A Survey: Multi-User Video Chat Application

Nipun Gupta
B.Tech (Cse)Ims Engineering College, Ghaziabad

Sumit Awasthi
B.Tech (Cse)
Ims Engineering College, Ghaziabad

Paawan Mishra
B.Tech (Cse)
Ims Engineering College, Ghaziabad

ABSTRACT
This document aims at defining the overall software requirements for "Video-Chat Application" and describes overview of capabilities that will be provided by the android application. The use of online meetings is increasing each and every day and has grown to have a positive effect in our daily lives. This is due to the fact that it helps reduce travelling expenses and that it is accessible to everyone even those who are far away [2].

The demand for social networking sites is increasing day by day. Eighty seven percent of remote users feel more connected to their team and process when using video conferencing. Sixty five percent of communication is non-verbal [3]. A social networking site that allows you to video chat online is the primary inspiration for my project. This application will be used by general users to do chat, file sharing, voice calls, video calls using android device. Also audio and video conferencing facility has been provided. To make a call both the users need to be registered at the server. This communication is via Wi-Fi and will incur no expenditure to the end users. Communication between any two users will be enabled as long as both are logged-in at the server.

In this paper we briefly describe current Audio and Video conferencing experiences available to the general public.

Keywords
QoS; Compression; Error tolerance; Video conference; Audio conference

1. INTRODUCTION
Some of the existing video chat applications are Skype, Hangout, You-Cam, iChat, vsee. My application offers similar functionality like these social networking sites, and it is simple to execute and doesn't have to rely on third party sites. Users can directly enter to the application after signing up. Since online meetings occur over the internet, if bandwidth is low it results in the performance being low and causing some interruptions leading to task taken by clients over internet being a complete failure [2].

Open source tools such as Adobe connect and Zobo Show and many more do exist to support online meetings. But when using these tools in South Africa they seem to fail since South Africa's bandwidth is very low. They were built with assumptions of fast, usable and stable connections in a high bandwidth environment. In this project the aim is to mainly build a different application which has more or less similar features but the basic difference is that it is designed for a low bandwidth environment [4].

The QoS relies on the hardware environment such as the computer, webcam and microphone. Computer requires electricity to work and thus the internet is vulnerable to power outages and results in connections of clients and information being lost. In this project another aim is to solve the issue by the fact that every piece of information gets saved on the server and clients can always retrieve this information.

In the present day where communication plays such an important role in day to day activity, this project was proposed with the aim to provide free communication over smaller distance e.g. within an organisation campus. The application will be developed to operate an android based devices, like mobile phones and tablets with android versions 4.0 (API level 14) or greater.

2. BRIEF LITERATURE SURVEY
Online meeting tools allow remote meeting and collaborative work. Poor internet service however makes most web conferencing solutions unreliable for developing countries in general. In past surveys and review papers the most important point is the improvement in the user experience with low bandwidth and unstable Internet conditions for video conferencing.

A special focus is put on audio/video stream optimization, which is the most affected feature of a web conferencing system. The ongoing research in this area can be grouped into three main domains. Firstly, is research on rate adaptation schemes that aims to provide the best quality multimedia stream to several receivers with optimal use of available bandwidth. Secondly, is research on compression with attempts to reduce bandwidth requirements with acceptable content quality. The last research domain studies how to weaken the influence of transmission errors and problems over the content provided[4].

As per today, it is impossible to ensure a perfect QoS over the internet. So, lots of research is being conducted to enhance the user experience under poor QoS. Several research avenues are exploited to enhance the user experience under limited Internet transmission, which can be summarized in the following three main domains [5]:-

Transmission rate adaptation: Research in this area studies how to efficiently use the available bandwidth to broadcast a multimedia stream to several recipients.

Error tolerance: the aim here is to develop correction schemes to make transmission errors less noticeable by the user.

Compression: a good compression scheme reduces the network load while keeping a reasonable content quality. Findings in this area help to reduce the effect of low bandwidth on the user experience.

Research on rate adaptation aims to provide the best streaming broadcast quality adapted to the available bandwidth. The source based approach poorly uses the available bandwidth. On the other hand, the multilayer model approach offers the best rate adaptation since it allows the definition of different QoS on layers (leading to graceful signal degradation). Error correction techniques based on retransmission and redundancy are simply not applicable in Web meeting contexts, as they increase the network overhead. The buffering approach is a better solution[5].

