xmpp.org/extensions/xep-0045.pdf>