Since meeting exchanges are not as interactive as a phone call, for example, the small delay introduced by the buffering will almost not be perceived by other participants, as long as the delivered stream is smooth. There is good progress on video compression too, but the bandwidth required (100 Kbps) to keep an acceptable quality can still be too much. So, it's better to combine a compression scheme with frame rate minimization to deliver a usable video stream over very low bandwidth conditions.

3. PROBLEM FORMULATION

The use of conferencing technology continues to grow as accessibility increases and costs decline. Open source and commercial tools often fail in a low bandwidth environment and they exists a great need to build a tool that works efficiently in such conditions. The aim of building a bandwidth-aware application tool is to help improve the user experience and also help in development projects such as ICT4D. ICT4D projects try to build applications that can impact developing countries and also tries to help achieve broader development goals, such as the MDGs. Web conferencing can sometimes be affected from a client perspective since client experience can degrade rapidly as the number of conference clients rises, since it becomes harder to support video and audio from all clients.

When bandwidth is low and a client cannot use audio and video applications, the sound quality is not good and if this is the case the client has the ability to use a chat application where the communication is through instant messaging. Chat application will allow for clients to still communicate and the meeting to continue even if the bandwidth is low as it is known not to use a lot of bandwidth [6].

4. OBJECTIVES

Video conferencing is our main design goal. It is very important to make the video quality good in this kind of voice communication application. Minimum lag in the transmission is one of our design goals. Along with this, audio conferencing, group chat, group file sharing are also our goals.

To use any functionality of the application, the user needs to be registered at the server. This communication is via Wi-Fi and will incur no expenditure to the end users. Communication between any two users will be enabled as long as both are logged in at the server. An application server is required to register the users and maintain the information regarding the MAC, current IP address, username, and password and availability status of all the logged-in users. The clients may be concerned to same/different Wi-Fi given both the routers are registered at the same network. Server must be connected to same Wi-Fi network.

5. METHODOLOGY

[1] Establishing connection between Server and Client using Wi-Fi.

[2] Database management and Database connectivity.

[3] Parsing the Messages in the server and updating the database.

[4] Fetching IP's of all available users from server.

[5] Recording audio and video and playing in devices.

[6] Transmitting live audio and video call/conference.

[7] Setting up of Calling Functionality between Clients for video conference.

[8] Video conference testing.

[9] Audio conference testing.

[10] Group chat testing along with group file sharing.

5.1 Functions of different Classes

5.1.1 Client

DataBaseHandler :

This class handles the client side database. When first time application is installed, it creates a database contactManager with three fieldsuid, name and owner. Primary key is uid+owner. Database stores all the contact saved on the device. This class handles the operations like add contact, delete contact in the contact list.

TcpActivityAudio :

This thread listens for incoming call requests from all the users registered with server. When user receives a call It shows a alert box with two options accept or reject. It creates a tcp connection with the caller. And it sends and receives all the control messages during the call. This tcp connection closes when either of side ends the call.

TcpActivityFile :

This thread listens for incoming fil transfer requests from all the users registered with server. When user receives a request It shows a alert box with two options accept or reject. It creates a tcp connection with the caller. When user accept or reject the call, this tcp connection is closed. Receive file Thread receives the file in background.

TcpActivityVideo :

This thread listens for incoming video call requests

from all the users registered with server. When user receives a video call

It shows a alert box with two options accept or reject. It creates a tcp connection with the caller. And it sends and receives all the control messages during the call. This tcpconnection closes when either of side ends the call.

AccepCallThread :

This thread starts working when user accepts the audio call.

RejectCallThread :

This thread starts working when user rejects the video call.

AccepCallThreadF :

This thread starts working when user accepts the file transfer request.

RejectCallThreadF :

This thread starts working when user rejects the file transfer request.

AccepCallThreadV :

This thread starts working when user accepts the video call request.

RejectCallThreadV :

This thread starts working when user rejects the video call request.

PlayAudio :

This thread starts working when a audio call starts. It plays the UDP audio packets received from remote user.

RecordSend :

This thread starts when a audio call starts, It forms UDP audio packets and send them to remote users.

TcpConnection(Audio) :

This thread makes a TCP connection with remote user when user makes audio call.

Contact :

This class has two fieldsuid and name. Object of this class represent the contact.

ContactsFrag :

This class maintains all the contact activities. When user select a contact for audio/video/file this class initiates the threads
AudioConnection/VideoConnection/ConnectionProgress.

AudioConnectionProgress :

This thread starts working, when user selectsa contact for audio call. It shows the message like "user is not online","contact is not registered on server" or "user is busy". If user is online it receives the IP of user from server and makes audio call.

VideoConnectionProgress :

This thread starts working, when user selects a contact for video call. It shows the message like "user is not online","contact is not registered on server" or "user is busy". If user is online it receives the IP of user from server and makes video call.

ConnectionProgress :

This thread starts working, when user selects a contact for file transfer. It shows the message like "user is not online","contact is not registered on server" or "user is busy". If user is online it receives the IP of user from server and makes file transfer request to remote user.

SettingsFrag :

This class is used to change password. User have to provide three things: Old password, new password and confirm passwod.

FileShare :

This class starts working when user makes a file transfer request to remote user.

TcpConnection(File) :

It establishes a TCP connection with remote user to make a file transfer request.

SendAsynFile :

This background process starts when user send a file to

remoteuser.It shows a dialogue box to show the amount of file transferred.

Login:

This class loads when user starts the app. It fetches the userPreferences (server IP) saved on user the device and send the login request

to the server. On successful login Home intent is opened.

PrefActivity :

This class save the preferences of user on the device. Usercan give his preference on the login page.

Group audio conference call:

This class handles the main functionality of retrieving Ip's of selected users (for audio conference) from the server and sending them appropriate messages on clicking "start conference" and "stop conference" button.

SendIp :

This class sends the List of Ip's of all the users selected in the audio conference, to each and every user in the conference. This class starts when conference initiator starts the conference.

SendAudioMessage :

This class is used to send audio conference starting request to the selected users from audio conferencing page. Send exit message: This class is used to send exit message to all the users in the conference call. This class is invoked when any user leaves the conference.

ContactAdapter :

This class creates a view for displaying the contacts of

the user with a checkbox(for selecting them for audio conference).

FetchAsyncIp :

This class fetches Ip's of selected users(for audio conference) from the server.

Group audio conference receive :

This class handles the functionality of receiving audio conference requests from other users and also actions to be taken after receiving different types of requests.

SendAcceptFlag :

This class sends a confirmation to all users in the conference call that he she has accepted the call request and is now starting communication.

saveMessages append :

This class decodes messages received from other users and performs actions according to received messages.

Send exit message :

This class is used to send message to all other users in the conference call that he she has rejected the conference request.

Group chat list:

This class extends the fragment class, which is used to generate the list of friends added by the user in the contats in the selected fragment of group chat. Also the implementation of this class starts the group chat between all the selected friends from the contact list by fetching the IP address from the server.

GroupchatActivity:

This is the main UI class for having a group chat, this class has its own layout for showing messages, the list for showing online users and selecting the file and sending the same to the whole group. The messages and file are sent in seperate threads to all the selected friends.

SendMessage :

Inner class of group chat activity to send messages to the friends selected in the group chat. group message receive thread : A separate thread which starts at the home page, the thread handles all the messages requests on a particular port, either it is a new group chat request or a new message or an exit message. This thread also replies for a bad request.

SendFile Group Thread:

A thread class which spawn separate threads and manages all the spawned threads for sending, completion, failures of the specified file to all the friends in the group simultaneously.

ReceiveFile Group Thread:

A thread class which spawn separate threads and manages all the spawned threads for validating file request with group code start receiving, completion, failures of the file from the sending party, a maximum of 3 simultaneously receive is possible.

Send exit message:

A thread class to inform all the friends in selected group that the he has left the chat.

5.1.2 Server

VDOServer:
It initiates the OnlineChecker and Server Time threads. It continuously listens at port 6500 for all client requests. It creates a new Process Packet thread to process each received packet from a client.

OnlineChecker:

This class pings each online client every 5 minutes. If it does not receive a reply it updates the database to mark the client as offline.

ServerTime:

Send the current time of the server to the requesting clients.

ServerSetup:

sets up the mysql server and connects to it by taking the details from admin

Layout:

consists the main server frame

ListUsers:

shows the present number of online and offline users with their names

UserRegistration:

registers a user, deletes him from the database and forcefully kicks him out of the server when required.

6. REFERENCE

[1] www.developers.android.com

[2] https://vsee.com/blog/videoconference

[3] http://research.gigaom.com/report/why-videoconferencing-is-critical-to-business-collaboration

[4] http://people.cs.uct.ac.za/~zmanzi/downloads/Project%20Report.pdf

[5] http://people.cs.uct.ac.za/~tmvumbi/lit_rev_tresor.pdf